

FULL STACK WEB DEVELOPER

PART I

HTML & CSS

from basics to advanced



Panos M.

Full Stack Web Developer Part I: HTML & CSS

Panos Matsinopoulos

This book is for sale at <http://leanpub.com/full-stack-webdeveloper-part-1-html-and-css>

This version was published on 2020-01-27



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2019 - 2020 Tech Career Booster - Panos M.

Contents

| | |
|---|----------|
| The Bundle and the TCB Subscription | 1 |
| Part of Bundle | 2 |
| Bundle Goes with a TCB Subscription | 3 |
| Each TCB Subscription Goes with the Bundle | 4 |
| Credits To Photo on Cover Page | 5 |
| About and Copyright Notice | 6 |
| HTML & CSS | 7 |
| Chapter 1 - Introduction | 8 |
| Summary | 8 |
| Learning Goals | 8 |
| What is HTML | 9 |
| What is CSS | 10 |
| Tasks and Quizzes | 11 |
| Chapter 2 - Setup Development Environment | 13 |
| Summary | 13 |
| Learning Goals | 13 |
| Setup your development environment | 13 |
| We are using Linux | 14 |
| Linux Users - Ideal Environment | 14 |
| Windows Users - Let's get a Linux | 14 |
| HTML & CSS Editor | 20 |
| Tasks | 21 |
| Chapter 3 - Hello World! | 22 |
| Summary | 22 |
| Learning Goals | 22 |
| Hello World Page | 22 |
| Write the HTML code | 23 |
| View your HTML page on your browser | 23 |
| Tasks | 24 |
| Chapter 4 - Minimum HTML Document | 25 |
| Summary | 25 |
| Learning Goals | 27 |
| Minimum HTML Document | 28 |
| One <code>html</code> tag | 29 |
| One <code>head</code> and one <code>body</code> like humans | 29 |

CONTENTS

| | |
|---|----|
| Let's fill in the <head> | 32 |
| The title of our page | 36 |
| Let's fill in the <body> | 37 |
| Tasks and Quizzes | 38 |
| Chapter 5 - Heading Tags | 40 |
| Summary | 40 |
| Learning Goals | 40 |
| Heading Tags | 40 |
| Tasks | 42 |
| Chapter 6 - About White Space | 43 |
| Summary | 43 |
| Learning Goals | 44 |
| About White Space | 44 |
| Line break HTML tag | 47 |
| White spaces between words | 49 |
| Tasks | 51 |
| Chapter 7 - HTML Entities | 52 |
| Summary | 52 |
| Learning Goals | 53 |
| HTML Entities | 53 |
| Some Common Ones | 53 |
| HTML reserved characters | 53 |
| Currencies | 57 |
| Special Alphabet Characters | 58 |
| Other Symbols | 58 |
| Full References | 58 |
| Tasks & Quizzes | 58 |
| Chapter 8 - Paragraphs, Horizontal Lines And Comments | 60 |
| Summary | 60 |
| Learning Goals | 60 |
| Paragraphs, Horizontal Lines and Comments | 60 |
| Paragraphs | 60 |
| Horizontal Lines | 64 |
| Comments | 66 |
| Tasks | 68 |
| Chapter 9 - HTML Lists | 70 |
| Summary | 70 |
| Learning Goals | 70 |
| Lists | 71 |
| Unordered Lists | 71 |
| Ordered Lists | 72 |
| Nested Lists | 73 |
| Tasks | 74 |
| Chapter 10 - HTML Tables | 76 |
| Summary | 76 |
| Learning Goals | 76 |

CONTENTS

| | |
|---|-----|
| Tables | 77 |
| Starting and closing tags | 77 |
| Rows of the table | 78 |
| Columns of the table | 78 |
| Table Head and Table Body | 79 |
| Head Cells | 80 |
| Cell spacing | 82 |
| Cell padding | 84 |
| Column Span | 85 |
| Aligning | 87 |
| Row span | 92 |
| Styling tables | 100 |
| Tables and Emails | 100 |
| Tasks & Quizzes | 101 |
| Quiz: | 101 |
| Chapter 11 - First Simple CSS Rules | 102 |
| Summary | 102 |
| Learning Goals | 102 |
| First Simple CSS Rules | 102 |
| Style Rules and CSS Language | 102 |
| Default style values | 103 |
| Set the color of the text - color | 103 |
| Set the text size - font-size | 105 |
| Set border - border | 106 |
| Set background / fill color - background-color | 107 |
| The <style> tag | 108 |
| The reference to the CSS file | 110 |
| Tasks | 111 |
| Chapter 12 - First Encounter With CSS Selectors | 113 |
| Summary | 113 |
| Learning Goals | 113 |
| First Encounter with CSS Selectors | 113 |
| Selecting by the HTML tag name | 113 |
| Specificity - Containing-Contained Relationships | 116 |
| Multiple selectors | 126 |
| Tasks | 128 |
| Chapter 13 - More Advanced CSS Selectors | 131 |
| Summary | 131 |
| Learning Goals | 131 |
| More Advanced CSS Selectors | 132 |
| id attribute | 133 |
| CSS Selecting elements by id | 134 |
| The class attribute | 136 |
| Combining class selection with tag name selection | 140 |
| Multiple classes on same element | 142 |
| Tasks | 146 |

CONTENTS

| | |
|--|-----|
| Chapter 14 - Talk About Color | 151 |
| Summary | 151 |
| Learning Goals | 151 |
| Let's Talk about Color | 151 |
| Text Color and Background Color | 151 |
| Specifying the color value | 153 |
| RGB Color With Transparency | 156 |
| Tasks | 158 |
| Chapter 15 - Formatting Parts Of Text | 159 |
| Summary | 159 |
| Learning Goals | 159 |
| Formatting parts of text | 160 |
| tag | 160 |
| HTML formatting tags | 161 |
| Closing note | 165 |
| Tasks | 166 |
| Chapter 16 - CSS Box Model | 170 |
| Summary | 170 |
| Learning Goals | 171 |
| CSS Box Model | 171 |
| Boundaries of Body | 172 |
| Visible Border Box | 173 |
| Padding Area | 174 |
| Margin Area | 176 |
| CSS Box Model again | 176 |
| Use Chrome Developer Tools | 179 |
| Width and Box Sizing | 181 |
| Line Height | 190 |
| Inherited and Non-Inherited values | 193 |
| Applying Rule To All Elements | 199 |
| Why body background-color covers the whole canvas? | 202 |
| Tasks | 207 |
| Chapter 17 - Block And Inline Elements | 214 |
| Summary | 214 |
| Learning Goals | 214 |
| Block and Inline HTML Elements | 214 |
| Block HTML Elements | 214 |
| Inline HTML Elements | 218 |
| Default Block Elements | 220 |
| Default Inline Elements | 221 |
| Setting width and height for inline elements | 221 |
| Closing Note | 225 |
| Tasks | 225 |
| Chapter 18 - Dividing Page To Sections Using Divs | 228 |
| Summary | 228 |
| Learning Goals | 228 |

CONTENTS

| | |
|---|-----|
| Dividing Page to Sections Using <div> | 229 |
| Step 1: Empty Page | 230 |
| Step 2: Let's set the basic rules | 230 |
| Step 3: Dividing our page | 230 |
| Step 4 - Left Column | 233 |
| Step 5 - Middle Column | 237 |
| Step 6 - Right Column | 243 |
| Step 7 - Use float to position the elements. | 254 |
| Step 8 - Footer | 259 |
| Step 9 - Fixing footer problem with clear | 264 |
| Final Styling Touches | 266 |
| Tasks | 267 |
| Chapter 19 - HTML Links | 269 |
| Summary | 269 |
| Learning Goals | 269 |
| HTML Links | 270 |
| <a> tag | 270 |
| http:// or https:// | 273 |
| Anything a link | 275 |
| Styling the Anchor Links | 277 |
| Style links as buttons | 284 |
| Side Note - CSS comments | 285 |
| Links within the same page - Bookmarks | 286 |
| Links to Other Pages within Same Site | 297 |
| Styling the links on navigation bar | 313 |
| Tasks | 321 |
| Chapter 20 - Deploying Our Work | 323 |
| Summary | 323 |
| Learning Goals | 323 |
| Deploying Our Work | 323 |
| Static Sites | 323 |
| Sites are hosted on Servers | 324 |
| Companies Offering Hosting | 324 |
| Netlify - Create an Account | 324 |
| Deploy our first Site | 324 |
| Deploying a Site with Stylesheets/CSS | 325 |
| Deploying to Netlify using the command line | 332 |
| Netlify Optimizes our CSS | 332 |
| Disable Optimization of your CSS content | 333 |
| Tasks | 333 |
| Chapter 21 - Images | 335 |
| Summary | 335 |
| Learning Goals | 336 |
| Images | 336 |
| Image Files | 337 |
| tag | 337 |

CONTENTS

| | |
|--|-----|
| The <code>src</code> attribute | 338 |
| The <code>alt</code> and the <code>title</code> attributes | 339 |
| Image size | 340 |
| Images as Links | 348 |
| Text Around Images | 350 |
| Images and line height | 357 |
| Using images stored on our site | 362 |
| Tasks | 365 |
| Chapter 22 - Image Backgrounds | 369 |
| Summary | 369 |
| Learning Goals | 370 |
| Image Backgrounds | 370 |
| Image as a body background | 371 |
| Let's Be More Accurate | 373 |
| How To Stop Propagation | 375 |
| Body Without Content is Not Displayed | 376 |
| Adding Content to Body | 376 |
| Make body High Enough | 377 |
| Remove Margin | 378 |
| Where is The Happy Dogs Text? | 379 |
| One Image to Cover Whole Page | 381 |
| Center Image | 383 |
| Cover The Whole Body | 385 |
| Adding Text On the Page | 386 |
| Make the Image Fixed | 388 |
| Use A Big Image | 389 |
| Background Images on Parts of Page | 390 |
| Bottom Margin Will Not Work | 401 |
| A Blank Div Will Do | 402 |
| Why Bottom Margin Didn't Work? | 405 |
| Move Background to HTML instead | 407 |
| DRYing our Code | 407 |
| Background Size | 408 |
| Background Color | 414 |
| background shorthand property | 414 |
| Tasks | 416 |
| Chapter 23 - Element Positioning | 419 |
| Summary | 419 |
| Learning Goals | 424 |
| Element Positioning | 425 |
| Inline Elements | 425 |
| Block Elements | 427 |
| Float Property | 429 |
| Clear Property | 431 |
| Positioning Relative to Browser Window | 433 |
| Positioning Element Relative To Their Normal Position | 462 |

CONTENTS

| | |
|--|-----|
| Positioning Elements Relative To Other Element | 476 |
| Positioned Elements | 483 |
| z-index | 483 |
| Tasks | 502 |
| Chapter 24 - CSS Units | 506 |
| Summary | 506 |
| Learning Goals | 506 |
| CSS Units | 506 |
| Everything is pixels | 506 |
| Absolute Units | 507 |
| Default Font Size | 507 |
| Default Font Size and Absolute Units | 508 |
| Relative Units | 511 |
| Tasks | 516 |
| Chapter 25 - Collapsing Margins | 518 |
| Summary | 518 |
| Learning Goals | 520 |
| Collapsing Margins | 520 |
| What is collapsing margins? | 526 |
| Parents and Children | 527 |
| How can we preserve both parent and children margins? | 528 |
| How can we eliminate the collapsing vertical margins phenomenon? | 531 |
| Why do the vertical margins collapse? | 534 |
| Closing Suggestion | 534 |
| Tasks | 535 |
| Chapter 26 - Fonts and Typeface | 536 |
| Summary | 536 |
| Learning Goals | 539 |
| Fonts and Typeface | 540 |
| Font Families | 540 |
| General Families | 540 |
| Specifying Font Family with CSS | 543 |
| Default Font Families | 549 |
| Safe CSS Font Stacks | 555 |
| Make Sure Browser Is Using the Font We Want | 555 |
| Other Properties of the Fonts | 559 |
| Tasks | 559 |
| Chapter 27 - HTML Forms | 562 |
| Summary | 562 |
| Learning Goals | 573 |
| HTML Forms | 574 |
| input of type="text" | 575 |
| label | 577 |
| select | 579 |
| input of type=submit | 587 |
| form | 593 |

CONTENTS

| | |
|---|-----|
| Input elements' ID | 595 |
| autofocus | 599 |
| label for attribute | 601 |
| required | 603 |
| Sign Up Form | 605 |
| input with type email | 607 |
| input with type password | 607 |
| input with type checkbox | 608 |
| Multiple Checkboxes | 610 |
| Multiple Select Box | 612 |
| input of type radio | 615 |
| checked attribute | 620 |
| textarea | 622 |
| placeholder attribute | 626 |
| title | 629 |
| autocomplete attribute | 631 |
| input with type number | 633 |
| min and max attributes | 636 |
| step attribute | 637 |
| input with type range | 641 |
| value attribute | 643 |
| Grouping Selection Options | 645 |
| input of type color | 647 |
| input of type date | 650 |
| input of type file | 652 |
| Grouping Form Fields | 654 |
| Tasks | 657 |
| Chapter 28 - Advanced CSS Selectors | 659 |
| Summary | 659 |
| Learning Goals | 662 |
| Pseudo-classes | 663 |
| Pseudo-elements | 677 |
| Other CSS Selectors | 687 |
| Tasks | 687 |
| Chapter 29 - HTML5 | 690 |
| Summary | 690 |
| Learning Goals | 692 |
| HTML5 | 692 |
| Semantic Elements | 693 |
| HTML5 Graphics | 712 |
| HTML5 Multimedia | 712 |
| Tasks | 718 |
| Chapter 30 - Some Other HTML Elements | 721 |
| Summary | 721 |
| Learning Goals | 724 |
| iframe | 724 |

CONTENTS

| | |
|--|-----|
| blockquote | 729 |
| code and pre | 732 |
| Tasks | 734 |
| Chapter 31 - Mobile Friendly Web Pages | 736 |
| Summary | 736 |
| Learning Goals | 745 |
| Introduction | 745 |
| The Viewport | 745 |
| Media Queries | 759 |
| A more complex example page with responsive design | 764 |
| Loading CSS files with media query | 791 |
| Responsive Images | 791 |
| Responsive Videos | 797 |
| Responsive Frameworks | 797 |
| Tasks | 798 |

The Bundle and the TCB Subscription

Part of Bundle

This book is sold alone but better if you buy as part of the bundle [Full Stack Web Developer - The Bible](#).

Bundle Goes with a TCB Subscription

When you purchase the bundle, then you have full access to the contents of the [TCB Courses](#).

Each TCB Subscription Goes with the Bundle

Moreover, this goes vice-versa. If you purchase the subscription to the [TCB Courses](#), then you are automatically eligible for the [Full Stack Web Developer - The Bible](#) bundle.

Credits To Photo on Cover Page

We have designed the cover page, but the photo in the middle is the creation of our friend Telis Marin. He is an amateur photographer. He doesn't have an official Web site where you could find more of his amazing photos. Hence, if you want to see more of his work, the only way you can now do it is by making him an FB Friend [here](#).

Telis Marin is an author, a publisher [Edizioni Edilingua](#) and a teacher trainer. He has written more than 20 books for learning Italian, which are used by schools and universities in over 80 countries.

About and Copyright Notice

Full Stack Web Developer - Part I - HTML & CSS 1st Edition, August 2019

by Panos M. for Tech Career Booster (<https://www.techcareerbooster.com>)

Copyright (c) 2019 - Tech Career Booster and Panos M.

All rights reserved. This book may not be reproduced in any form, in whole or in part, without written permission from the authors, except in brief quotations in articles or reviews.

Limit of Liability and Disclaimer of Warranty: The author and Tech Career Booster have used their best efforts in preparing this book, and the information provided herein “as is”. The information provided is delivered without warranty, either express or implied. Neither the author nor Tech Career Booster will be held liable for any damages to be caused either directly or indirectly by the contents of the book.

Trademarks: Rather than indicating every occurrence of a trademarked name as such, this book uses the names only in an editorial fashion and to the benefit of the trademark owner with no intention of infringement of the trademark.

For more information: <https://www.techcareerbooster.com>

HTML & CSS

This section will teach you two languages that the browsers are using to render a Web Page. HyperText Markup Language (HTML), which is used to structure the Web Page with visual elements, and CSS (Cascading StyleSheets), which is used to tell things about the look and feel of the content, like the font size and the color.

Chapter 1 - Introduction

Summary



In this chapter we are going to learn about HTML, the HyperText Markup Language. HTML is a markup language that web browsers use to interpret and compose text, images and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser.



However, we can use CSS, Cascading Style Sheets rules, in order to style the HTML elements the way we want. In this chapter you will see a sample CSS file.

Learning Goals

1. Understand what HTML is.
2. Understand what CSS is.

3. Know how to view the source of an HTML page and download its content to a file.

I would like to welcome you to the HTML and CSS Section of the book *Full Stack Web Developer - The Bible*. This is the first section of the book. I will teach you how to write HTML pages and style them using CSS.

What is HTML

HTML stands for **HyperText Markup Language**. It is the language that we use to describe the contents of an HTML document. The HTML document is a text document, i.e. you can open it and edit it with a simple text editor. It contains HTML code that can be understood by your Internet browser. Your browser reads the HTML document and draws, on your screen, in its window, the content that the HTML document defines, using some specific HTML rules. The final result is the HTML page that you look at every time you visit a Web page.

As a language, HTML follows some rules. You cannot just write anything you like. You need to learn the language and write content that is well structured and syntactically correct, according to the HTML language specification.

This section, **HTML & CSS**, will teach you HTML, how to write correct HTML content.

HTML sample

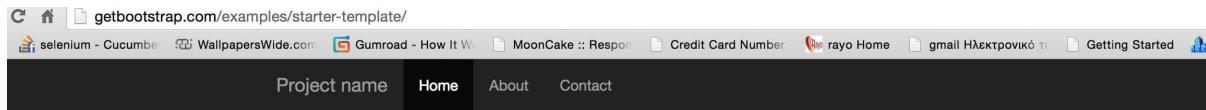
Do you want to see a simple HTML document? Here it is:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Simple HTML Document</title>
5   </head>
6   <body>
7     <h1>This is a simple HTML document.</h1>
8   </body>
9 </html>
```

(the above code snippet online)

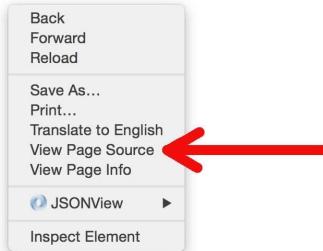
Do you want to see a more complex HTML document? Here it is how:

Visit the sample [Twitter Bootstrap page](#) with your browser. Then, right click on a blank area on the page and select *View Source*. Like I am doing that using my Chrome browser here:



Bootstrap starter template

Use this document as a way to quickly start any new project.
All you get is this text and a mostly barebones HTML document.



[View HTML Page Source Example](#)

What you will see, is the actual HTML code for this particular page. You may not be able to understand all the bits and pieces of this HTML document. Don't get overwhelmed. You will learn a lot by reading this book.

What is CSS

CSS stands for Cascading Style Sheets and it is the language that we use to style our HTML page, with nice colors and fonts. Hence, whereas HTML defines the structure of the document, CSS defines the look and feel of the page.

Like HTML documents, CSS documents are stored in pure text document format, which means that you can edit them with a simple text editor. But, again, it is a language and the author of a CSS document needs to "speak" the same language in order to write a well behaving CSS document.

In this section we are going to learn how to write CSS to style our HTML pages.

CSS sample

Do you want to see a simple CSS fragment? Here it is:

```

1 h1 {
2   color: green;
3   font-weight: bold;
4   font-size: 32px;
5 }
```

(the above code snippet online)

Tasks and Quizzes

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create a file with the contents of the HTML page of [Navbar example](#).
2. Send the file to your mentor, if you want feedback.

Note: It's just the plain code of the page returned back to browser is the main page that we are after. Not code from other associated / referenced resources.

Quiz:

Question 1:

Type: Fill in the Blanks

What is HTML used for?

Your answer here: _____

Question 2:

Type: Fill in the Blanks

What is CSS used for?

Your answer here: _____

Question 3:

Type: Fill in the Blanks

What is the file name extension for HTML documents?

Your answer here: _____

Question 4:

Type: Fill in the Blanks

What is the file name extension for CSS documents?

Your answer here: _____

Question 5:

Type: Multiple Choice - Choose Only One

The following document is an HTML document:

```
1 <!DOCTYPE html>
2 <html>
3   <head></head>
4   <body></body>
5 </html>
```

- a. Yes
- b. No

Your answer here: _____

Question 5:

Type: Multiple Choice - Choose Only One

What type of document is this one?

```
1 body {
2   font-size: 8pt;
3   color: red;
4 }
```

- a. HTML
- b. Java
- c. CSS
- d. Ruby

Your answer here: _____

Chapter 2 - Setup Development Environment

Summary



Linux Logo

In this chapter we are setting up our development environment. This is going to be Linux. We explain why Linux is the ideal environment for Web development. However, even if you have a Windows machine, we explain how you can install a Linux virtual machine, or install Linux alongside Windows.



Code Editor Logo

We also give you some hints about HTML and CSS editors that you can use while you will be doing your development and your exercises for this section.

Learning Goals

1. Ways to install Ubuntu on a Windows machine.
2. Find out about HTML & CSS editors on Linux.

Setup your development environment

In order to study and implement the exercises and tasks of this section, you need to setup your development environment to be able to write HTML and CSS code.

This chapter here is trying to help you setup your development environment.

We are using Linux

A big portion of the course uses *Ruby* and *Ruby on Rails*. These programming tools have problems when installed on Windows machines. In general, this course favors the usage of machines that are running Linux. Any decent variation of it (Debian or Ubuntu or Mac OS X or other similar).

However, if you do have Windows machine, this is not a problem as long as you follow the instructions to install a Linux OS alongside Windows, or as a virtual machine inside your Windows instance.

Go to the section that applies to you:

- [Linux Users & Debian or Ubuntu or Mac or similar](#)
- [Windows Users](#)

Linux Users - Ideal Environment

Note: Skip this section if you are not running a Linux distribution operating system (or a Linux-like one). For example, if you are not running Ubuntu, neither Debian, neither Mac OS X, neither similar, then you need to skip.

You have the ideal machine for development. Go to next paragraph to read what editor we recommend in order to carry out the exercises on this section.

Windows Users - Let's get a Linux

Note: Skip this section if you do not have a Windows machine

If you have Windows you need to find a way to have access to a Linux machine. There are several ways you can do that. These are the two that we recommend.

1. Install Ubuntu alongside your Windows installation and create a dual boot system. This will allow you to have both systems installed on the same machine. When the machine starts, you have to choose which operating system to log into, OR
2. Install a virtual machine with Linux operating system.

Option 1 - Install Ubuntu alongside your Windows installation

This is our preferred way to gain access to a very popular Linux distribution. The Ubuntu one. It will install Linux alongside your existing Windows installation (as long as there is enough disk space on your hard disk).

[This is the official Ubuntu page](#) if you want to download Ubuntu and install it on your desktop machine.

1. Download the Ubuntu version that corresponds to your machine architecture (64-bit vs 32-bit).
2. Then follow [the instructions here](#) in order to burn a DVD with Ubuntu.
3. Then follow [the installation instructions here](#) in order to install Ubuntu alongside your Windows installation. Watch out on the **Allocate drive space** step on which instructions explain to you how you can install Ubuntu alongside Windows.

Option 2 - Create a Linux virtual machine inside your Windows instance

If you do not want to install Ubuntu as dual boot alongside Windows, you can install it *inside* Windows. You may have heard about virtual machines. They are pieces of software that simulate an operating system, the guest, but they run inside another, the host. They are not real machines. That's why we call them virtual machines.

One program that can help you run virtual machines is Oracle Virtual box. So, the plan here is:

1. Download and Install Oracle VM Virtual Box
2. Download and Install Oracle Virtual Box Extension Pack
3. Download the Ubuntu virtual machine that we prepared for you
4. Start Oracle VM
5. Import the appliance that you have downloaded from step 3

and you are ready to go with a Linux inside your Windows.

Let's see in a little bit more detail the above steps:

1. Download and Install Oracle VM Virtual Box

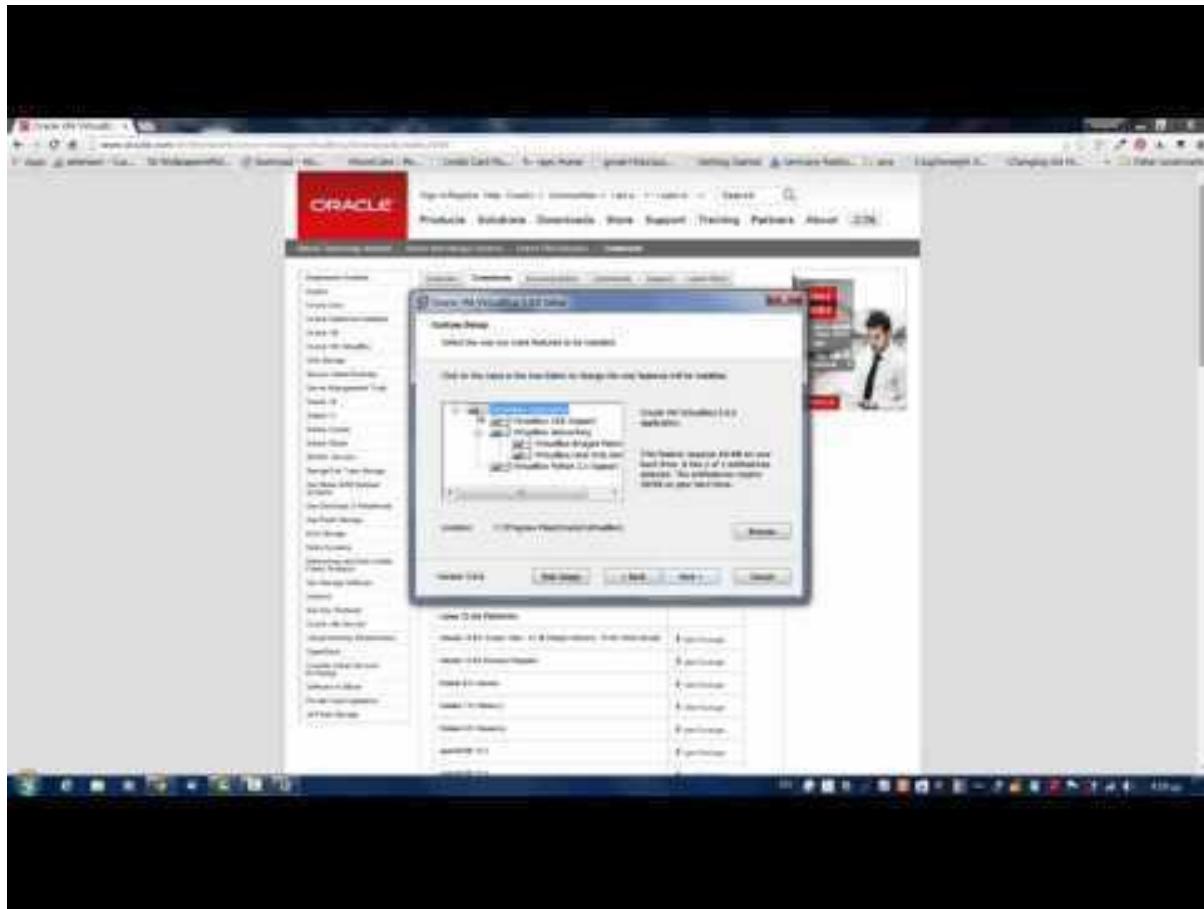
Download the Oracle VM Virtual Box from the official site: [Oracle VM Virtual Box](#).

Important Since you are a Windows user, you need to download the *Windows Installer (Windows 32/64 bit)*. The fact that we are later on running an Ubuntu virtual appliance inside the Virtual Box is irrelevant. The point here is to download the Oracle VM Virtual Box that will work as a host of your virtual appliance and need to be compatible with your host operating system, which is Windows.

After the download completes successfully, start the Windows Installer and follow the steps to install Oracle VM Virtual Box.

Here is a short video that shows how one can install Oracle VM Virtual Box on a Windows machine.

[Download and Install Oracle VM Virtual Box](#)



Download and Install Oracle VM Virtual Box - Preview Image

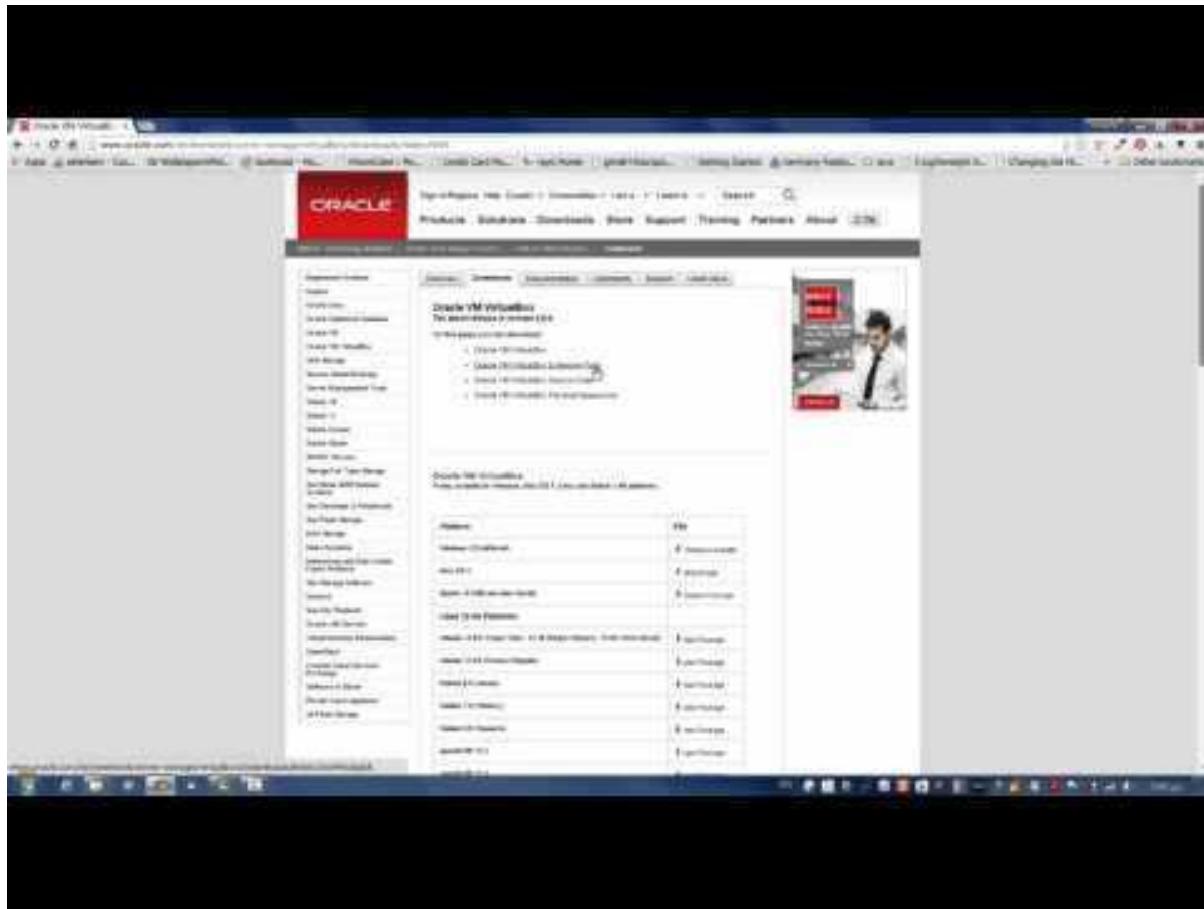
2. Download and Install Oracle Virtual Box Extension Pack

On the same page [here](#), there is a link to download and install Oracle VM Virtual Box Extension Pack. This is the same download irrespective of your operating system. Find the link and execute the download. Download the latest one that corresponds to the Oracle VM Virtual Box that you have downloaded on the previous step.

After you download, double click the file to install it.

Here is a short video that shows how one can install Oracle VM Virtual Box Extension Pack on a Windows machine.

[Download and Install Oracle Virtual Box Extension Pack](#)



Download and Install Oracle Virtual Box Extension Pack - Preview Image

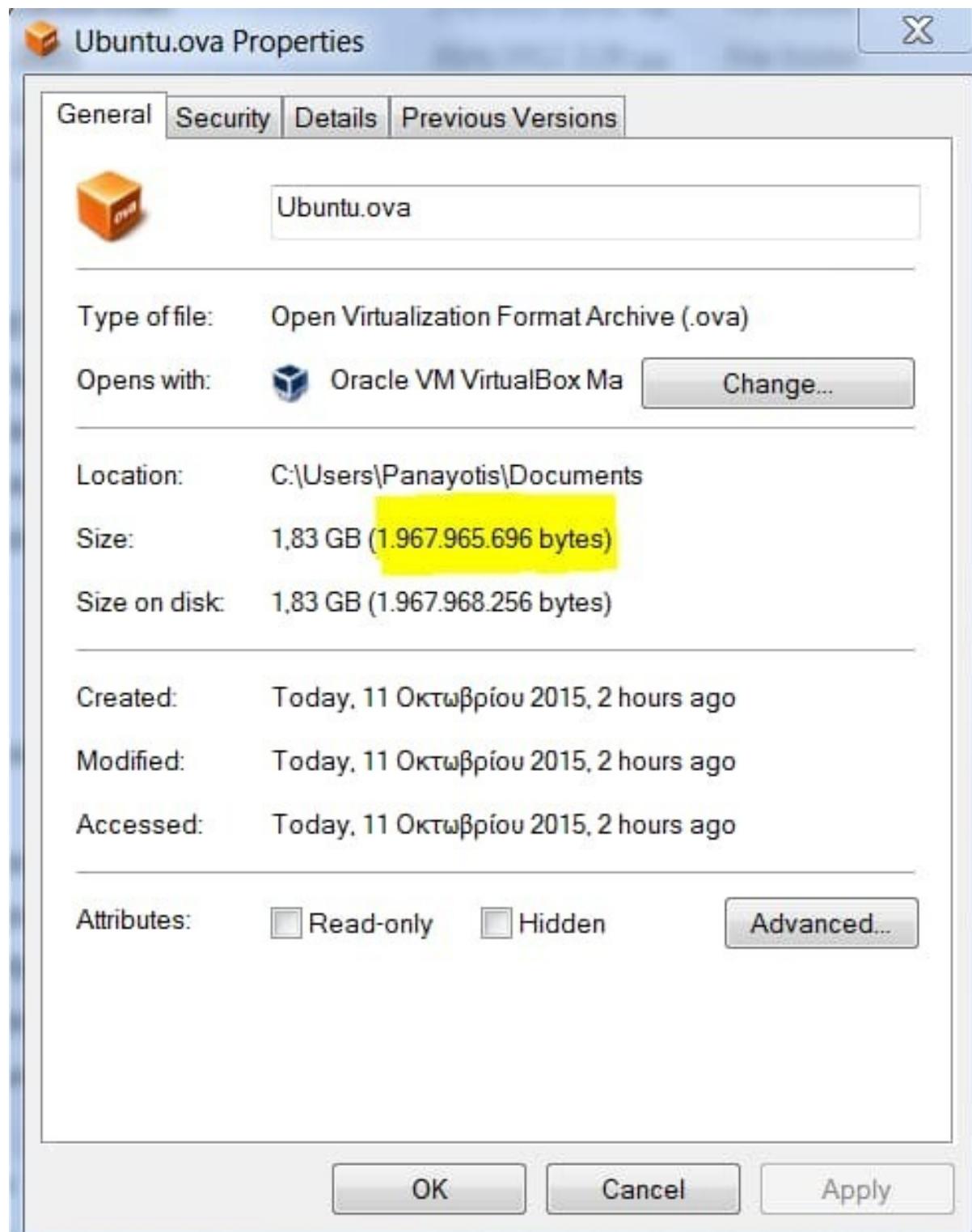
3. Download the Ubuntu Virtual Machine

In order to make your life easier, we have prepared an Ubuntu virtual machine that you can download. If you already have another Ubuntu VM, you can use your own, but ours is ready to jump in and take your course.

Here is the download link: [Ubuntu Virtual Machine](#)

Note that the final downloaded file needs to be exactly 1,967,965,696 bytes long. Here is how you can check that:

- Right click on the file (after you locate it on your file browser)
- Select “Properties”
- See the number that is displayed for the file size:



Number of bytes of Ubuntu.ova file

If the file does not have exactly this number of bytes, then it has not been downloaded correctly, which means it might be corrupted and step 7, below, will fail to import it into Oracle VM Virtual Box Manager.

Note: The Ubuntu.ova file, which is the file of the Ubuntu virtual machine that we

are asking you to download is a very big file (1.83 GBytes). So, under slow internet connections this will take too long to be downloaded.

4. Start Oracle VM Virtual Box

Ok. This is easy. Start the Oracle VM Virtual Box manager that you have downloaded and installed on step 1. (Of course, you do it only if it is not already started)

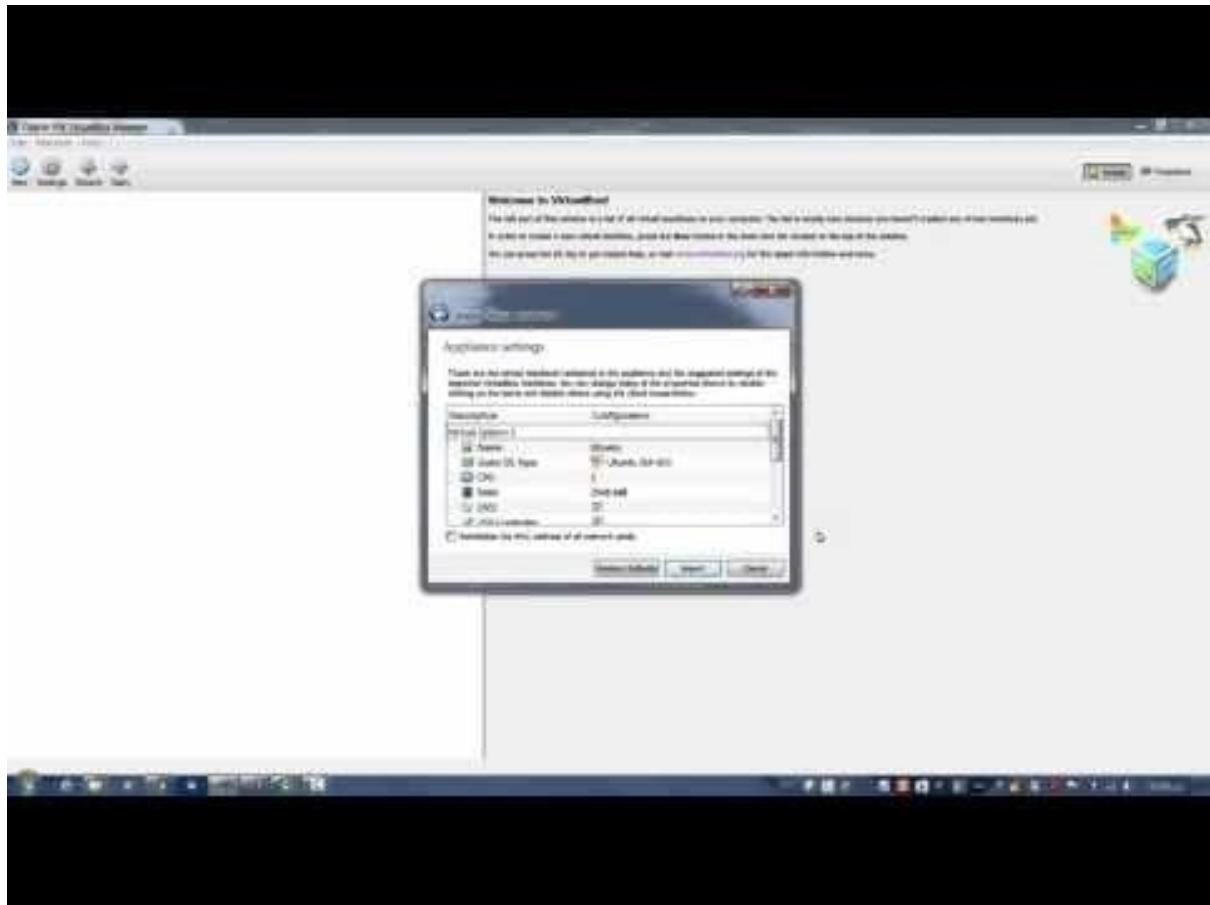
5. Import the Ubuntu Virtual Machine

Here are the instructions that you should follow to import the Ubuntu VM that you have downloaded on step 5.

1. Go to File > Import Appliance...
2. On the dialog box that opens, click the folder with the green arrow pointing up and select the Ubuntu.ova file that you have just downloaded.
3. Click Next to move to “Appliance Settings” dialog box.
4. Select re-initialize the MAC address of all network cards.
5. Click the “Import” button.

Here is a short video that demonstrates the import process:

[Import Virtual Box Appliance](#)



Import Virtual Box Appliance - Preview Image

Import will start and it will take some time.

As soon as it finishes, continue as follows:

1. Start appliance. Now that you have imported the appliance, you need to start it. Select the appliance in the list of appliances and click on the button Start on the toolbar of Oracle VM Virtual Box Manager.
2. The virtual appliance will start and it is asking you to log in. You need to log in with username `lione1` and password `lione1`.

Then you are good to go! You now have a brand new Linux machine inside your own Windows host.

From now on and till the end of the course you will be working inside your virtual machine, i.e. Ubuntu.

HTML & CSS Editor

This section of the course requires you to have an editor that will allow you to write HTML and CSS. So, you need a simple Web developer editor that supports, at least, some kind of syntax highlighting.

Here is a list of our favourite:

- Mac OS X: CotEditor. Get it from [here](#).
- Debian, Ubuntu and other Linux: Geany. Get it by giving the command `sudo apt-get install geany`, on a terminal. The official page is [here](#).

Note:

1. Users of our Ubuntu virtual appliance already have Geany installed. They can find its icon on the launch toolbar on the left.
2. If you want, send us your operating system and preferred HTML & CSS editor. We will include it in the above list. Thanks!

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. You need to have Linux installed, either natively or using a virtual machine.
2. Tell your mentor what operating system workstation do you have, and what you did in order to have access to a Linux operating system.

Chapter 3 - Hello World!

Summary

In this chapter we are going to create our first HTML page, which will be looking like this:



Hello World

Hello World Page

Learning Goals

1. Create an HTML document
2. Save it with a filename extension that is compatible with browsers standard HTML pages
3. Open the HTML document using the browser

Hello World Page

It has become a tradition in computer programming schools, when one is introduced to a new language, programming or not, to create a first artifact that would say hello to the world.

Note: You can read about this tradition [here](#)

So, let's write our first HTML page that will be displaying to the visitor, the phrase *Hello World*.

Write the HTML code

Follow the instructions below to create your first HTML page:

1. Open the [text editor](#). Create a new file and save it with the name *hello-world.html*
2. Enter the following content:

Important: Do not copy the content from this reading to your text editor. Type the content yourself. This is the only way you can learn and become a good developer. Note that the filename extension needs to be `html`. This is how your operating system and your browser will recognize this file as an HTML page.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Hello World Page</title>
6    </head>
7
8    <body>
9      <h1>Hello World</h1>
10   </body>
11 </html>
```

(the above code snippet online)

View your HTML page on your browser

Open your browser if it is not open. Select to open a file and browse to locate the file that you have just created (*hello-world.html*). When you open it, the browser will read the code that you have written and will display that as an HTML page. Something like the following:



Hello World

Hello World Page

Congratulations! You have just written your first HTML page and it has been successfully rendered by your browser.

Go back to the chapter about development environment to see what is the text editor that we have chosen to use to write simple text files. If you have any doubts contact your mentor.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

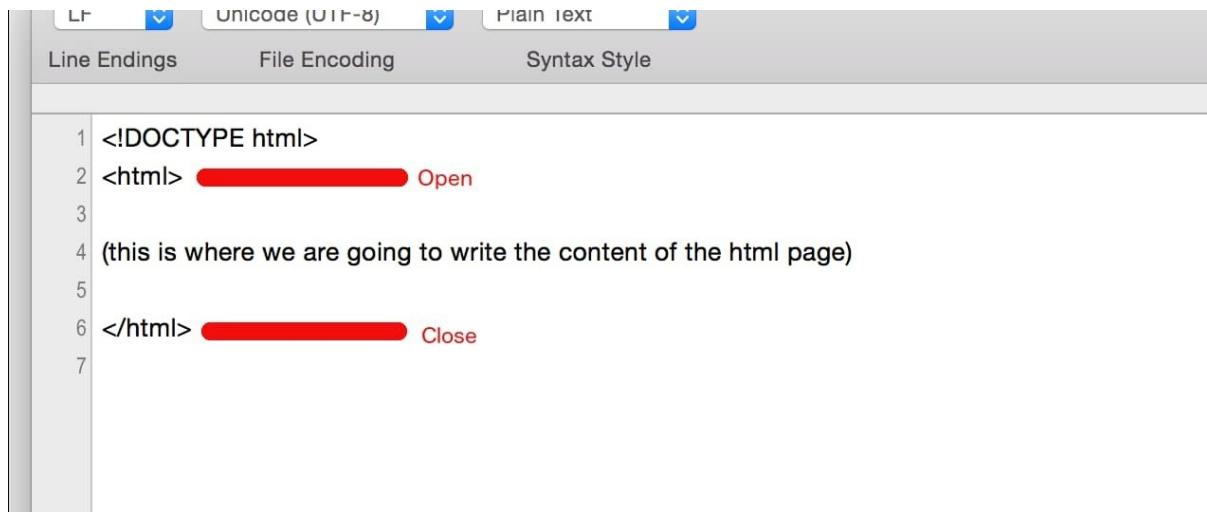
Task details:

1. Send a message to your mentor with the screenshot of your browser displaying your HTML page.
2. Send the HTML file with your HTML Hello World page to your mentor.

Chapter 4 - Minimum HTML Document

Summary

In this chapter we are going to analyze the content of the minimum HTML document. Also, we are going to talk about the `html` element, and how we open and close HTML tags.



The screenshot shows a code editor window with the following interface elements at the top:

- Line Endings dropdown
- File Encoding dropdown set to "Unicode (UTF-8)"
- Plain text dropdown
- Syntax Style dropdown

The code editor displays the following text:

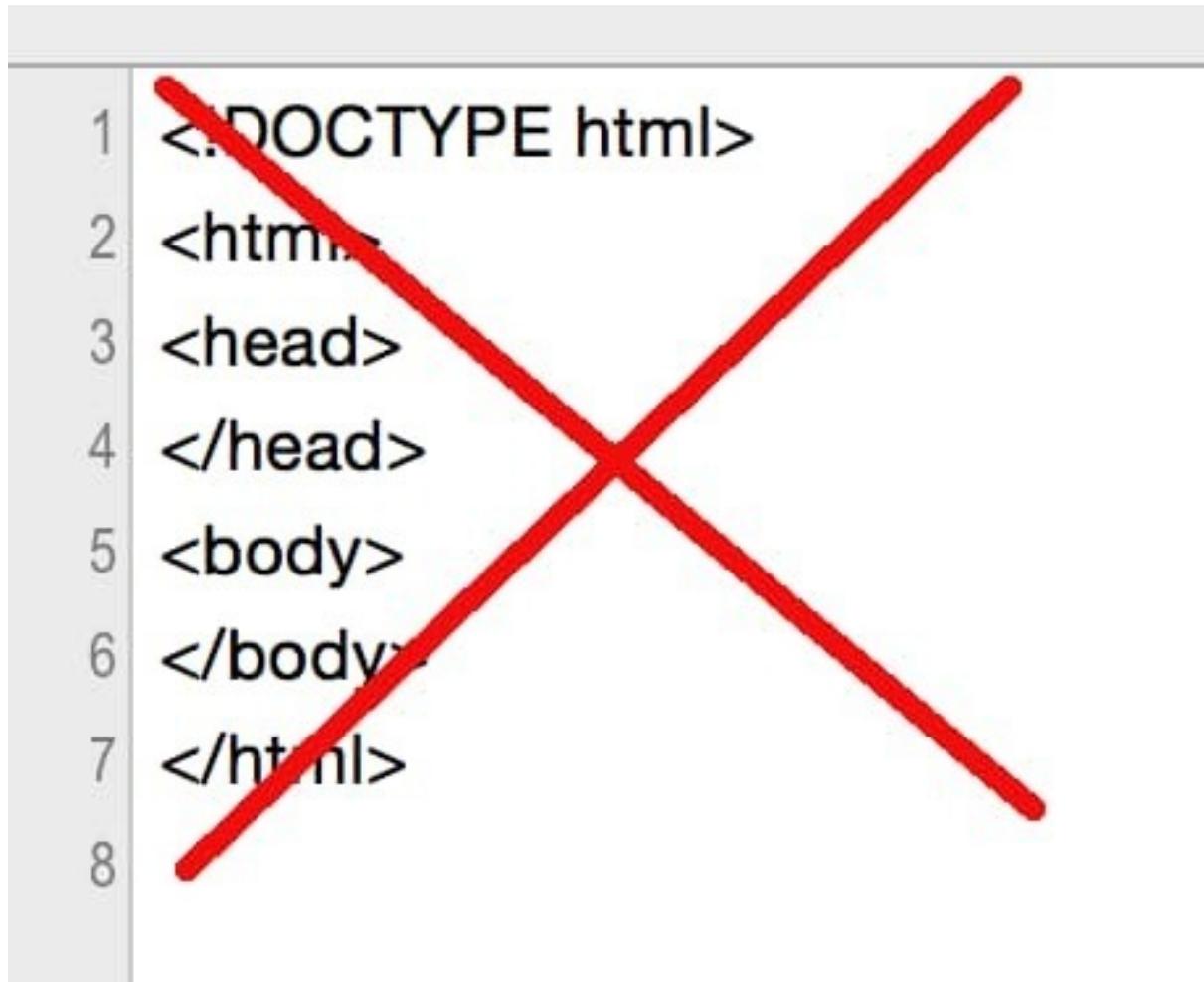
```
1 <!DOCTYPE html>
2 <html> Open
3
4 (this is where we are going to write the content of the html page)
5
6 </html> Close
7
```

The word "Open" is highlighted with a red background in the editor, and the word "Close" is also highlighted with a red background.

Opening and Closing Tags

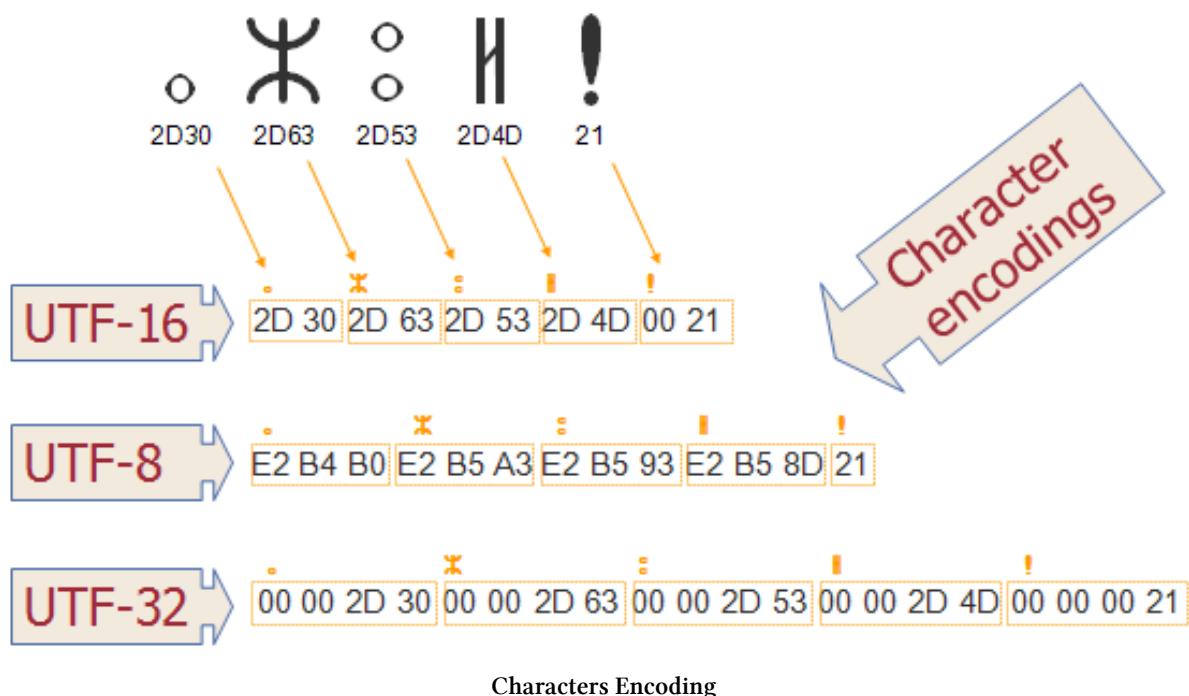
We are going to talk about the `body` element.

We are going to talk about correct styling of HTML documents and how we should avoid writing code like this:



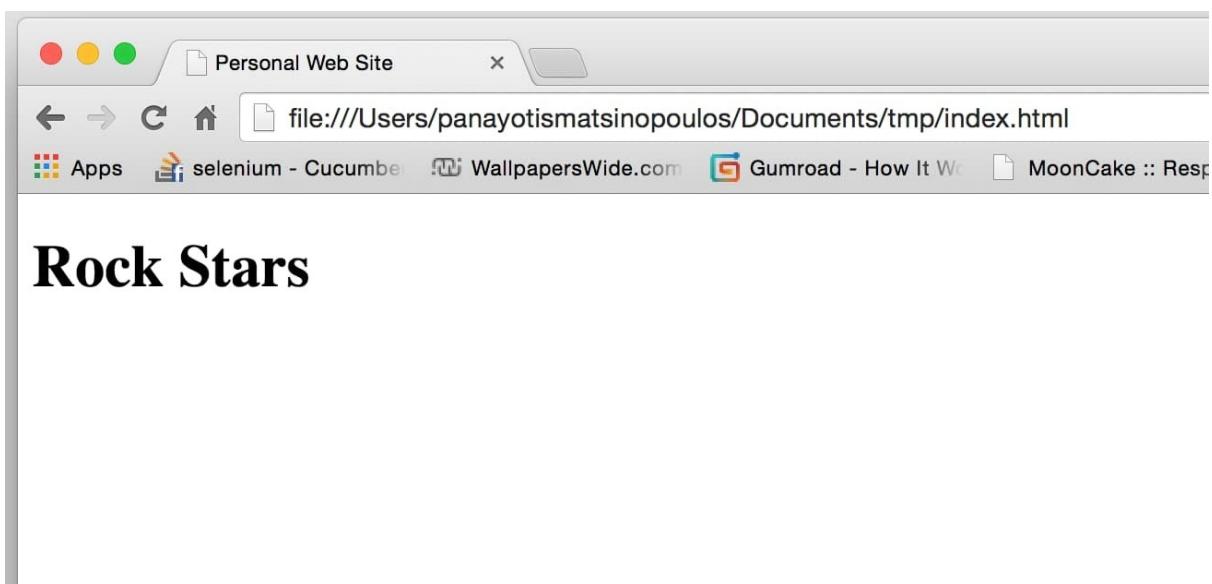
Don't write documents without indentation

We are also going to talk about character encoding, ASCII, Extended ASCII and UTF-8. Specifying the character encoding will make sure that the browser is interpreting and displaying special characters correctly.



We are also going to talk about the `meta` tag and the `title` tag.

Finally, we are going to build another simple web page like this:



Web Page we build on this Chapter

Learning Goals

- Understand why you need the DOCTYPE on top of the HTML page.
- Learn about the `<html>` tag.
- Learn how tags have two parts, the opening and the closing part.
- Learn about indentation

- Learn about the `<head>` section.
- Learn about the `<body>` section.
- Learn about the `<meta charset>` tag.
- Learn about the tag attributes and values.
- Learn about the content of an HTML tag.
- Learn about the `<title>` tag.
- Learn about the `<h1>` tag.

Minimum HTML Document

An HTML Document is composed of a list of **tags** that have special meaning and need to be put on specific positions in the document so that the whole content is finally interpreted correctly by the browser.

You do not have to know all the available tags in order to be able to write HTML pages. An HTML page can be constructed with the minimum available tags as given in the next example, which is an HTML page that only has a title and a phrase in its body.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Hello World Page</title>
6      </head>
7
8      <body>
9          <h1>Hello World</h1>
10     </body>
11 </html>
```

(the above code snippet online)

We start an HTML page with the fragment:

```
1  <!DOCTYPE html>
```

(the above code snippet online)

Ok, actually, this is not real HTML code, according to the HTML specification. We, the developers, put that there, in order to tell the browser that it has to render the page using the standard HTML specification rules. If we do not do that, the browser might consider that this is an *old* document page and might try to interpret it, render it, using *quirks* old-pages rendering logic. And, in that case, our brand new web page might not be displayed properly.

Important Let's make sure that all of our web pages start with `<!DOCTYPE html>`.

One `html` tag

Our HTML page, with real HTML code, actually starts with the opening tag: `<html>` This is the start and parent tag of all the other tags that we will put inside our HTML page. There is 1 and only 1 `<html>` element inside every HTML page. Also, it is the container of all the other elements that comprise the HTML page.

And since every opening tag needs to have a corresponding closing tag, whenever you create the opening tag, don't forget to immediately type in the closing one, and then leave some space in between to insert the rest of the tags that will comprise your document.

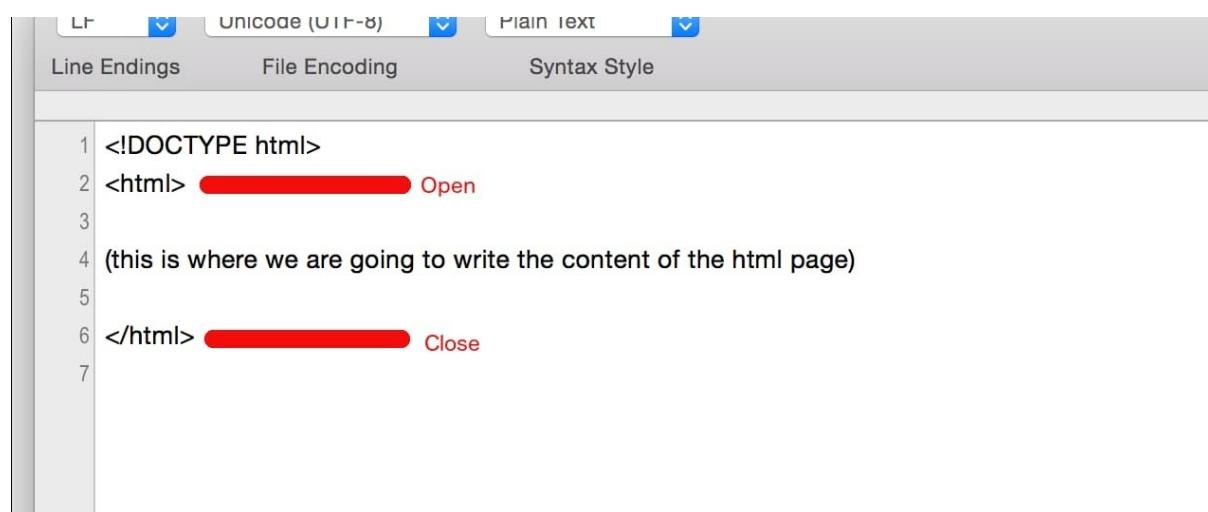
Here it is:

```
1 <!DOCTYPE html>
2 <html>
3
4
5 </html>
```

(the above code snippet online)

Is that clear?

Note The opening tags encloses the tag name within `<` and `>` symbols. The closing tag encloses the tag name within `</` and `>` symbols. See the image below:



The screenshot shows a code editor window with a toolbar at the top. The toolbar includes buttons for Line Endings (LF), File Encoding (Unicode (UTF-8)), Syntax Style (Plain text), and a zoom control. Below the toolbar is a status bar with tabs for Line Endings, File Encoding, and Syntax Style. The main area of the editor displays the following code:

```
1 <!DOCTYPE html>
2 <html> Open
3
4 (this is where we are going to write the content of the html page)
5
6 </html> Close
7
```

A red horizontal bar highlights the word "Open" next to the closing parenthesis of the opening tag, and another red horizontal bar highlights the word "Close" next to the opening parenthesis of the closing tag.

Opening and Closing Tags

One head and one body like humans

All HTML pages have one head section and one body section. Like the humans (or at least most of them).

- head needs to appear before the body section.

- body needs to appear after the head section.
- There can only be **one** head section.
- There can only be **one** body section.
- Sections start with the *opening* tag and end with the *closing* tag.

How do we *open* a head section? With `<head>`.

How do we *close* a head section? With `</head>`.

How do we *open* a body section? With `<body>`.

How do we *close* a body section? With `</body>`.

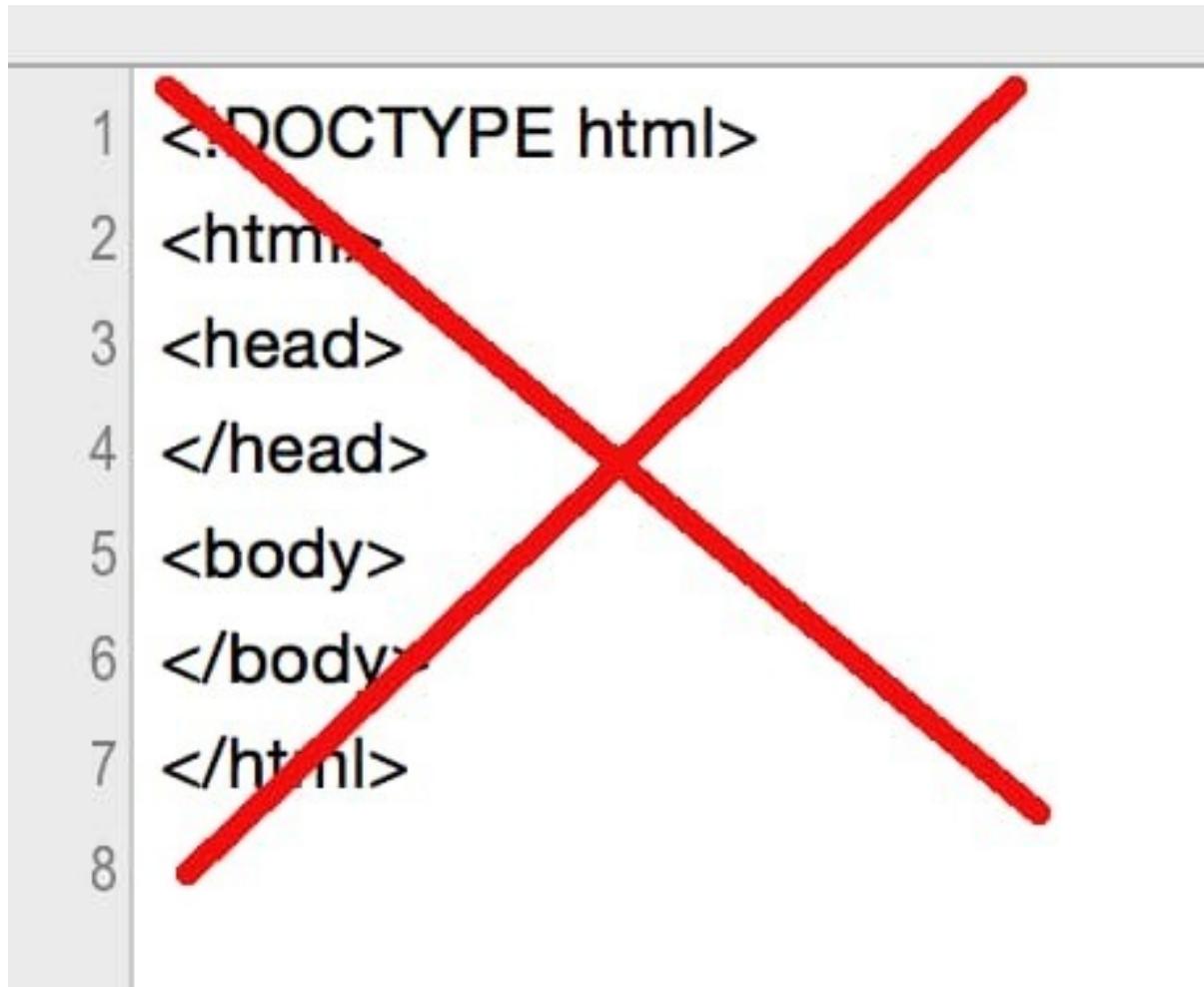
There it is:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5 <body>
6 </body>
7 </html>
```

(the above code snippet online)

Oooooops!!! Who can read such an ugly document?

Nobody!



Don't Write Documents Without Indentation

Don't write documents without indentation

You should be using indentation in order to make your document/code look clean and easy to read. The indentation needs to be there in order to show the parent - child (or container - contained) tags relationship. The contained (or child) opening tag needs to be 2 spaces to the right of its parent opening tag. Same goes for the closing tags.

So, the correct way to write the above ugly document is this nice one:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4
5      (here we are going to have the contents of the head section)
6
7  </head>
8
9  <body>
10
11     (here we are going to have the contents of the body section)
```

```
12
13     </body>
14 </html>
```

(the above code snippet online)

You can see that `<head>` and `</head>` are 2 spaces to the right of the column of corresponding `<html>` and `</html>`. You do know that `<head>` section is child of the `<html>` section, i.e. it is first-level contained element of the `<html>` section. Hence, nesting, or indentation needs to be 2 spaces.

Same goes for `<body>` and `</body>`.

And `<head>` and `<body>` are on the same column-starting level, because they are both same-level (1st) children of the `<html>` tag.

Important * Make sure that you use 2 spaces for indentation * Closing tags need to exist on the same column depth / level like their corresponding opening one.

Does this picture make it more clear?

```
1 <!DOCTYPE html>
2 <html>parent
3 <head>child
4 (here we are going to have the contents of the head section)
5 </head>
6
7 <body>
8 (here we are going to have the contents of the body section)
9 </body>
10
11 </html>
```

Indentation with picture

Let's fill in the `<head>`

Specifying the `<head>` start and the `<head>` end is not enough. We need to give the content of the `<head>` section. But you need to know that whatever we put inside the `<head>` section is not

displayed by the browser. It is used by the browser to make sure it has enough information in order to display the actual content of the page, which is in the `<body>` section.

So, what do we put inside `<head>`? We are going to see the most important ones.

Meta info about the encoding

We are using the following tag in order to say that the actual character set used in the document is `utf-8`.

1 `<meta charset="utf-8"></meta>`

(the above code snippet online)

Don't you know about character encoding? You can read these resources here:

- * [ASCII](#)
- * [Extended ASCII](#)
- * [UTF-8](#)

Basically, character encoding is the way a character is internally represented so that it can be understood by the computer.

S T O P!

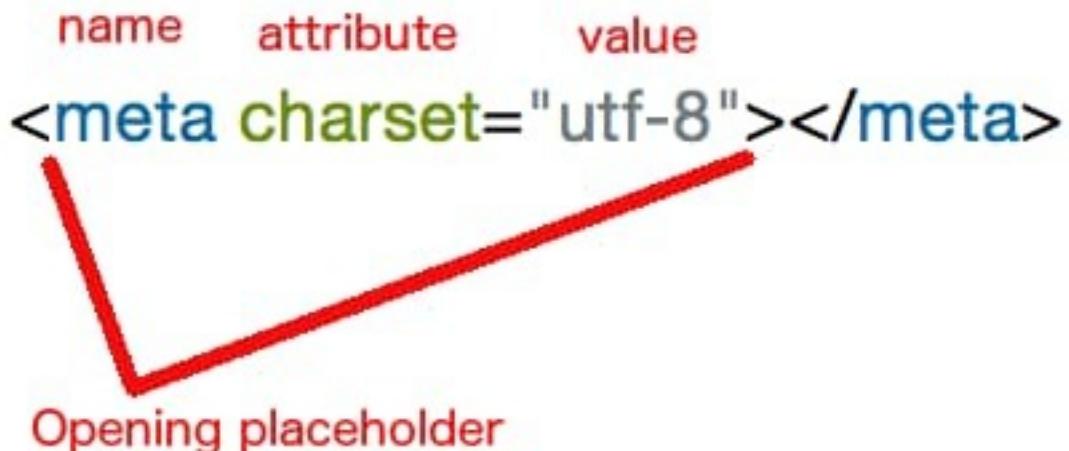
This tag, has the name `meta` but its opening version is not `<meta>` but it has something next to the `meta` word. What is that?

HTML tag attributes

An HTML tag may have an *attribute*.

- An attribute has a name and a value.
- The name is separated from the value using the `=` (equals sign) character.
- The value is enclosed in double quotes (`"`) characters.
- The attribute accompanies the name of the tag and it is placed inside the opening placeholder.
- A tag may have more than one attribute.

So, in the preceding example (`<meta charset="utf-8"></meta>`) the `meta` tag has one attribute that has the name `charset` and the value `utf-8`. The attribute and its value are placed next to the name of the tag, inside the opening placeholder.



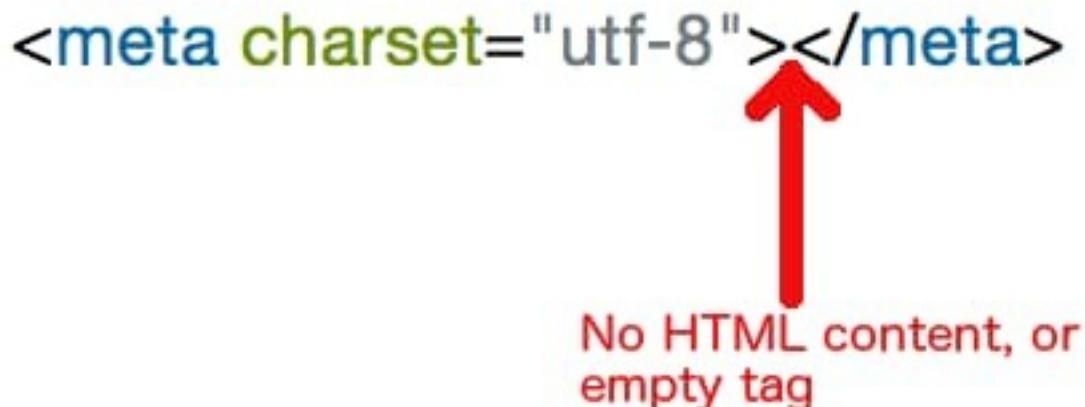
HTML Tag with Attribute" title="HTML Tag with Attribute

Reminder * We have learned about HTML tags that have content, like `html` and `head` tags * We have learned how to open/start them and how to close/end them: `<html>....</html>`. * We have learned that some HTML tags may have attributes, like the `<meta>` tag, which may have the `charset` attribute.

But we are still on the how we can fill in the `<head>` section. And we saw that the first tag that we should be using is the `<meta>` with attribute `charset`. There is another thing that you might have noticed here:

HTML tags without HTML content

The example of the `<meta>` tag, is an example of an HTML tag that does not have HTML content. It only has attributes.



No HTML Content or Empty Tag

You will learn, soon, that there are many HTML tags that do not have actual content. In that case...

Hint You can omit the closing tag for HTML tags that they do not expect to have content. But only for the tags that they do not have content by their definition. Not HTML tags that they may have content but you just leave them empty. In that case you cannot omit the closing tag.

Hence, the better way to write the above `<meta>` is:

1 `<meta charset="utf-8">`

(the above code snippet online)

Perfect! Those were the details behind the first HTML tag that we can put inside the `<head>` section.

This is how our HTML code should look like until now:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6
7      </head>
8
9      <body>
10
11         (here we are going to have the contents of the body section)
12
13     </body>
14 </html>
```

(the above code snippet online)

Let's go to the next tag that can be put inside the `<head>` section.

The title of our page

We are using the HTML tag with name `title` in order to give a title on our page. This title will be displayed on the browser tab. I will show you how. But first let's give a `title` to our page:

```
1  <title>Personal Web Site</title>
```

(the above code snippet online)

This seems to be more like the normal HTML tags. It has the opening and closing tag, and there is a **content**, HTML content inside it. Actually, it is a phrase that plays the role of the title of our HTML page.

Does that tag have any attribute? **No**, it doesn't.

Does that tag have content? **Yes**, it does.

Hence, our HTML code has as follows, until now:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6          <title>Personal Web Site</title>
7
8      </head>
9
10     <body>
```

```
11
12     (here we are going to have the contents of the body section)
13
14     </body>
15 </html>
```

(the above code snippet online)

Indentation again Hey! Did you notice that `<meta>` and `<title>` are indented 2 spaces with regards to the column where `<head>` starts? Yes you did! This is because these two tags are now children of the `<head>` tag. So, they need to be nested, indented, by 2 spaces.

Ok. Nice! We added a `<title>` inside the `<head>` section. What does this affect? Does it have any visual impact? Yes it does. When you open your HTML page, you will see that the browser uses this title as the title of the tab used to display your page.

Here is how:

[Video showing how title tag affects the name of the tab of the page](#)

And that was the minimum that we could add to the `<head>` section of our page. Let's move on now to the `<body>`

Let's fill in the `<body>`

The list of HTML tags that one can use inside the `<body>` section is very long. And whatever we put inside this section it is displayed by the browser, as part of the final HTML page that it displays. Or, at least, the majority of the tags work like that.

Today, we are only going to use the `<h1>` tag that is used to display text using large bold font style. It is the bigger header style text that we can use (at least according to the defaults that come with the browser out of the box).

You can see the `<h1>` example used in our page:

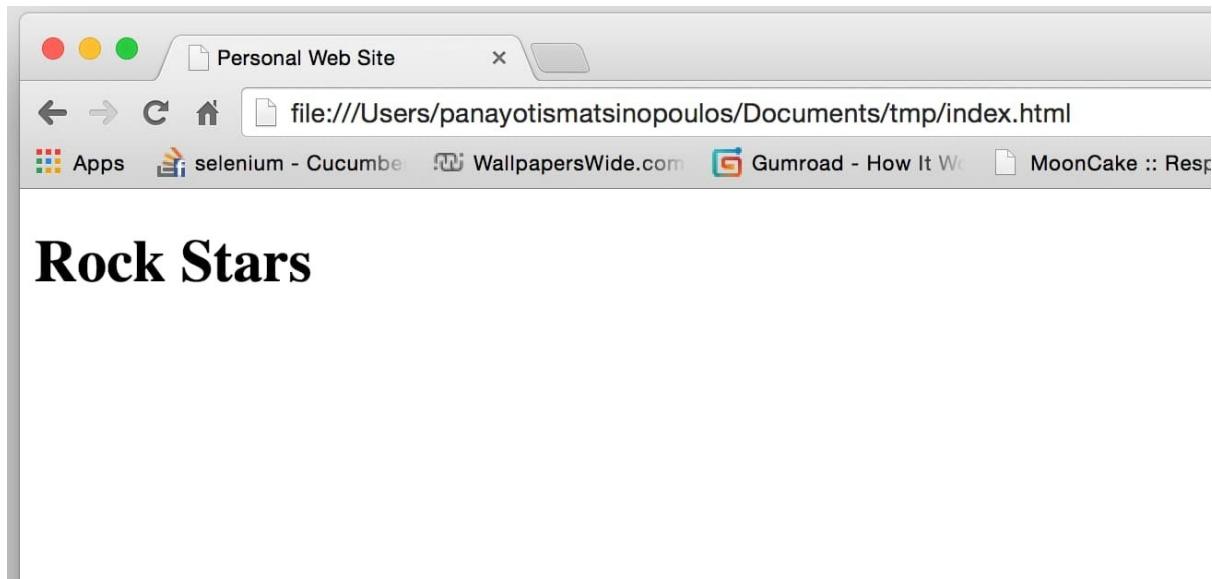
```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6          <title>Personal Web Site</title>
7
8      </head>
9
10     <body>
11
12         <h1>Rock Stars</h1>
```

```
13
14      </body>
15  </html>
```

(the above code snippet online)

As you can see, we have put the `<h1>` inside the `<body>` section. And nested with 2 spaces, since it is a child of `<body>`. Also, the `<h1>` has HTML content and it ends with the corresponding closing tag `</h1>`.

Let's see the final result. Open this document in your browser:



First Rock Stars Page

That's it. We have finished our first HTML page analysis. Next chapters will teach you more HTML tags.

Tasks and Quizzes

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Study very well the content of the chapter and make sure that you have learned and understood everything that is listed in the learning goals.
2. Create the HTML page as described in the chapter.

3. Practice to create this HTML page on a blank document. You need to be able to do that in less than 1 minute.
4. Send your mentor the HTML page that you have created.

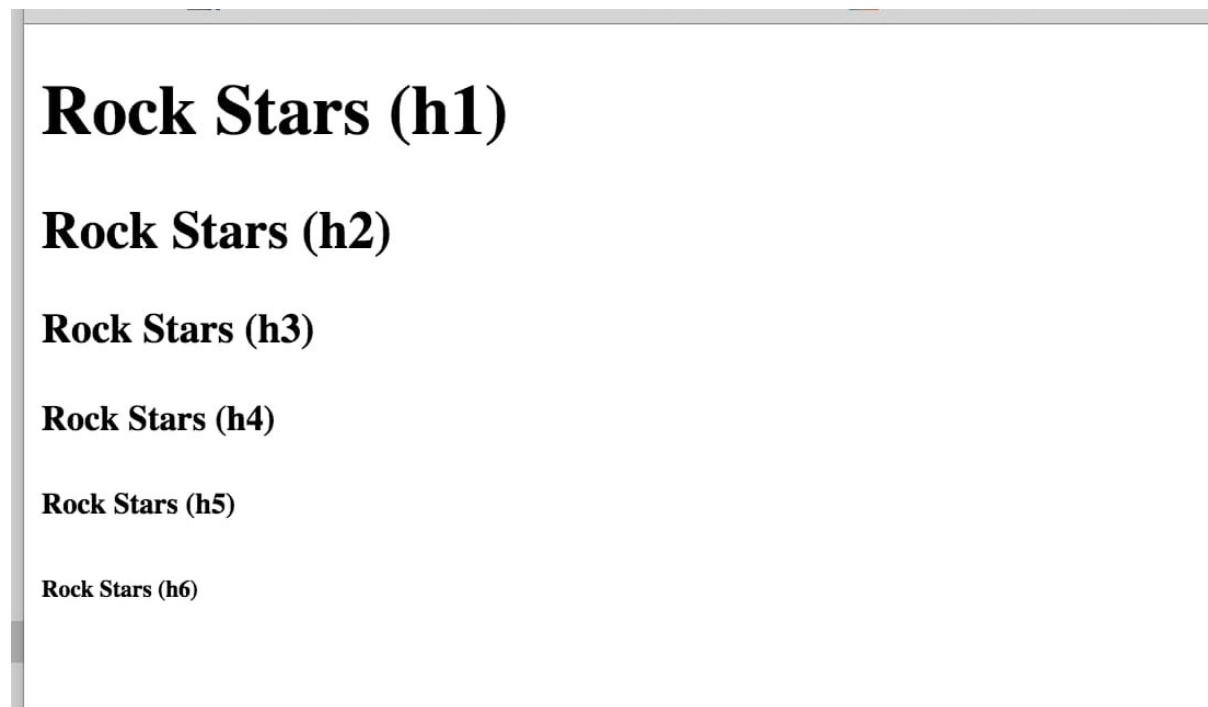
Quiz:

The quiz for this chapter can be found [here](#)

Chapter 5 - Heading Tags

Summary

In this chapter we are learning about the heading tags and generate a Web page like this:



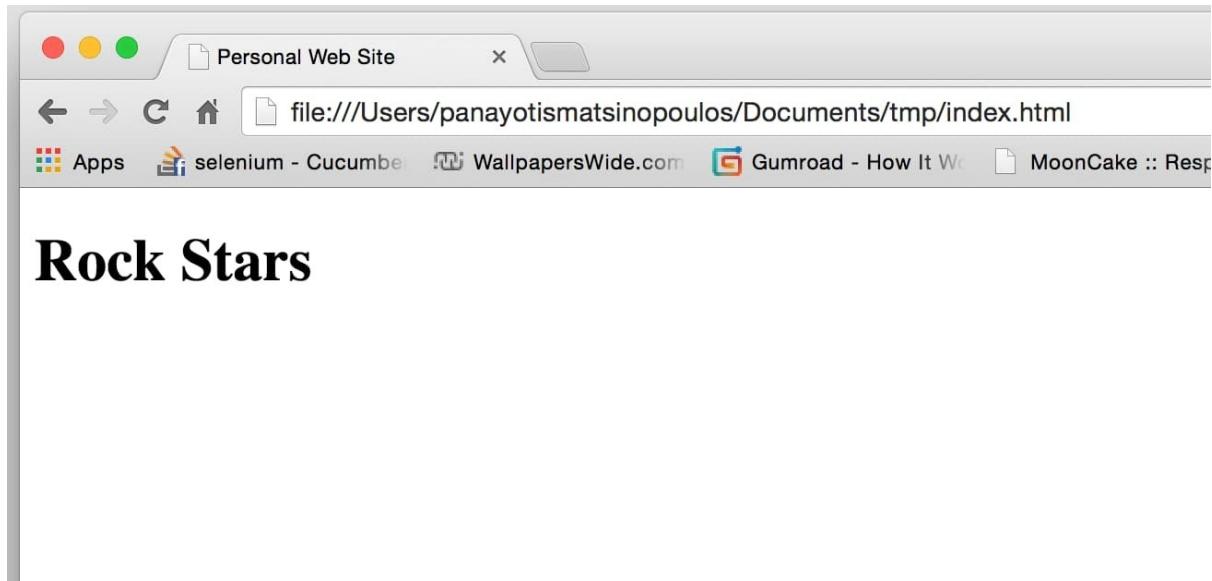
Web Page With Headings

Learning Goals

- Learn about the heading tags `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`.

Heading Tags

We have already learned about the heading tag `<h1>`. It makes text look very big and bold, like here:



First Rock Stars Page

However, you can use smaller heading tags. The one a little bit smaller is `<h2>`, the next smaller one is `<h3>` and you can go up to `<h6>`.

In order to see the different sizes, you can create the following page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6          <title>Personal Web Site</title>
7
8      </head>
9
10     <body>
11
12         <h1>Rock Stars (h1)</h1>
13         <h2>Rock Stars (h2)</h2>
14         <h3>Rock Stars (h3)</h3>
15         <h4>Rock Stars (h4)</h4>
16         <h5>Rock Stars (h5)</h5>
17         <h6>Rock Stars (h6)</h6>
18
19     </body>
20 </html>
```

(the above code snippet online)

So, create the above HTML page and save it with the name that has extension `.html`. Then open that in your browser. You will see something like that:

Rock Stars (h1)

Rock Stars (h2)

Rock Stars (h3)

Rock Stars (h4)

Rock Stars (h5)

Rock Stars (h6)

Heading Tags Web Page

We use the headings to write a heading of a section or paragraph content in our HTML page. We could have created the same visual effect with other tools from the HTML tool set, but this is very handy. Only one HTML tag to type in and then the content that will be styled according to the heading tag.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Write an HTML document that is not the same as the one from the exercise that uses the tags `<h2>` and `<h3>`.
2. Send the HTML document to your mentor.

Chapter 6 - About White Space

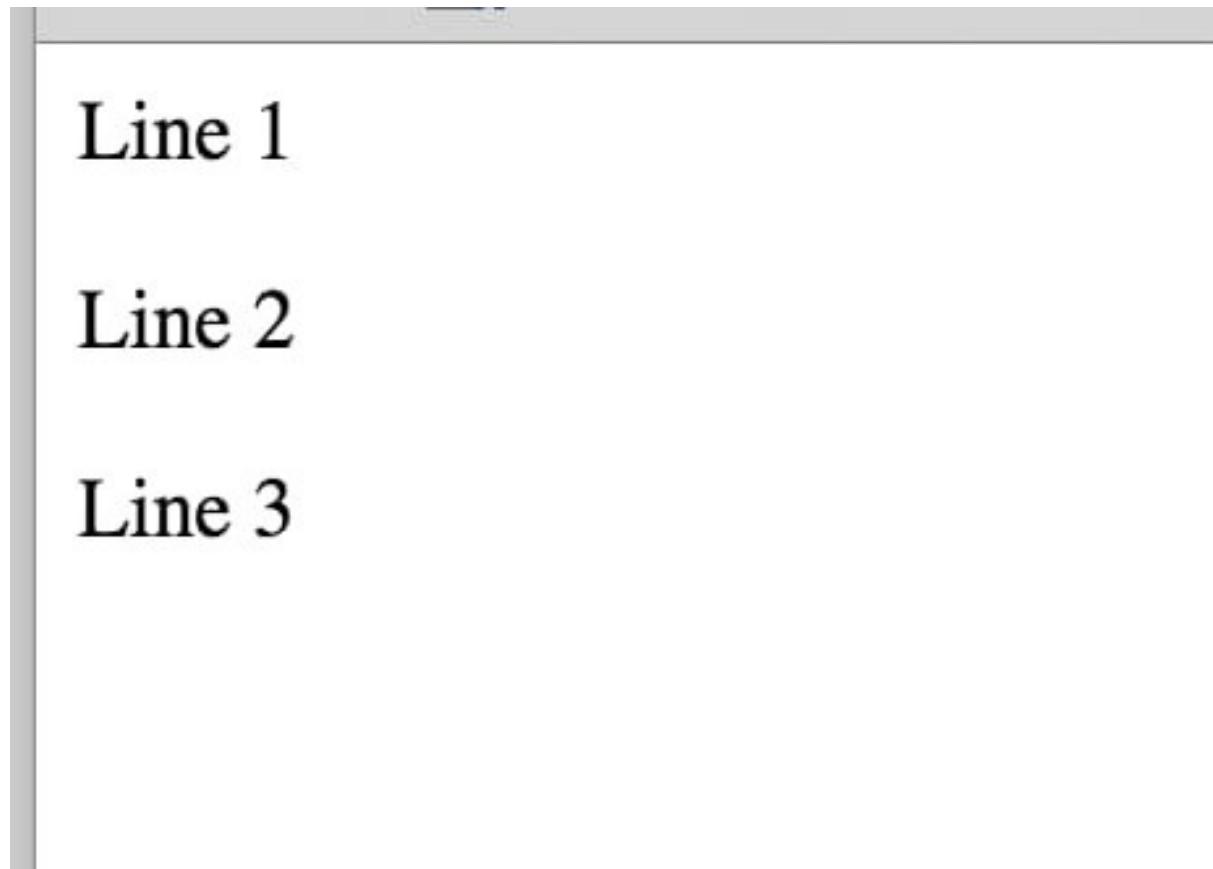
Summary

In this chapter we explain what does white space mean for a browser. For example, how do we create an HTML page with content like the following:

| | | |
|------------------|------------------|------------------|
| Word 1 on Line 1 | Word 2 on Line 1 | Word 3 on line 1 |
| Word 1 on Line 2 | Word 2 on Line 2 | Word 3 on line 2 |
| Word 1 on Line 3 | Word 2 on Line 3 | Word 3 on line 3 |

Line breaks and Multiple White Spaces

or the following:



Lines and One Blank Line in Between

Understanding how browsers deal with white space, blanks, new lines, e.t.c. is very important.

Learning Goals

1. Understand how browser interprets the white space and line breaks inside the HTML document.
2. Learn about the `
` tag.
3. Learn about the `&nbsp`.

About White Space

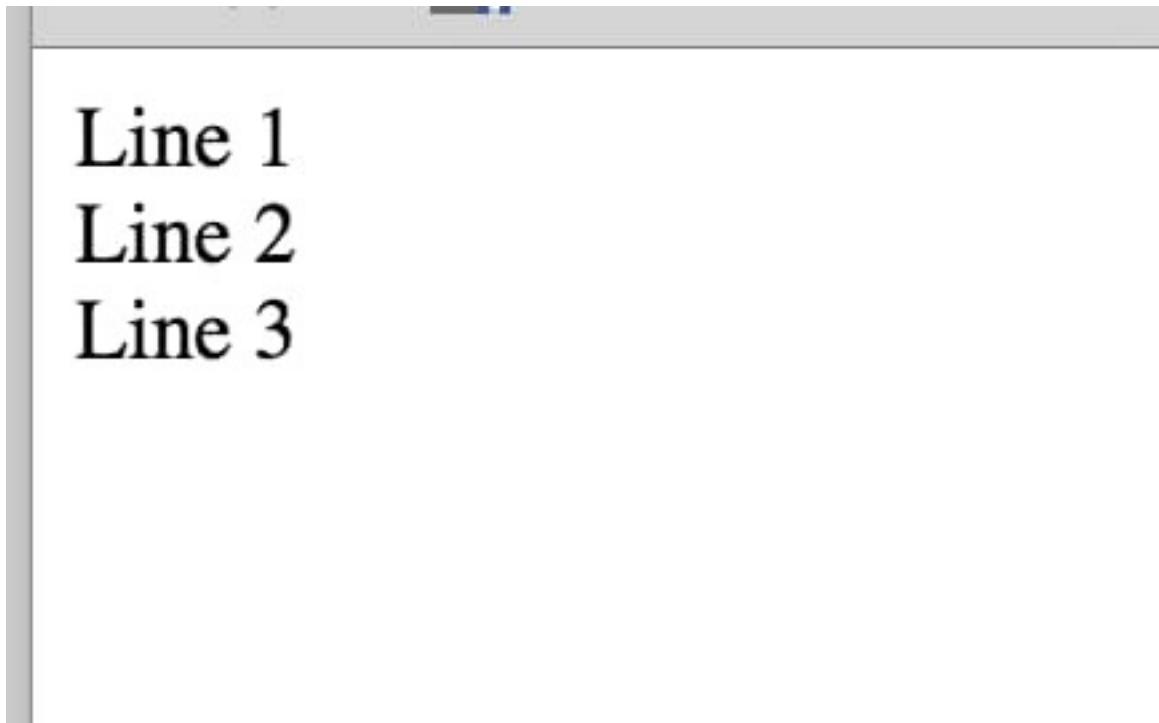
This chapter will teach you how browsers understand the white space that you have inside your HTML documents. It is very important to understand that browsers interpret your big white spaces or new line breaks to a single blank/white space character.

It will be made easier for you if we could do an example. Create a new HTML document with the following content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6          <title>Personal Web Site</title>
7
8      </head>
9
10     <body>
11
12         Line 1
13         Line 2
14         Line 3
15
16     </body>
17 </html>
```

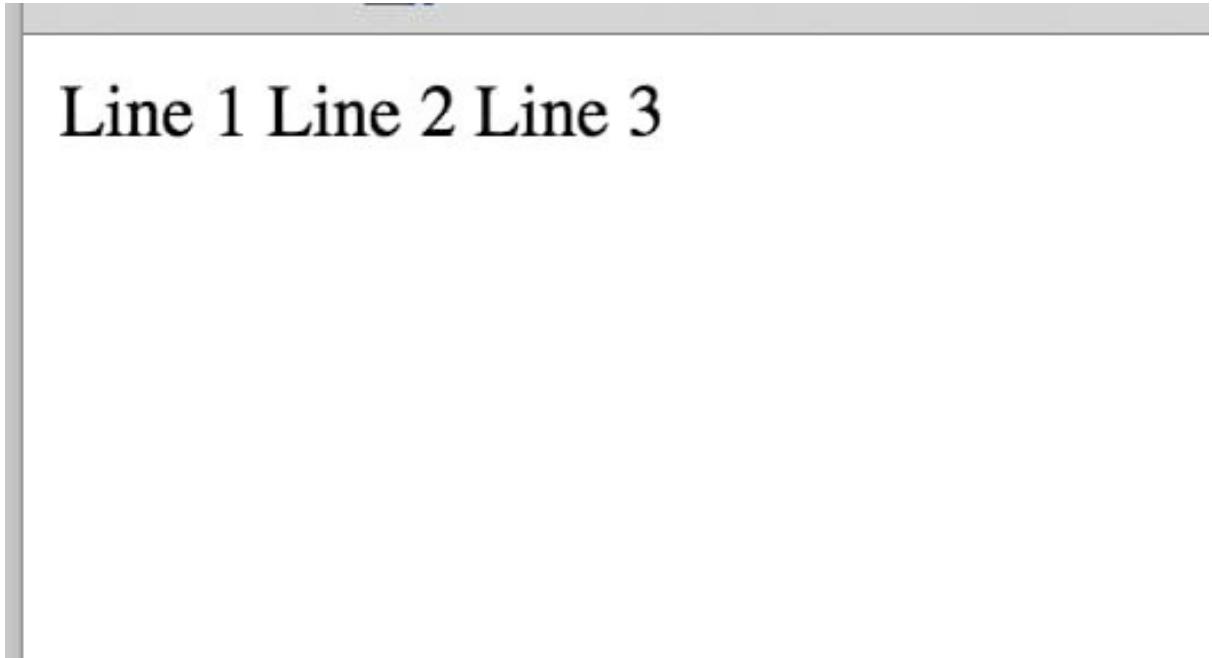
(the above code snippet online)

Nice. Now, you believe that you have created a Web page that will display 3 lines like this:



Three lines one below the other

but the reality will not be with you. The actual content will be displayed like this:



Line 1 Line 2 Line 3

Three lines displayed as one

This is because browser does not understand about the traditional white spaces and new line breaks that we are used to insert when typing a normal document, like in MS Word. Browsers separates printable chunks of text, words, by only 1 space, even if you have plenty of them in between.

Try another example:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4
5      <meta charset="utf-8">
6      <title>Personal Web Site</title>
7
8    </head>
9
10   <body>
11
12     Word 1 on Line 1          Word 2 on Line 1          Word 3 on\
13     line 1
14     Word 1 on Line 2          Word 2 on Line 2          Word 3 on\
15     line 2
16     Word 1 on Line 3          Word 2 on Line 3          Word 3 on\
17     line 3
18
19   </body>
20 </html>
```

(the above code snippet online)

The actual result will be:

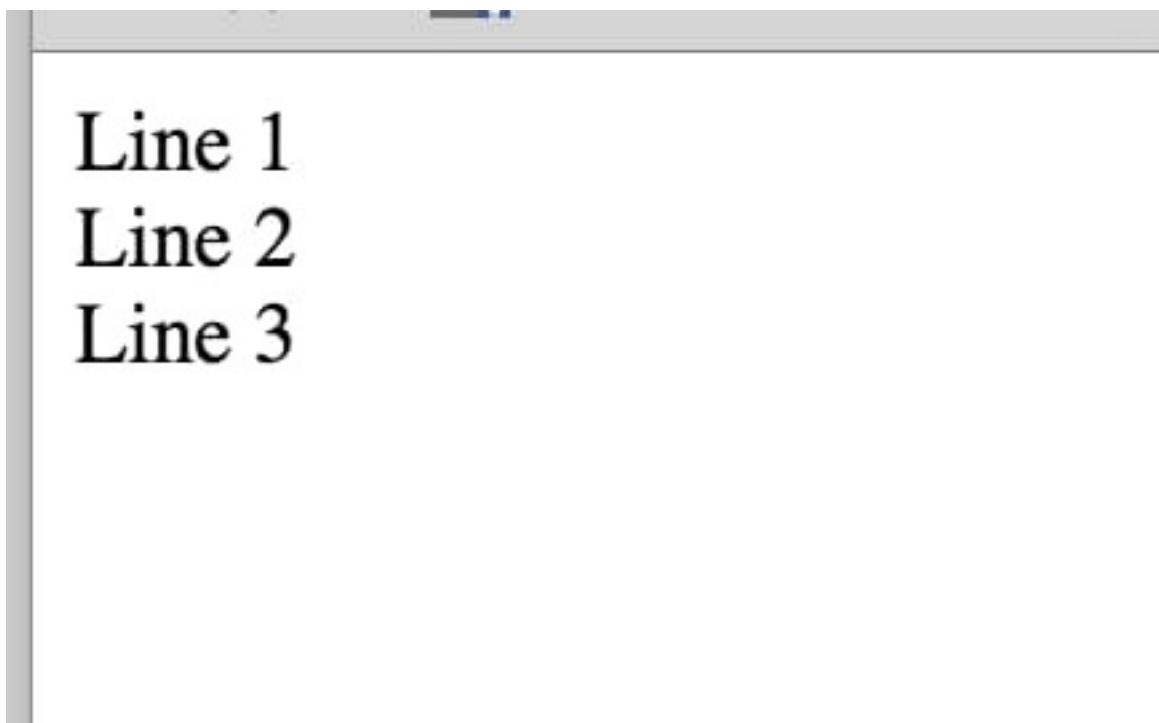
```
Word 1 on Line 1 Word 2 on Line 1 Word 3 on line 1 Word 1 on Line 2 Word 2 on Line 2 Word 3 on line 2 Word 1 on Line 3 Word 2 on Line 3 Word 3 on line 3
```

All in One Line

So, how do we actually create the content the way we want? We will start with the basics, and this is the non-breaking white space and the break HTML tag.

Line break HTML tag

So, you want to create 3 lines, one below the other. Like this:



Three lines one below the other

In order to do that you need to explicitly tell the browser to change line at the end of the lines. For this purpose, we are using the
 tag. So, your HTML document should be like that:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6          <title>Personal Web Site</title>
7
8      </head>
9
10     <body>
11
12         Line 1<br>
13         Line 2<br>
14         Line 3
15
16     </body>
17 </html>
```

(the above code snippet online)

Note:
 tag is a tag that does not have content. Hence, there isn't corresponding closing tag </br>. The tags that do not have closing counterpart, may alternatively be written with their closing angle bracket following the / symbol. For example:
. In other words,
 and
 can be used alternatively.

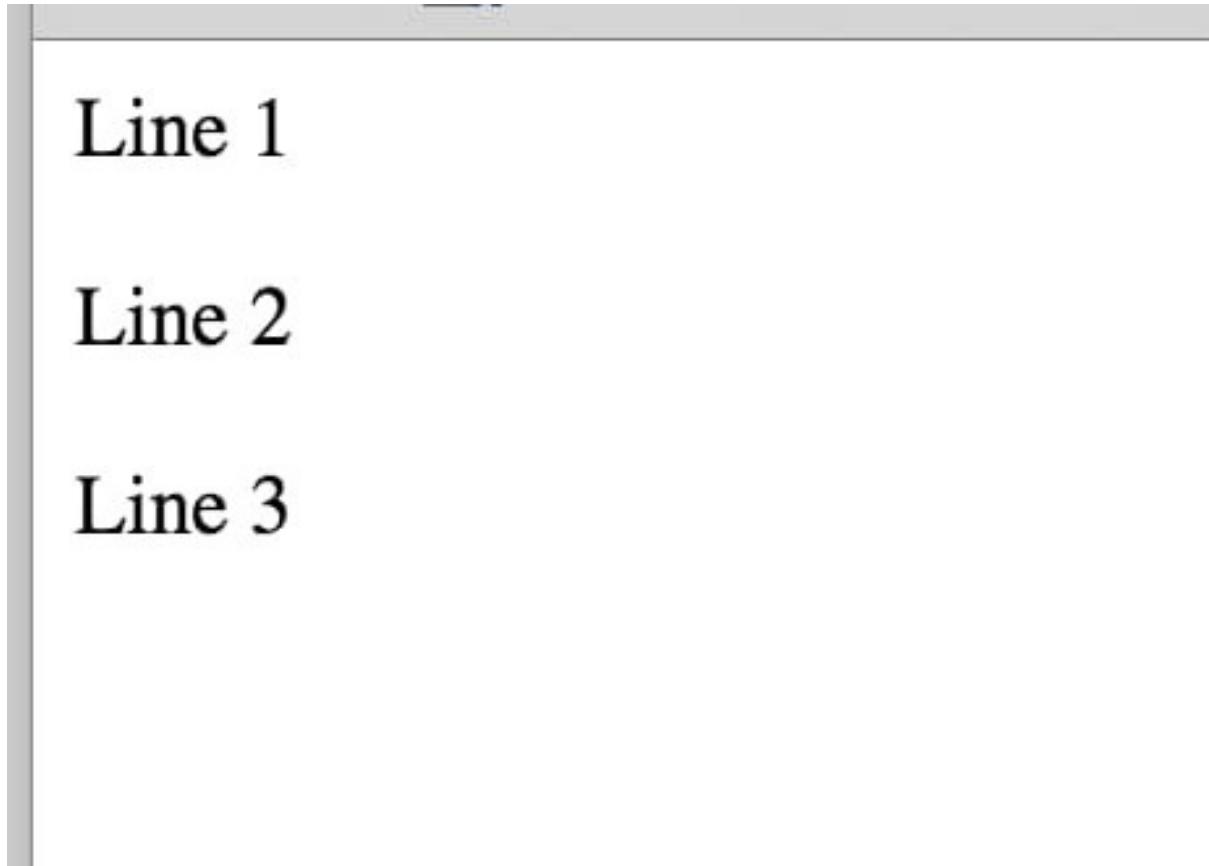
And what if you wanted to leave one empty line in between the lines? Then you just need to give
 alone. Try in your editor the following document and save it as an HTML page. Then open it with your browser.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6          <title>Personal Web Site</title>
7
8      </head>
9
10     <body>
11
12         Line 1<br>
13         <br>
14         Line 2<br>
15         <br>
16         Line 3
17
```

```
18    </body>
19 </html>
```

(the above code snippet online)

This will be the result:



Lines and One Blank Line in Between

White spaces between words

And now, you want to add extra white spaces between words. How can you do that? The answer is that you should use the special sequence of characters: . For each one extra space, you should write one . It is the non breaking space character. Note that the sequence starts with the symbol & and ends with the semi-colon ;. Again, you will need that only if you need more than one blank space. The first one can be the normal blank space character, but the rest, need to be .

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4
5          <meta charset="utf-8">
6          <title>Personal Web Site</title>
7
8      </head>
9
10     <body>
11
12         Word 1 on Line 1 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Word 2\
13         on Line 1 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs\
14         p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Word 3 on line 1<br>
15         Word 1 on Line 2 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Word 2\
16         on Line 2 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs\
17         p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Word 3 on line 2<br>
18         Word 1 on Line 3 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Word 2\
19         on Line 3 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs\
20         p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Word 3 on line 3
21
22     </body>
23 </html>

```

(the above code snippet online)

Create the above document and save it to a file. Then open the file using your browser. The final result will be:

| | | |
|------------------|------------------|------------------|
| Word 1 on Line 1 | Word 2 on Line 1 | Word 3 on line 1 |
| Word 1 on Line 2 | Word 2 on Line 2 | Word 3 on line 2 |
| Word 1 on Line 3 | Word 2 on Line 3 | Word 3 on line 3 |

Line breaks and Multiple White Spaces

You can see how the lines are one on top of the other due to the `
` tags. And you can see that there is enough white space in between the words, exactly like we wanted it, thanks to the `&nbs;`

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create an HTML document that, when displayed using the browser, would look like that:

Letter to Mary

Dear Mary

I am writing to you to let you know that I will be visiting your city next weekend.

So, I was wondering whether you would like to have dinner with me. We can discuss about the new project that I am working on and whether you can offer to me some special price for Internet Marketing services via your company. I know that you are marketing genius!

Looking forward to it!

John Papas
4120 Williams Street
Sun City, AZ 85351
A C M E

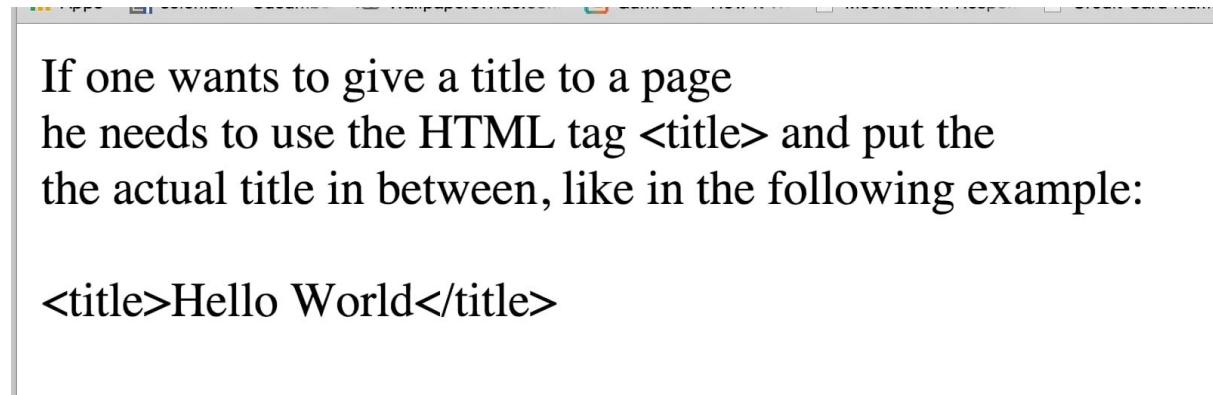
Letter to Mary

2. Send the HTML document to your mentor for review.

Chapter 7 - HTML Entities

Summary

This chapter will teach you about the HTML Entities. These are used to print special characters, including HTML ones, as part of the content of your page. For example, you will find out how you can write an HTML page like this:



You will also learn how you can print characters like these:

| Character |
|-----------|
| ¢ |
| £ |
| ¥ |

or these:

| Character |
|-----------|
| à |
| à |
| â |
| ã |
| Ò |
| Ó |
| Ô |
| Õ |
| À |
| Ã |
| Ѓ |

or even these:

| Character |
|-----------|
| ™ |
| © |
| ® |
| ∀ |
| ∂ |

Learning Goals

- Learn about the HTML Entities
- The most common ones like <, >, &, © e.t.c.

HTML Entities

In the previous chapter you had to introduce the character sequence in order to be able to add extra blank spaces in your text. Actually, this sequence forms an HTML Entity. HTML Entities are sequences of characters that are used when we want to print inside our text characters that are otherwise reserved by the HTML language, or that have special use, or that cannot be directly typed from the keyboard.

Some Common Ones

HTML reserved characters

Suppose that you want to print the character < as part of your text, like in the following image:

Below you can see the less than character:

<

Less than character printed

If you try the following code...:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Printing the less than character</title>
6      </head>
7      <body>
8          Below you can see the less than character:<br/>
9          <
10     </body>
11 </html>
```

(the above code snippet online)

...it will work. However, you need to know that the character < is an HTML reserved character and it shouldn't be actually used in the text of your page for other reason than to start an HTML tag. And it may work in this simple example above, but, in more complex documents, if you start using this character literally, you might experience errors on your page and it might be difficult for you to find out where the bug is.

Hence, you need to start getting used to it: Whenever you want to use the character < as part of your text, in the payload of your page, then you need to actually use the corresponding HTML entity: < (from less than).

Important * All HTML entities start with the symbol & * All HTML entities end with the symbol ;

So, the correct content of the above HTML page should be:

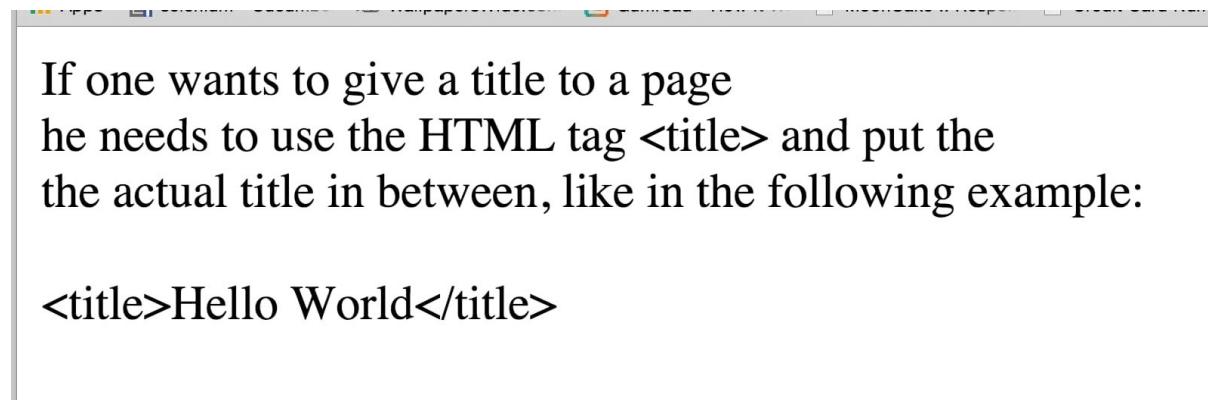
```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Printing the less than character</title>
6      </head>
7      <body>
8          Below you can see the less than character:<br/>
9          &lt;
10     </body>
11 </html>
```

(the above code snippet online)

Same goes for the reserved character >. You need to use the > instead.

Here is a small exercise for you. With the knowledge that you have acquired so far, how would you produce the following?



Prompt: Spend sometime trying to do it alone, before reading the solution below

If you try to do it like that:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Printing the less than character</title>
6      </head>
7      <body>
8          If one wants to give a title to a page<br/>
9          he needs to use the HTML tag <title> and put the<br/>
10         the actual title in between, like in the following example:<br/>
11         <br/>
12         <title>Hello World</title>
13     </body>
14 </html>
```

(the above code snippet online)

the result will be this:

If one wants to give a title to a page he needs to use the HTML tag

HTML code incorrectly put inside text

It will not be the one you want. That bad result will be, if you don't use HTML entities for the reserved characters < and >.

Go ahead and change the reserved HTML characters that exist inside your actual page payload. Make them be the corresponding HTML entities. Here is the correct HTML code:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Printing the less than character</title>
6    </head>
7    <body>
8      If one wants to give a title to a page<br/>
9      he needs to use the HTML tag &lt;title&gt; and put the<br/>
10     the actual title in between, like in the following example:<br/>
11     <br/>
12     &lt;title&gt;Hello World&lt;/title&gt;
13   </body>
14 </html>
```

(the above code snippet online)

Common HTML Entities for HTML reserved characters

Here is a table with the common HTML reserved characters and their corresponding HTML Entities:

| Character | Entity |
|-----------|--------|
| &nbsp | |
| < | < |
| > | > |
| & | & |

How can you literally print the HTML Entity* without being translated to its corresponding symbol?

Suppose that you want to produce the following outcome:

The HTML Entity for the symbol < is <

Printing the HTML Entity name itself

In order to achieve that, you first need to understand how many and which HTML reserved characters this page has. Can you think about it?

It has 2: the < and the &. Hence, these are the 2 characters that need to be replaced with their corresponding HTML Entity equivalent, in order to be able to produce the required result.

Hence, here is the actual HTML code for this page:

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <title>Literally Print HTML Entity</title>
6    </head>
7    <body>
8      The HTML Entity for the symbol &lt; is &lt;;
9    </body>
10   </html>

```

(the above code snippet online)

Try to write this on your HTML editor, save it and open it. Does it produce the desired result?

As we said before, except from the HTML reserved characters, you may need to use HTML Entities for special characters and symbols. Here are some of them:

Currencies

| Character | Entity |
|-----------|---------|
| ¢ | ¢ |
| £ | £ |
| ¥ | ¥ |
| € | € |

Special Alphabet Characters

Note: Some of the HTML Entities below are used to accent a character. And instead of a name (like: which has the name nbsp) we use numbers. Example: ̀ All the HTML Entities can be expressed with numbers, but it is easier for us to remember the name rather than the number, especially for the ones that we use more frequently. However, there are some HTML Entities that only have a number and not a name. In that case we cannot avoid using the number. And if we do, we need to prefix the number with the characters sequence: &#.

Also, you can see that the ̀ means accent on the character that it follows. So, if this one follows the character a, then it will print a. If it follows the character A it will print A

| Character | Entity |
|-----------|---------|
| à | à |
| à | á |
| â | â |
| ã | ã |
| Ò | Ò |
| Ó | Ó |
| Ô | Ô |
| Õ | Õ |
| A | Α |
| B | Β |
| Γ | Γ |

Other Symbols

| Character | Entity |
|-----------|----------|
| ™ | ™ |
| © | © |
| ® | ® |
| ∀ | ∀ |
| ∂ | ∂ |

Full References

The above tables are only indicative. If you want to have access to the full HTML Entities, you only have to google for that.

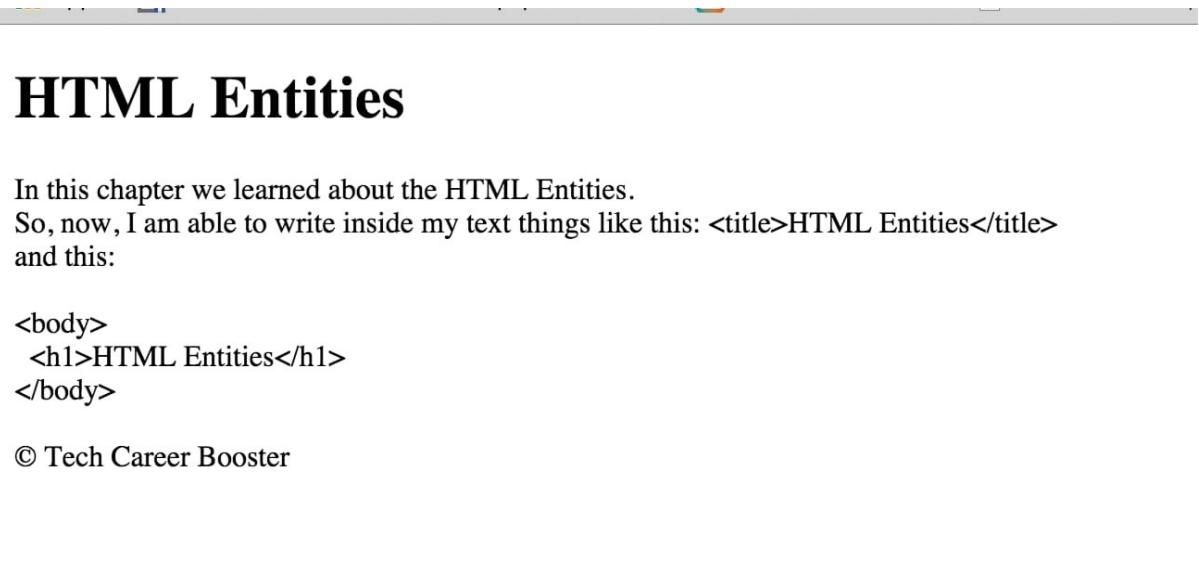
Tasks & Quizzes

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Write an HTML page that, when displayed on browser, will be looking like the following:



HTML Entities

In this chapter we learned about the HTML Entities.

So, now, I am able to write inside my text things like this: <title>HTML Entities</title> and this:

```
<body>
  <h1>HTML Entities</h1>
</body>
```

© Tech Career Booster

Task for HTML Entities

2. Make sure that you have the correct <code>head</code> with a title included.
3. Send the HTML document of this page to your mentor for review.

Quiz:

The quiz for this chapter can be found [here](#)

Chapter 8 - Paragraphs, Horizontal Lines And Comments

Summary

In this chapter we are going to learn about

- paragraphs (`<p>`)
- horizontal lines (`<hr/>`) and
- HTML comments (`<!-- ... -->`)

You will be able to organize your text into paragraphs and use horizontal lines like this:

The screenshot shows a web browser window displaying a page titled "Of Mice And Men". Below the title, a section is labeled "Extract 1". The text within the extract is as follows:

Curley stepped over to Lennie like a terrier. 'What the hell you laughin' at?'
Lennie looked blankly at him. 'Huh?'
Then Curley's rage exploded. 'Come on, ya big bastard. Get up on your feet. No big son-of-a-bitch is gonna laugh at me. I'll show you who's yella!'
Lennie looked helplessly at George, and then he got up and tried to retreat. Curley was balanced and poised. He slashed at Lennie with his left, and then smashed down his nose with a right. Lennie gave a cry of terror. Blood welled from his nose. 'George,' he cried. 'Make 'um let me alone, George.' He backed until he was against the wall, and Curley followed, slugging him in the face. Lennie's hands remained at his sides; he was too frightened to defend himself.
George was on his feet yelling, 'Get him, Lennie. Don't let him do it.'
Lennie covered his face with his huge paws and bled with terror. He cried, 'Make 'um stop, George!' Then Curley attacked his stomach and cut off his wind.

...

Paragraphs and Horizontal Lines

Learning Goals

- Learn about HTML paragraphs: `<p>...</p>`
- Learn about HTML horizontal rules/line: `<hr/>`
- Learn about HTML comments: `<!-- ... -->`

Paragraphs, Horizontal Lines and Comments

Paragraphs

We have learned how we break a line and we start a new one. Do you remember how? It is the tag `
` or `
`. However, usually, we let browser create lines automatically, based on the width of the browser window. And we only care about separating one paragraph from the other.

So, basically, when we write the text of a page, we divide it into paragraphs using the paragraph `<p>` tag. We start a paragraph with `<p>` and we end it with `</p>`.

Browser, at the end of each paragraph, changes line to start printing the content that follows.

Try the following document:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Simple Paragraph Document</title>
6      </head>
7      <body>
8
9          This is text before the paragraph starting tag.<p>This is the first line of t\
10 he paragraph.</p>This is text after the paragraph closing tag.
11
12      </body>
13
14 </html>
```

(the above code snippet online)

How do you think this page will appear on the browser? Do you see that the `<p>` and `</p>` are on the same line as the rest of the text inside the `<body>` content? Will that appear as we have written that above? No. It will appear like this:

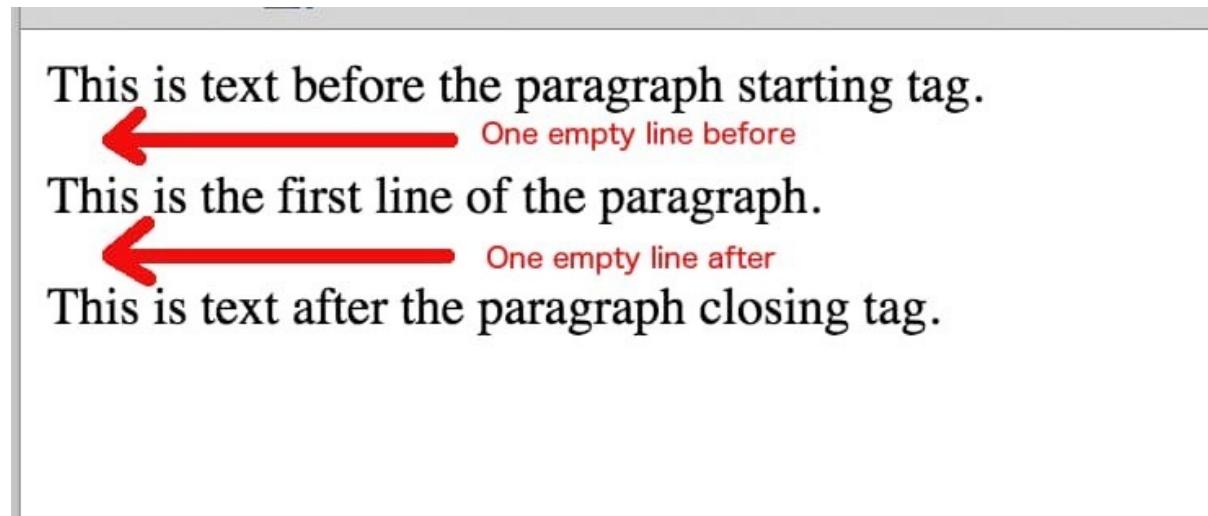
This is text before the paragraph starting tag.

This is the first line of the paragraph.

This is text after the paragraph closing tag.

First Paragraph Example

As you can see, browser creates an empty line before the start of the paragraph and another empty line exactly after the end of it:



First Paragraph Example - One Line Before, One Line After

The whole width of the browser window is occupied by the paragraph text. Text is automatically wrapped to the next line if cannot fit the width of the window. Try the following document:

Hint: If you want a dummy text generator to copy & paste in your HTML pages you can try searching Google for `lorem ipsum` text generator. You will find plenty of them. This is how we got the text for the paragraph below.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Paragraph with long text</title>
6      </head>
7      <body>
8
9          This is text before the paragraph starting tag.
10
11         <p>
12             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\por incididunt ut labore
13                 et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation \
14                 ullamco laboris nisi ut
15                     aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in \
16                     voluptate velit esse cillum
17                         dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proiect,
18                         sunt in culpa qui officia
19                             deserunt mollit anim id est laborum.
20
21         </p>
22
23         This is text after the paragraph closing tag.
```

```
24  
25    </body>  
26  
27  </html>
```

(the above code snippet online)

Open this page on your browser and see how the text of the paragraph flows from the left to the right side of the browser window. The text wraps automatically to the next line when there is not enough space for the text to appear on the same line. Try to resize the width of your browser and make it more narrow. See how browser wraps and creates more lines for your text to appear properly.

Like this, which is the text on wide window:

This is text before the paragraph starting tag.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
This is text after the paragraph closing tag.

Paragraph text on wide window

And like this, which is the same text on narrow window:

This is text before the paragraph starting tag.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
This is text after the paragraph closing tag.

Paragraph text on narrow window

Note how the text is wrapped irrespectively of how you have broken the lines in your HTML text. If you really want to break the lines at specific points, and still be on the same paragraph, you need to use `
` at the points you want the breaks to appear. Here is the same example as above but with a line break on the first sentence after

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Paragraph with long text</title>
6      </head>
7      <body>
8
9          This is text before the paragraph starting tag.
10
11         <p>
12             Lorem ipsum dolor sit amet,<br/>
13             consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore
14             et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation \
15             ullamco laboris nisi ut
16                 aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in \
17             voluptate velit esse cillum
18                 dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proiect\
19             dent, sunt in culpa qui officia
20                 deserunt mollit anim id est laborum.
21         </p>
22
23         This is text after the paragraph closing tag.
24
25     </body>
26
27 </html>
```

(the above code snippet online)

Save this document and load it on your browser. Do you see that browser changes line on `amet`, `point`?

Horizontal Lines

This is nice and easy. The tag `<hr/>` creates a horizontal line (rule) that extends from left to right, for the whole width of the browser. This is useful to separate your text in sections.

Let's assume that we want a horizontal line below the `Lorem ipsum` paragraph on the page above. It needs to be done like this:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Paragraph with long text</title>
6      </head>
7      <body>
8
9          This is text before the paragraph starting tag.
10
11         <p>
12             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\por incididunt ut labore
13                 et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation \ullamco laboris nisi ut
14                     aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in \
15                     voluptate velit esse cillum
16                         dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proiect, sunt in culpa qui officia
17                             deserunt mollit anim id est laborum.
18                     </p>
19
20         <hr/>
21
22
23         <body>
24
25             This is text after the paragraph closing tag.
26
27     </body>
28
29 </html>
```

(the above code snippet online)

This will be displayed like this:

This is text before the paragraph starting tag.

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proiect, sunt in culpa qui officia deserunt mollit anim id est laborum.

This is text after the paragraph closing tag.

Paragraph with horizontal line

Do you see the horizontal gray line below the paragraph text and before the last sentence?

Comments

We will finish this chapter by describing how we can write text inside the body of our page and make sure that it is not parsed as HTML code. It will only be there to work as an explanation text to the reader of the HTML code rather than to the reader of the web page itself. The construct is called **comments** and the actual tool that we use to create them is the following:

```
1 <!-- these are the comments -->
```

(the above code snippet online)

We start with the character sequence `<!--` and we put our comments next to that. Then we finish our comments with the character sequence `-->`. We can put whatever we like inside this block. And as long text as we like. It will not be interpreted as HTML code and the browser will essentially ignore it.

We use comments when we want to give some explanations to the HTML source code reader, to another developer that might read our HTML source code. Or as a reminder to ourselves, if we happen to open the HTML source of the page and read quite some time after we initially wrote it.

Hence, comments work like documentation of our work and it is very helpful when we want to keep notes of things that will make easier for us or other person to understand what we have done.

Here is an example HTML page with comments used.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Paragraph with long text</title>
6   </head>
7   <body>
8
9     <h1>My fantastic essay</h1>
10
11    <!-- In the first paragraph I will give the introduction to my essay
12        It will be a short paragraph, no more than 3-4 sentences -->
13    <p>
14      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\por incididunt ut labore
15      et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation \
16      ullamco laboris nisi ut
17      aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in \
18      voluptate velit esse cillum
19      dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proi\
20      dent, sunt in culpa qui officia
```

```
22     deserunt mollit anim id est laborum.  
23 </p>  
24  
25     <!-- Then I will write the main body of my essay. it will be a series of para\  
26 graphs that will  
27     explain my subject in many details. -->  
28 <p>  
29     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\  
30 por incididunt ut labore  
31     et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation \  
32 ullamco laboris nisi ut  
33     aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in \  
34 voluptate velit esse cillum  
35     dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proi\  
36 dent, sunt in culpa qui officia  
37     deserunt mollit anim id est laborum.  
38 </p>  
39  
40 <p>  
41     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\  
42 por incididunt ut labore  
43     et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation \  
44 ullamco laboris nisi ut  
45     aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in \  
46 voluptate velit esse cillum  
47     dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proi\  
48 dent, sunt in culpa qui officia  
49     deserunt mollit anim id est laborum.  
50 </p>  
51  
52 <p>  
53     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\  
54 por incididunt ut labore  
55     et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation \  
56 ullamco laboris nisi ut  
57     aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in \  
58 voluptate velit esse cillum  
59     dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proi\  
60 dent, sunt in culpa qui officia  
61     deserunt mollit anim id est laborum.  
62 </p>  
63  
64     <!-- I will close my essay with a short closing message -->  
65 <p>  
66     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\  
67 por incididunt ut labore
```

```

68      et dolore magna aliqua.
69      </p>
70
71      <hr/>
72
73      </body>
74
75  </html>
```

(the above code snippet online)

And if you see this page on the browser, you will see something like this:

My fantastic essay

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Paragraph with horizontal line and comments

As you can experience yourself, the comments are not visible.

Important Comments may not be visible. This does not mean that the reader cannot see them. He can see them if he chooses to view the source of your page. See this video below on how one can do that.

[Video showing how one can see the source code of the HTML page and read the HTML comments](#)

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create the following HTML page.

The screenshot shows a Microsoft Word document window. The title 'Of Mice And Men' is at the top in a large, bold, black serif font. Below it, the section 'Extract 1' is in a bold, black sans-serif font. The text is in a black serif font. It contains several paragraphs of text from the novel, starting with Curley's speech and Lennie's response, followed by Curley's rage and Lennie's reaction, and ending with George's intervention. There is a horizontal line at the bottom of the page.

Of Mice And Men

Extract 1

Curley stepped over to Lennie like a terrier. 'What the hell you laughin' at?'

Lennie looked blankly at him. 'Huh?'

Then Curley's rage exploded. 'Come on, ya big bastard. Get up on your feet. No big son-of-a-bitch is gonna laugh at me. I'll show you who's yella.'

Lennie looked helplessly at George, and then he got up and tried to retreat. Curley was balanced and poised. He slashed at Lennie with his left, and then smashed down his nose with a right. Lennie gave a cry of terror. Blood welled from his nose. 'George,' he cried. 'Make 'um let me alone, George.' He backed until he was against the wall, and Curley followed, slugging him in the face. Lennie's hands remained at his sides; he was too frightened to defend himself.

George was on his feet yelling, 'Get him, Lennie. Don't let him do it.'

Lennie covered his face with his huge paws and bleated with terror. He cried, 'Make 'um stop, George.' Then Curley attacked his stomach and cut off his wind.

...

Task for Paragraphs and Horizontal Lines

2. Note that the title of the page needs to be "Of Mice and Men".
3. Note that there is a horizontal line at the bottom. Please, include that too.
4. Send the HTML document to your mentor for review.

Chapter 9 - HTML Lists

Summary

In this chapter we are going to talk about

- Lists
- Ordered Lists and
- Nested Lists

We will be able to create content like this:

Planning my Trip

1. Find City
2. Find Hotel
3. Pick up Destinations
 - Don't forget:
 - Museums
 - Squares
 - Theaters
 - Natural Places
4. Book Flight

Web Page With Lists

Learning Goals

- Learn about unordered lists
- Learn about ordered lists
- Learn about nested lists

Lists

Those that are aware of word processing applications, like MS Word or Pages, they know to create lists, either bullet lists or number lists.

For example, this is a bullet list:

- Milk
- Wine
- Coffee
- Sugar

And this one is a number list:

1. Milk
2. Wine
3. Coffee
4. Sugar

The first list type, in HTML parlance, is called *unordered list* or *ul*. The second list type is called *ordered list* or *ol*. And the items within the lists are called *list items* or *li*.

Unordered Lists

Having said that, in order to create an unordered list in HTML, you need to use the HTML tag ``. This has an opening `` tag and a closing `` tag and in between we put the list items using the HTML tag ``. List items have opening `` and closing tags, ``. In between we write the content of the corresponding list item.

Here is our first HTML page unordered list example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Unordered List Example</title>
6      </head>
7
8      <body>
9          <h1>This is my first unordered list example</h1>
10
11         <ul>
12             <li>Milk</li>
13             <li>Wine</li>
14             <li>Coffee</li>
15             <li>Sugar</li>
```

```
16    </ul>
17  </body>
18
19 </html>
```

(the above code snippet online)

Type the above into an HTML document, save it and open it on your browser. You should see an HTML page like this:

This is my first unordered list example

- Milk
- Wine
- Coffee
- Sugar

Unordered List Example

Ordered Lists

In order to created an ordered list, you just use the `` tag instead of the ``. Hence the previous list, if turned to ordered list would have been:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Ordered List Example</title>
6    </head>
7
8    <body>
9      <h1>This is my first ordered list example</h1>
10
11      <ol>
12        <li>Milk</li>
13        <li>Wine</li>
14        <li>Coffee</li>
15        <li>Sugar</li>
```

```
16      </ol>
17  </body>
18
19 </html>
```

(the above code snippet online)

and the result of displaying this HTML page on the browser would have been something like this:

This is my first ordered list example

1. Milk
2. Wine
3. Coffee
4. Sugar

Ordered List Example

Nested Lists

I guess that this was pretty straightforward until this point. But, what happens is you want to do something more advanced like this:

This is my first unordered list example

- Milk
- Wine
- Coffee
- Sugar
- Some Optional:
 - Cheese
 - Pizza
 - Butter
 - Chips

Nested Unordered List Example

As you can inspect on the picture, the last item of the list includes an extra, another list of items. This is possible, because, you can almost write any valid HTML content inside each list item. But when you write another list, this is drawn using a different bullet than the nesting / parent list. So, the nesting parent list uses a black-filled circle, whereas the nested list is drawn with a white-filled circle bullet.

Here is the HTML code that creates the above result:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Unordered List Example</title>
6      </head>
7
8      <body>
9          <h1>This is my first unordered list example</h1>
10
11         <ul>
12             <li>Milk</li>
13             <li>Wine</li>
14             <li>Coffee</li>
15             <li>Sugar</li>
16             <li>
17                 Some Optional:
18                 <ul>
19                     <li>Cheese</li>
20                     <li>Pizza</li>
21                     <li>Butter</li>
22                     <li>Chips</li>
23                 </ul>
24             </li>
25         </ul>
26     </body>
27
28 </html>
```

(the above code snippet online)

Same works for ordered lists and nesting. An, on top of that, you can mix unordered with ordered lists. Why don't you experiment?

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech

Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create the following HTML page.

Planning my Trip

1. Find City
2. Find Hotel
3. Pick up Destinations

Don't forget:

- Museums
- Squares
- Theaters
- Natural Places

4. Book Flight

Task for Lists

2. The title of the page should be “My Trip Plan”
3. Send the HTML document to your mentor for review.

Chapter 10 - HTML Tables

Summary

In this chapter we are talking about tables. Tables are used to organize our data in tabular format. We will be able to implement content like this:

| | | | | |
|---------------|----------------------|---------------------------------|--------------|---------|
| TCB | Order no: | #ABC001 | | |
| | Order Date: | 23-Mar-2016 | | |
| | Customer | | | |
| | Name: | John Papas | | |
| | Address: | Independence Day 5th str, 11511 | | |
| | Order Details | | | |
| | 1 | Of Mice and Men | Book | 10.00€ |
| | 2 | Les Miserables | Book | 12.00€ |
| | 3 | Game of Thrones - S01 | DVD | 50.00€ |
| | 4 | Samsung Galaxy | Mobile Phone | 200.00€ |
| Total: | | | | 272.00€ |

Web Page with Table

Learning Goals

- Learn about the `<table>` tag.
- Learn about the `<thead>` tag.
- Learn about the `<tbody>` tag.
- Learn about the `<tr>` tag.
- Learn about the `<td>` tag.
- Learn about the `<th>` tag.
- Learn about the `border` attribute.
- Learn about the `cellspacing` attribute.
- Learn about the `cellpadding` attribute.

- Learn about the `align` attribute.
- Learn about the `colspan` attribute.
- Learn about the `attribute.`
- Learn how you can approach the design of complex table structures.

Tables

A table is another tool that we can use to organize the text inside our HTML page. It is used to present tabular data.

Important: Tables are used to present tabular data. Not to divide your HTML page to columns and rows. Dividing your HTML page to columns and rows is done with the `<div>` tag and it is presented in later chapters.

This is an example of an HTML table:

| Name | Email | Phone | Mobile |
|-----------------|-------------------------------|----------------|----------------|
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| Douglas Hayes | douglas.hayes92@example.com | (231)-391-6269 | (790)-838-2130 |
| Cameron Brown | cameron.brown32@example.com | (204)-488-5204 | (508)-463-6811 |
| Nevaeh Diaz | nevaeh.diaz99@example.com | (436)-578-2946 | (906)-412-3302 |
| Kathy Miller | kathy.miller62@example.com | (724)-705-3555 | (764)-841-2531 |
| Susan King | susan.king88@example.com | (774)-205-7754 | (639)-267-9728 |
| Jeffery Ramirez | jeffery.ramirez83@example.com | (723)-243-7706 | (172)-597-3422 |
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |

Table Example

But let's start from the basics:

Starting and closing tags

We start a table with the starting tag `<table>` and we close it with the tag `</table>`. Pretty standard I guess.

```

1 <table>
2
3 </table>
```

(the above code snippet online)

Inside that, we put the columns and the rows of the table.

Rows of the table

Each table is composed of rows. The rows are the `<tr>` (table row) tags. They have content and they close with `</tr>`. Note that the only descendant elements of a table row are table columns. You should not put anything else inside the content of a `<tr>`.

Here we are. A table with 5 rows.

```

1 <table>
2   <tr>    </tr>
3   <tr>    </tr>
4   <tr>    </tr>
5   <tr>    </tr>
6   <tr>    </tr>
7 </table>
```

(the above code snippet online)

Columns of the table

Each row of a table is composed of a series of columns (or cells). The columns can contain almost any valid HTML content that is useful to present the content of the cell/column. The contents of a cell are enclosed within the starting `<td>` (t table data) and closing `</td>` tags.

And here we are. We now have all the information necessary to build our table (but not enough for styling it).

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Demo Table</title>
6   </head>
7   <body>
8     <table>
9       <tr><td>Name</td><td>Email</td><td>Phone</td><td>Mobile</td></tr>
10      <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332-5\</td>
11      385</td><td>(888)-677-3719</td></tr>
12      <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-224\</td>
```

```

13 7</td><td>(216)-889-4933</td></tr>
14      <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213-3\ 
15 499</td><td>(207)-516-4474</td></tr>
16      <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782</td\ 
17 d><td>(565)-627-3002</td></tr>
18      </table>
19  </body>
20 </html>

```

(the above code snippet online)

Type the above into an HTML document, save it and open it in your browser. The result will be something like this:

| Name | Email | Phone | Mobile |
|--------------|----------------------------|----------------|----------------|
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |

Paragraph text on wide window

This is a good start. We are not at the final result, but this is a good start.

Let's make some small improvements

Table Head and Table Body

Usually, we want to tell the browser which part of the table is the head and which part of the table is the body of it. We do that by using the tags `<thead>` and `<tbody>`. So, in order to specify the table head, we enclose the corresponding rows inside `<thead>...</thead>` section, and in order to specify the body, we enclose the corresponding rows inside `<tbody>...</tbody>` section. Let's do this improvement in our table example and see how things change.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Head and Body</title>
6      </head>
7      <body>
8          <table>
9              <thead>
10                 <tr><td>Name</td><td>Email</td><td>Phone</td><td>Mobile</td></tr>
11             </thead>

```

```

12
13      <tbody>
14          <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\
15 -5385</td><td>(888)-677-3719</td></tr>
16          <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\
17 247</td><td>(216)-889-4933</td></tr>
18          <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\
19 -3499</td><td>(207)-516-4474</td></tr>
20          <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\
21 /td><td>(565)-627-3002</td></tr>
22      </tbody>
23  </table>
24  </body>
25 </html>
```

(the above code snippet online)

If you view this page on your browser, you will see that browsers do not treat head differently from body, and the result will be exactly the same as before. However, I would like you to keep that in mind, that when you separate the head from the body you have a way to easier style the whole table, with the head using a style different from the body. This will be taught when we start learning how to style our HTML page, later on in this section.

Head Cells

One improvement that does have a visual effect on the table is the use of the tag `<th>` instead of `<td>`. This tag, `<th>` is to be used for cells that hold data that we need to be treated as headers of column.

Let's do this small improvement on our example table.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Head and Body</title>
6      </head>
7      <body>
8          <table>
9              <thead>
10                 <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11             </thead>
12
13             <tbody>
14                 <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\
15 -5385</td><td>(888)-677-3719</td></tr>
16                 <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\
```

```

17 247</td><td>(216)-889-4933</td></tr>
18      <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\
19 -3499</td><td>(207)-516-4474</td></tr>
20      <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\
21 /td><td>(565)-627-3002</td></tr>
22      </tbody>
23  </table>
24  </body>
25 </html>

```

(the above code snippet online)

What we did, was to change the Name, Email, Phone and Mobile column headers to be inside a `<th>` tag, instead of a `<td>` tag. See the visual result here:

| Name | Email | Phone | Mobile |
|--------------|----------------------------|----------------|----------------|
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |

Table with th

As you can see, the `<th>` tags are specially treated by the browser. It makes the content bold and centered. Nice.

Let's do one more improvement.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Head and Body</title>
6      </head>
7      <body>
8          <table border="1">
9              <thead>
10                 <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11             </thead>
12
13             <tbody>
14                 <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\
15 -5385</td><td>(888)-677-3719</td></tr>
16                 <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\

```

```

17 247</td><td>(216)-889-4933</td></tr>
18      <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\
19 -3499</td><td>(207)-516-4474</td></tr>
20      <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\
21 /td><td>(565)-627-3002</td></tr>
22      </tbody>
23  </table>
24  </body>
25 </html>

```

(the above code snippet online)

We only changed the line #8 in the code above by adding the attribute `border` with value `1`. This does the following: It creates a border around the table and around each one of the cells of the table. Here it is:

| Name | Email | Phone | Mobile |
|--------------|----------------------------|----------------|----------------|
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |

Table with border

Note that if you increase the border value from `1` to `3` for example, you will increase the thickness of the border that surrounds the table but not the thickness of the borders of the cells. You can try that out and see.

But let's do one more improvement.

Cell spacing

As you can see, there is some free / blank space between the border of cells. This is because there is default cell spacing applied when a table is drawn by the browser. You can remove that by applying the attribute `cellspacing` with value `0`.

Hence, the new code will be:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Head and Body</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0">
9              <thead>
10                 <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11             </thead>
12
13             <tbody>
14                 <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\-
15 -5385</td><td>(888)-677-3719</td></tr>
16                 <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\-
17 247</td><td>(216)-889-4933</td></tr>
18                 <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\-
19 -3499</td><td>(207)-516-4474</td></tr>
20                 <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\-
21 /td><td>(565)-627-3002</td></tr>
22             </tbody>
23         </table>
24     </body>
25 </html>

```

(the above code snippet online)

The final result of displaying this page will be:

| Name | Email | Phone | Mobile |
|--------------|----------------------------|----------------|----------------|
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |

Table with no cell spacing

Do you see how cell spacing has disappeared? This makes the border look nicer too, because the border of one cell is drawn exactly next to the border of the neighbouring cell.

Let's do one more improvement.

Cell padding

As you can see, our current table example has one styling flaw. The content of the cells are very tight, and very close to the borders of the cells. There is no room for the content to breath. We usually want to have some free space around the text of the cells. This can be achieved using the attribute `cellpadding`. Let's give it a try:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Cell Padding</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="5">
9              <thead>
10                 <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11             </thead>
12
13             <tbody>
14                 <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\-
15 -5385</td><td>(888)-677-3719</td></tr>
16                 <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\-
17 247</td><td>(216)-889-4933</td></tr>
18                 <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\-
19 -3499</td><td>(207)-516-4474</td></tr>
20                 <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\-
21 /td><td>(565)-627-3002</td></tr>
22             </tbody>
23         </table>
24     </body>
25 </html>
```

(the above code snippet online)

We have added the `cellpadding="5"` attribute value pair on the `<table>` tag. If we see this page on our browser, it will be something like this:

| Name | Email | Phone | Mobile |
|--------------|----------------------------|----------------|----------------|
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |

Table with cell padding

As you can see, there is some breathing space between the text inside the cells and the cells borders. You can increase this space if you increase the value of the `cellpadding` attribute.

Column Span

There are cases in which some rows of the table do not have the same number of columns like all the others. Let's say for example that we want to have a row, at the top of the table, that has the value "Customers".

One might think that adding a row with only 1 cell would do the trick:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Cell Padding</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <thead>
10                 <tr><th>Customers</th></tr>
11                 <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
12             </thead>
13
14             <tbody>
15                 <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\-
16 -5385</td><td>(888)-677-3719</td></tr>
17                 <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\-
18 247</td><td>(216)-889-4933</td></tr>
19                 <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\-
20 -3499</td><td>(207)-516-4474</td></tr>
21                 <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\-
22 /td><td>(565)-627-3002</td></tr>
```

```

23      </tbody>
24  </table>
25 </body>
26 </html>

```

(the above code snippet online)

We have added the line: `<tr><th>Customers</th></tr>` as the first row inside the `<thead>` of our table. Let's see the visual result:

| Customers | | | |
|--------------|----------------------------|----------------|----------------|
| Name | Email | Phone | Mobile |
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |

Paragraph text on wide window

The problem is the border of the first top cell. The right side of it. It shouldn't be there. One quick solution to that is to tell table that the particular cell does not occupy only 1 column, but occupies all the columns of the table. In our case 4 columns. This is done with the attribute `colspan`. So, we need to give the attribute `colspan` with value 4 on the "Customers" cell:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Demo Table With Colspan</title>
6    </head>
7    <body>
8      <table border="1" cellspacing="0" cellpadding="8">
9        <thead>
10       <tr><th colspan="4">Customers</th></tr>
11       <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
12     </thead>
13
14     <tbody>
15       <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\-
16 -5385</td><td>(888)-677-3719</td></tr>
17       <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\-

```

```

18 247</td><td>(216)-889-4933</td></tr>
19      <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\
20 -3499</td><td>(207)-516-4474</td></tr>
21      <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\
22 /td><td>(565)-627-3002</td></tr>
23      </tbody>
24  </table>
25  </body>
26 </html>

```

(the above code snippet online)

And the visual result of that will be:

| Customers | | | |
|------------------|----------------------------|----------------|----------------|
| Name | Email | Phone | Mobile |
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |

Table with Colspan Example

Things have become much better. Haven't they? As you can see the "Customers" cell is treated as one. Also, being a `<th>` cell makes sure that is centered and bold.

Aligning

Now let's suppose that we want to print the number of customers as the last row of our table. Probably we would do something like that:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Number of Customers</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <thead>
10                 <tr><th colspan="4">Customers</th></tr>
11                 <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
12             </thead>
13
14             <tbody>
15                 <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\-
16 -5385</td><td>(888)-677-3719</td></tr>
17                 <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\-
18 247</td><td>(216)-889-4933</td></tr>
19                 <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\-
20 -3499</td><td>(207)-516-4474</td></tr>
21                 <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\-
22 /td><td>(565)-627-3002</td></tr>
23                 <tr><td colspan="4">Number of Customers: 4</td></tr>
24             </tbody>
25         </table>
26     </body>
27 </html>

```

(the above code snippet online)

What we did, actually, was to add one more row in the `<tbody>` section of our table. The row we added was:

```
1  <tr><td colspan="4">Number of Customers: 4</td></tr>
```

and create a cell that spans 4 columns. The visual result is the following:

| Customers | | | |
|------------------------|----------------------------|----------------|----------------|
| Name | Email | Phone | Mobile |
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| Number of Customers: 4 | | | |

Table with number of customers

We will improve that by making the “Number of Customers” to be bold. How can we do that according to what we have learned so far? `<th>`. Yes. We can use this type of cell, even if we are not on the `<thead>`. We are going to change the row for “Number of Customers” to be:

```
1 <tr><th colspan="3">Number of Customers:</th><td>4</td></tr>
```

Watch out. We do not only change the `<td>` to `<th>` but we also make sure that we put in two different cells the heading “Number of Customers:” and the actual value. Also, we change the `colspan` now to be 3 instead of 4, since the 4th column is now occupied by the new cell for the actual value of the number of customers.

So, the actual code now becomes:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Demo Table With Number of Customers and Th</title>
6   </head>
7   <body>
8     <table border="1" cellspacing="0" cellpadding="8">
9       <thead>
10         <tr><th colspan="4">Customers</th></tr>
11         <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
12       </thead>
13
14       <tbody>
15         <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\-
16 -5385</td><td>(888)-677-3719</td></tr>
17         <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\-
```

```

18 247</td><td>(216)-889-4933</td></tr>
19      <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\
20 -3499</td><td>(207)-516-4474</td></tr>
21      <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\
22 /td><td>(565)-627-3002</td></tr>
23      <tr><th colspan="3">Number of Customers:</th><td>4</td></tr>
24  </tbody>
25  </table>
26  </body>
27 </html>

```

(the above code snippet online)

And the visual result is this:

| Customers | | | |
|-----------------------------|----------------------------|----------------|----------------|
| Name | Email | Phone | Mobile |
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| Number of Customers: | | | 4 |

Table with Number of Customer on Th

So far so good. But, what we want here is to change the alignment of these two new cells. We want the content “Number of Customers:” to be aligned to the right, and the content “4”, the actual number of customers, to be aligned in the center of their cells. You can do that using the `align` attribute with the values `right` and `center` respectively.

So, the line with the number of customers changes to:

```

1 <tr><th colspan="3" align="right">Number of Customers:</th><td align="center">4<\
2 td></tr>

```

and the whole HTML document becomes:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Number of Customers Aligned</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <thead>
10                 <tr><th colspan="4">Customers</th></tr>
11                 <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
12             </thead>
13
14             <tbody>
15                 <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332\-
16 -5385</td><td>(888)-677-3719</td></tr>
17                 <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2\-
18 247</td><td>(216)-889-4933</td></tr>
19                 <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213\-
20 -3499</td><td>(207)-516-4474</td></tr>
21                 <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782<\-
22 /td><td>(565)-627-3002</td></tr>
23                 <tr><th colspan="3" align="right">Number of Customers:</th><th align="center">4</th></tr>
24             </tbody>
25         </table>
26     </body>
27 </html>
```

(the above code snippet online)

The visual result of which is:

| Customers | | | |
|-----------------------------|----------------------------|----------------|----------------|
| Name | Email | Phone | Mobile |
| Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| Number of Customers: | | | 4 |

Align Table Cells

We have managed to bring the table content to a level that we want, using simple HTML tags and attributes.

Row span

Let's do one more improvement to our table. Suppose that we want to achieve the following result:

| TCB | Customers | | | |
|-----|-----------------------------|----------------------------|----------------|----------------|
| | Name | Email | Phone | Mobile |
| | Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| | Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| | Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| | Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| | Number of Customers: | | | 4 |

Paragraph text on wide window

We want to add the name of our company as the left most column. In order to be able to design tables like these you need to answer first the following questions:

1. Does my table have table head and table body? Clearly separated?

2. How many rows does my table have?
3. How many columns does my table have?

Let's answer these questions for this table that we want to design.

1. Does my table have table head and table body, **clearly defined and separated**? The answer here is **NO**. There are not rows that can be put in either of the sections, head or body. We cannot say, for example, that the first 2 rows go to head and the next 5 go to body. This is because there is the column with the "TCB" content, on the left that does not allow me to make this distinction. Is this a problem? No. It is not. We will not separate the rows to head and body. However, I can definitely use `<th>` for cells that work as column or data headers/names. All the bold ones in the example table that I am trying to build here.
2. How many rows does my table have? Count the maximum number of rows that you see scanning the table from top to bottom. On our example this is 7.
3. How many columns does my table have? Count the maximum number of columns that you see scanning the table from left to right. This is 5 in our example. The columns with contents "TCB", "Name", "Email", "Phone", "Mobile".

You now know how to start laying out your table:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Row Span</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <tr><td></td><td></td><td></td><td></td><td></td></tr>
10             <tr><td></td><td></td><td></td><td></td><td></td></tr>
11             <tr><td></td><td></td><td></td><td></td><td></td></tr>
12             <tr><td></td><td></td><td></td><td></td><td></td></tr>
13             <tr><td></td><td></td><td></td><td></td><td></td></tr>
14             <tr><td></td><td></td><td></td><td></td><td></td></tr>
15             <tr><td></td><td></td><td></td><td></td><td></td></tr>
16         </table>
17     </body>
18 </html>

```

(the above code snippet online)

As you can see above, I have a table without head, neither body, with 7 rows, 5 columns each.

Now, I am looking at the rows, from top to bottom and I change my design accordingly.

The first row has 2 columns:



| Customers | | | | |
|------------------|-----------------------------|----------------------------|----------------|----------------|
| TCB | Name | Email | Phone | Mobile |
| | Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| | Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| | Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| | Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| | Number of Customers: | | | 4 |

Table First Row

As you can see, the 1st of the 2 columns occupies 1 column size, but the 2nd occupies the rest of the table width. Hence, we need to set the colspan for the 2nd column to be 4. (Maximum number of columns 5 - 1 for the first, equals 4). I also use `<th>` to make the content bold.

Let's put that in the code:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Row Span</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <tr><th>TCB</th><th colspan="4">Customers</th></tr>
10             <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
11             <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
12             <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
13             <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
14             <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
15             <tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
16         </table>
17     </body>
18 </html>

```

(the above code snippet online)

The second row has 5 columns:

| Customers | | | | |
|----------------------|--------------|----------------------------|----------------|----------------|
| | Name ✓ | Email ✓ | Phone ✓ | Mobile ✓ |
| TCB | Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| | Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| | Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| | Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| Number of Customers: | | | | 4 |

Table 2nd row

Since the maximum number of columns in the table is 5, then we do not need any `colspan` attribute to any of the cells. We will also write `<th>` to make the content bold. Let's write the 2nd row:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Row Span</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <tr><th>TCB</th><th colspan="4">Customers</th></tr>
10             <tr><th>TCB</th><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11         r>
12             <tr><td></td><td></td><td></td><td></td><td></td></tr>
13             <tr><td></td><td></td><td></td><td></td><td></td></tr>
14             <tr><td></td><td></td><td></td><td></td><td></td></tr>
15             <tr><td></td><td></td><td></td><td></td><td></td></tr>
16             <tr><td></td><td></td><td></td><td></td><td></td></tr>
17         </table>
18     </body>
19 </html>

```

(the above code snippet online)

We proceed similarly for the 3rd, 4th, 5th and 6th rows of the table. We do not use `<th>` but we use `<td>` instead. Here is how we add these rows:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Row Span</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <tr><th>TCB</th><th colspan="4">Customers</th></tr>
10             <tr><th>TCB</th><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11         r>
12             <tr><th>TCB</th><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332-5385</td><td>(888)-677-3719</td></tr>
13             <tr><th>TCB</th><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2247</td><td>(216)-889-4933</td></tr>
14             <tr><th>TCB</th><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213-3499</td><td>(207)-516-4474</td></tr>
15             <tr><th>TCB</th><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782</td><td>(565)-627-3002</td></tr>
16             <tr><td></td><td></td><td></td><td></td><td></td></tr>
17         </table>
18     </body>
19 </html>

```

(the above code snippet online)

The last, 7th, row has 3 cells:

| Customers | | | | |
|-----------|--------------|----------------------------|----------------------|----------------|
| | Name | Email | Phone | Mobile |
| TCB | Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| | Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| | Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| | Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| ✓ | ✓ | | Number of Customers: | ✓ 4 |

Table Seventh Row

As you can see, the 2nd column (Number of Customers:) occupies 3 columns width. This means that we will need colspan with value 3 for this column.

The code now becomes:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Row Span</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <tr><th>TCB</th><th colspan="4">Customers</th></tr>
10             <tr><th>TCB</th><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11         <r>
12             <tr><th>TCB</th><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332-5385</td><td>(888)-677-3719</td></tr>
13             <tr><th>TCB</th><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-2247</td><td>(216)-889-4933</td></tr>
14             <tr><th>TCB</th><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213-3499</td><td>(207)-516-4474</td></tr>
15             <tr><th>TCB</th><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782</td><td>(565)-627-3002</td></tr>
16             <tr><th>TCB</th><th colspan="3" align="right">Number of Customers:</th><td align="center">4</td></tr>
17         </table>
18     </body>
19 </html>

```

(the above code snippet online)

Are we finished? Let's see the visual result of that. Save this document into a file and open it on your browser. You will see this:

| | | Customers | | |
|------------|-----------------------------|----------------------------|----------------|----------------|
| TCB | Name | Email | Phone | Mobile |
| TCB | Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| TCB | Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| TCB | Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| TCB | Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| TCB | Number of Customers: | | | 4 |

Table with repeated TCB

The problem is obvious. We repeated the TCB column but we do not really need it. We only need

it for the first row although it occupies the 1st column on the other rows too. This is achieved with the `rowspan` attribute which needs to have the value equal to the number of rows that a column occupies. In our particular case, “TCB” column starts from top and goes to bottom, hence `rowspan` needs to be with value 7.

So, let's remove the `<th>TCB</th>` from all the rows except the first one, and let's add the attribute `rowspan` on the `<th>TCB</th>` of the first row.

The document becomes:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Row Span</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <tr><th rowspan="7">TCB</th><th colspan="4">Customers</th></tr>
10             <tr><th>Name</th><th>Email</th><th>Phone</th><th>Mobile</th></tr>
11
12             <tr><td>Gary Coleman</td><td>gary.coleman21@example.com</td><td>(398)-332-5\ 
13 385</td><td>(888)-677-3719</td></tr>
14             <tr><td>Rose Parker</td><td>rose.parker16@example.com</td><td>(293)-873-224\ 
15 7</td><td>(216)-889-4933</td></tr>
16             <tr><td>Chloe Nelson</td><td>chloe.nelson18@example.com</td><td>(957)-213-3\ 
17 499</td><td>(207)-516-4474</td></tr>
18             <tr><td>Eric Bell</td><td>eric.bell16@example.com</td><td>(897)-762-9782</td\ 
19 d><td>(565)-627-3002</td></tr>
20             <tr><th colspan="3" align="right">Number of Customers:</th><td align="center\ 
21 r">4</td></tr>
22         </table>
23     </body>
24 </html>

```

(the above code snippet online)

This one, if you save it and load on a browser, you will have the result that you really want:

| Customers | | | | |
|------------------|-----------------------------|----------------------------|----------------|----------------|
| TCB | Name | Email | Phone | Mobile |
| | Gary Coleman | gary.coleman21@example.com | (398)-332-5385 | (888)-677-3719 |
| | Rose Parker | rose.parker16@example.com | (293)-873-2247 | (216)-889-4933 |
| | Chloe Nelson | chloe.nelson18@example.com | (957)-213-3499 | (207)-516-4474 |
| | Eric Bell | eric.bell16@example.com | (897)-762-9782 | (565)-627-3002 |
| | Number of Customers: | | | 4 |

Paragraph text on wide window

Finally, you may have noticed that the HTML code is not easy to read. In that case, when the tables start to become more complex, we decide to use nesting and indentation as follows:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Demo Table With Row Span</title>
6      </head>
7      <body>
8          <table border="1" cellspacing="0" cellpadding="8">
9              <tr>
10                 <th rowspan="7">TCB</th>
11                 <th colspan="4">Customers</th>
12             </tr>
13
14             <tr>
15                 <th>Name</th>
16                 <th>Email</th>
17                 <th>Phone</th>
18                 <th>Mobile</th>
19             </tr>
20
21             <tr>
22                 <td>Gary Coleman</td>
23                 <td>gary.coleman21@example.com</td>
24                 <td>(398)-332-5385</td>
25                 <td>(888)-677-3719</td>
26             </tr>
27

```

```
28     <tr>
29         <td>Rose Parker</td>
30         <td>rose.parker16@example.com</td>
31         <td>(293)-873-2247</td>
32         <td>(216)-889-4933</td>
33     </tr>
34
35     <tr>
36         <td>Chloe Nelson</td>
37         <td>chloe.nelson18@example.com</td>
38         <td>(957)-213-3499</td>
39         <td>(207)-516-4474</td>
40     </tr>
41
42     <tr>
43         <td>Eric Bell</td>
44         <td>eric.bell16@example.com</td>
45         <td>(897)-762-9782</td>
46         <td>(565)-627-3002</td>
47     </tr>
48
49     <tr>
50         <th colspan="3" align="right">Number of Customers:</th>
51         <td align="center">4</td>
52     </tr>
53
54 </table>
55 </body>
56 </html>
```

(the above code snippet online)

Styling tables

It is good to know how you can do basic styling of a table using the above techniques. However, when you will start learning CSS, you will prefer to use CSS rules in order to do that.

Tables and Emails

Note that tables, as we said at the beginning, are not to be used to divide the document in sections. It is the existence of the `<div>` tag that serves this need. However, there is an exception to this rule. HTML Emails. HTML emails, are emails that have rich content and they look like a normal HTML page. The problem is that email clients do not treat well HTML emails that are using `<div>` tags but they treat very well HTML emails that are using `<tables>`. Have that in mind when you start designing HTML email templates. We will see that later in another chapter.

Tasks & Quizzes

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

1. Create an HTML page that would like this:

| | | | | |
|---------------|----------------------|---------------------------------|--------------|---------|
| TCB | Order no: | #ABC001 | | |
| | Order Date: | 23-Mar-2016 | | |
| | Customer | | | |
| | Name: | John Papas | | |
| | Address: | Independence Day 5th str, 11511 | | |
| | Order Details | | | |
| | 1 | Of Mice and Men | Book | 10.00€ |
| | 2 | Les Miserables | Book | 12.00€ |
| | 3 | Game of Thrones - S01 | DVD | 50.00€ |
| | 4 | Samsung Galaxy | Mobile Phone | 200.00€ |
| Total: | | | | 272.00€ |

Task With Tables

2. Make sure that you use cell padding equal to 8

Quiz:

The quiz for this chapter can be found [here](#)

Chapter 11 - First Simple CSS Rules

Summary

In this chapter we start using CSS, Cascading Style Sheets. CSS is the tool that will allow us to style our HTML pages and make them looking nice to the visitor.

We will start by implementing something very simple like this:

This is my first task on CSS

CSS - WEB Page

In later chapters, you will learn a lot of techniques to make your pages looking really good.

Learning Goals

- Learn basic CSS properties, like `color`, `background-color`, `font-size`, `border`.
- Learn how we can style an HTML element by using the `style` attribute.
- Learn about the `<style>` tag.
- Learn about the `<link>` tag and how we use it to reference external CSS files.

First Simple CSS Rules

Before we continue with HTML code and more advanced HTML tags, I will ask you to have a break on core HTML and deal with little CSS. We will start styling our documents with various fonts and with colors.

Style Rules and CSS Language

How do we specify the styling rules of a document? The HTML elements of a page have some properties that deal with the style and the way an HTML element is drawn on the page. These properties are not the ones that we write as part of the HTML tag, but exist as accompanying properties and serve the purpose to change the way the element is displayed.

So, in simple terms, to change the value of a styling property of an element, you need to specify the name of the property and the new value that you want this property to have. Finally you need to tell which elements should use this new property with the new value.

When you want to set the style of an element you can think about being able to change things like:

- The color

- The background color
- The foreground color
- The border color
- The font
 - Family
 - Size
 - Style
 - Weight
- The size
 - width
 - height
- Text properties
 - alignment
 - letter spacing

and many-many more.

Let's see some examples:

Default style values

When you do not specify any style property value, browsers are using default values. This is called, the “user agent stylesheet”.

For example, if you write just a simple word inside the body, then:

- The font family used is Times
- Font size is 16px
- Font style is Normal
- Font weight is Normal
- Text color is Black

Set the color of the text - color

Let's start our short journey on CSS by setting the color of the text. The color of the text is specified by the property with name color.

Write the following HTML document and open it on your browser:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Color</title>
6   </head>
7
8   <body>
9     Hello World!
10  </body>
11
12 </html>
```

(the above code snippet online)

As you can see, the color of the phrase “Hello World!” is black, the default used by the browser.

We can apply a new style on the HTML tag element that contains that text, in order to change its color to blue.

Which HTML tag element contains the text “Hello World!” ? It is the `<body>`. In that case, we need to apply the new style to this particular element. Let’s do it:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Color</title>
6   </head>
7
8   <body style="color: blue;">
9     Hello World!
10  </body>
11
12 </html>
```

(the above code snippet online)

As you can see above in order to apply a new style on an element, we use the attribute with name `style`. The value of this attribute needs to be a series of style property value pairs. The style property name is separated from its value using the symbol `:`. Note that each pair ends with the symbol `;`.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Color</title>
6      </head>
7
8      <body style="color: blue;">
9          Hello World!
0      </body>
1
2  </html>

```

Style Attribute Structure

If you open the above document on a browser you will see the following:

Hello World!

CSS - Hello World

So, the `color` is used to set the text color. You can use any standard color values. You can also use numbers to specify the color. I will give you more details about that in later chapters. Until then, you can use color names like: `black`, `yellow`, `blue`, `green`, `red`, `orange` e.t.c.

Set the text size - `font-size`

Now, we will change the text size. In order to do that we use the style property `font-size`. `font-size` is calculated in pixels (which is the measure of display analysis). As we said, most of the browsers use `16px` size font as default. Let's try the double one.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Color</title>
6   </head>
7
8   <body style="color: blue; font-size: 32px;">
9     Hello World!
10    </body>
11
12 </html>
```

(the above code snippet online)

As you can see we have added the `font-size` property next to the `color` property. Let's see the visual result of this change:

Hello World!

Double Font Size

As you can see, the text size is double the size of the original version.

Set border - `border`

If we want to create a border around the content of an element we use the `border` property. This is actually a triple value property. You need to give value about the width of the border, the style of the border and the color of the border.

In the next example, we set the border of the body to be of 2px width, solid style and red color:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Color</title>
6   </head>
7
8   <body style="color: blue; font-size: 32px; border: 2px solid red;">
9     Hello World!
10    </body>
11
12 </html>
```

(the above code snippet online)

The visual result is the following:



Hello World!

Red Border on Body

Set background / fill color - background-color

Our final change on this page will be to change the background color of the body, or otherwise, the fill color. This is done by changing the value of the property `background-color`. Let's set that to the value `yellow` and see how this will affect our page content:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Color</title>
6   </head>
7
8   <body style="color: blue; font-size: 32px; border: 2px solid red; background-color: yellow;">
9     Hello World!
10    </body>
11
12 </html>
```

(the above code snippet online)

If we load this page on our browser, the result will be:

Hello World!

Body With Yellow Background Color

There are many style properties that can be changed for an HTML element and we learn more about styling in later chapters.

The `<style>` tag

The fact that I can set the style properties as values of the `style` attribute seems to be easy and handy, but it is not a good practice. Firstly, let's think about the case in which we want to change / set 30 different style properties on a particular element. That `style` attribute is going to be very long. In such cases we prefer to use the `<style>` tag and include the rules for styling inside its content.

So, the previous style rules for `body` should be written inside the `<style>` tag:

```
1 <style>
2   color: blue;
3   font-size: 32px;
4   border: 2px solid red;
5   background-color: yellow;
6 </style>
```

(the above code snippet online)

Ok. But hold on. Where is the fact that we want to apply these style rules to the `body` element? Good point! And this is the main difference here. If you put your style rules inside the `<style>` tags, you need to tell which element the rules apply to. We do that by enclosing the rules inside curly braces and specifying the element / tag name before that:

```
1 <style>
2   body {
3     color: blue;
4     font-size: 32px;
5     border: 2px solid red;
6     background-color: yellow;
7   }
8 </style>
```

(the above code snippet online)

So, if you put this inside our HTML document and remove the `style` attribute from the `body` element it will work like before. Note that we can put the `<style>` tag anywhere in the HTML document. In the `<head>`, in the `<body>` or even below closing `</body>`. But we usually prefer to have it in the `<head>` section. Also, you need to know that you can have as many `<style>` tags inside your document as you like.

Important Later, you will learn that we do not like neither having multiple `<style>` tags nor even having any such tag inside our HTML document.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Color</title>
6     <style>
7       body {
8         color: blue;
9         font-size: 32px;
10        border: 2px solid red;
11        background-color: yellow;
12      }
13    </style>
14  </head>
15  <body>
16    Hello World!
17  </body>
18
19 </html>
```

(the above code snippet online)

Load this document on your browser. The result will be the one we had before.

The reference to the CSS file

Again, we like to keep things neat, clean and separate. We do not like to mix the CSS rules, styling code, with HTML document structure code. Having the style interspersed with HTML tags and code is something that is a bad practice. It is not easily maintainable and does not allow easy reading of either the HTML or the CSS rules. There are other disadvantages too that you will learn later on as you become more experienced developer.

So, what do we do with that? We save our styling rules inside a separate text file, the CSS file. This file has filename extension .css and it is separately stored, usually inside a folder named `stylesheets`.

When we put style rules inside a CSS file, we do not enclose them inside `<style>` HTML tags because this file is not an HTML file.

Hence, create the folder `stylesheets` and save the following CSS rules inside a file named `main.css`. The folder and file should be like:

```
1 - [stylesheets]
2   | - main.css
```

(the above code snippet online)

The `main.css` file should have the content:

```
1 body {
2   color: blue;
3   font-size: 32px;
4   border: 2px solid red;
5   background-color: yellow;
6 }
```

(the above code snippet online)

The question now is how do we use this file in order to apply its rules to our HTML page. We use it by creating a special reference inside the `<head>` section of our HTML file. Like this:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
6     <title>Color</title>
7   </head>
8
9   <body>
10    Hello World!
11  </body>
12
13 </html>
```

(the above code snippet online)

Save this file with the name `index.html`. Save it on the same folder where the `stylesheets` folder is. Like this:

```
1 - [stylesheets]
2     |- main.css
3 - index.html
```

(the above code snippet online)

Then if you open the `index.html` file using your browser, you will see the same result as before.

The trick is done by the `<link>` tag that is used inside the `<head>` section.

```
1 <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
```

(the above code snippet online)

This tells the browser that has to load a referenced resource. The relationship to the current HTML document is a `stylesheet` relationship and is defined by the `rel` attribute. The actual path to the resource that needs to be loaded is defined by the attribute `href`, which has the value `stylesheets/main.css`, specifying, essentially, the path to the CSS file with the style rules that should be used.

When the browser parses the HTML document and encounters such a `<link>`, loads the corresponding rules in its memory and then proceeds with parsing the rest of the HTML document.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create a page like the following:

This is my first task on CSS

2. Pay attention to the following:
 1. The text should be included inside an `<h1>` tag.
 2. Font size that you should use is `48px`.
 3. The text color that you should use is White.
 4. The background color that you should use is Green.
 5. The border should be `5px` wide with Blue color.
3. Save your CSS file inside the `stylesheets` folder. Name the file with name `main.css`.
4. Save your main HTML file with name `index.html` on the same folder where the `stylesheets` folder is.
5. Create a zip file with the contents of this small project and send it to your mentor.

Chapter 12 - First Encounter With CSS Selectors

Summary

In this chapter we start understanding the CSS Selectors, and how we use them to specify which parts of the HTML document are styled by which CSS rules. You will be able to create a page like the following, in which different parts of the content are styled with their own specific style.

Politics

Fuming over Ryan, some conservative voices turn on the Freedom Caucus

Rep. Mark Meadows (R-N.C.) has been able to count on his Facebook page for stalwart support during his long-running battle with the House Republican leadership, including a successful effort to oust House Speaker John A. Boehner (R-Ohio).

“Keep up the great work,” read a comment posted last week. “We the people thank you for ridding us of John Boehner!”

David Weigel contributed to this report.

Web Page Styling and CSS Selectors

Learning Goals

- Learn how to select elements using the tag name.
- Learn how to select multiple element using the same tag name.
- Learn how to be more specific using the containing-contained relationship of elements.
- Learn how you can apply the same style properties using multiple selectors.

First Encounter with CSS Selectors

Selecting by the HTML tag name

In the previous chapter we used the following CSS rule to change the styling of the body element:

```

1 body {
2   color: blue;
3   font-size: 32px;
4   border: 2px solid red;
5   background-color: yellow;
6 }

```

(the above code snippet online)

The body word, before the starting { works as a CSS selector. It essentially selects, or specifies, which HTML tags will have the styling properties that are enclosed inside the following curly braces. And in this particular case, the specification, the selection rule, uses an HTML tag, the body, to specify the HTML tags that will have the styling properties that follow.

It is true that inside an HTML document there is only 1 body tag and hence, the body CSS selector actually matches only 1 tag. But this is not the general case. We might specify an HTML tag that appears many times inside the HTML document. In that case, the styling properties will be applied to all matching tags.

Let's see the following example. Create the following HTML content and save it inside the file index.html.

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
6     <title>Color</title>
7   </head>
8
9   <body>
10    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod te\mport incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ull\amco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptat\elit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat no\n proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
11    </p>
12
13    <p>
14      Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium \doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architect\o beatae vitae dicta sunt explicabo.
15      Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit,\n

```

```
27 sed quia consequuntur magni dolores
28     eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui do\
29 lorem ipsum quia dolor sit amet,
30     consectetur, adipisci velit, sed quia non numquam eius modi tempora incidun\
31 t ut labore et dolore magnam aliquam
32     quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem u\
33 llam corporis suscipit laboriosam,
34     nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehendit qui in ea voluptate velit
35     esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo\
36 voluptas nulla pariatur?
37     </p>
38     </body>
39
40
41 </html>
```

(the above code snippet online)

As you can see, this page references the `stylesheets/main.css` file. Hence, you need to create this file and make sure it has the following content:

```
1 p {
2     color: blue;
3 }
```

(the above code snippet online)

Our HTML page above has 2 `<p>` tag elements. 2 paragraphs. And the CSS rules inside the `main.css` file have a rule that says that the `p` tags need to have color blue.

If you load that page on your browser, you will see that both paragraphs are affected and they have text color blue.

*Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
 exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
 dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
 Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit
 anim id est laborum.*

*Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque
 laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi
 architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas
 sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione
 voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit
 amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut
 labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis
 nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi
 consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam
 nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla
 pariatur?*

Both Paragraphs With Text Color Blue

Specificity - Containing-Contained Relationships

Let's see a more complicated example. Here is the HTML code (write it and save inside your `index.html` file):

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
6          <title>Li Selection</title>
7      </head>
8
9      <body>
10         <ul>
11             <li>Cheese</li>
12             <li>Rice</li>
13             <li>Coffee</li>
14             <li>Milk</li>
15             <li>Wine</li>
16         </ul>
17
18         <ol>
```

```
19      <li>Blue</li>
20      <li>Green</li>
21      <li>Yellow</li>
22      <li>Black</li>
23      <li>Red</li>
24  </ol>
25  </body>
26
27 </html>
```

(the above code snippet online)

Change the content of your `stylesheets/main.css` file as follows:

```
1 li {
2   color: green;
3 }
```

(the above code snippet online)

If you load this page on your browser you should see this:

- Cheese
- Rice
- Coffee
- Milk
- Wine

1. Blue
2. Green
3. Yellow
4. Black
5. Red

All List Items Green

Easy stuff. We have specified that all list items (``) tags should have content that is colored with green color. So, the result is what we expected. But how can we say that only the unordered list, for example, needs to be colored with green color? We need to find a way to differentiate the `li` tag elements of the unordered list from the `li` tag elements of the second, ordered list.

I guess that it is easy to understand that the first `li` elements are **contained** inside `ul` and the second `li` elements are **contained** inside an `ol`. Hence, we can use the property of relationship, containing-contained relationship, to pick up in our CSS selector only the first `li` elements, those that are contained inside the `ul` element. This is how we should write our CSS rule:

```
1 ul li {  
2   color: green;  
3 }
```

(the above code snippet online)

The above rule says, “select all the `li` elements that are contained inside a `ul` element”.

Let’s try that. Save your new rule inside your `stylesheets/main.css` file (and make sure that you do not have the previous `li` rule anymore), and load your HTML page again. The result should be this:

- Cheese
- Rice
- Coffee
- Milk
- Wine

1. Blue
2. Green
3. Yellow
4. Black
5. Red

Only First List Items are Now Green

Well done. We have managed to be **more specific**. This is how CSS rules work. They can be generic, but they can be more specific. And the specific rules take precedence over the more generic ones. Let's see another example here that demonstrates this specificity:

Change your stylesheets/main.css to have the following content (and only this):

```
1 li {  
2   color: blue;  
3 }  
4 ul li {  
5   color: green;  
6 }
```

(the above code snippet online)

If you load your `index.html` page again, you will get the following result:

- Cheese
 - Rice
 - Coffee
 - Milk
 - Wine
-
1. Blue
 2. Green
 3. Yellow
 4. Black
 5. Red

CSS Rules For Nested Elements

The above rules say “set color blue to all list items, but for those that are inside a `ul` set the color to green”.

The containing-contained relationship can be specified in any depth level, although we do not like to go very deep. Let’s see another example.

Change the content of your `index.html` file to be:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
6          <title>Deep level</title>
7      </head>
8
9      <body>
10         <ul>
11             <li>Cheese</li>
12             <li>Rice</li>
13             <li>Coffee</li>
14             <li>Milk</li>
15             <li>Wine</li>
16             <li>Optional:
17                 <ol>
18                     <li>Bread</li>
19                     <li>Butter</li>
20                     <li>Tomatoes</li>
21                     <li>Chips</li>
22                 </ol>
23             </li>
24         </ul>
25
26         <ol>
27             <li>Blue</li>
28             <li>Green</li>
29             <li>Yellow</li>
30             <li>Black</li>
31             <li>Red</li>
32         </ol>
33     </body>
34
35 </html>
```

(the above code snippet online)

If you do not apply any styling on this page, it will be drawn like this: > Hint: If you want to disable, temporarily, the application of any CSS style, you only have to either remove the `<link>` line that references and loads the CSS file, or you can put this line into comments like this:

```
1  <!-- link href="stylesheets/main.css" rel="stylesheet" type="text/css" -->
```

Later, when you want that back, just remove the comment delimiters and bring back the normal opening and closing symbols, like this:

```
1 <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
```

- Cheese
- Rice
- Coffee
- Milk
- Wine
- Optional:
 1. Bread
 2. Butter
 3. Tomatoes
 4. Chips
- 1. Blue
- 2. Green
- 3. Yellow
- 4. Black
- 5. Red

Nested Ordered lists Without CSS

The question that we pose here is how can we specify that the “Optional:” ordered list has color blue, but the last ordered list has color green? This is the result that we want to achieve:

- Cheese
 - Rice
 - Coffee
 - Milk
 - Wine
 - Optional:
 1. Bread
 2. Butter
 3. Tomatoes
 4. Chips
-
1. Blue
 2. Green
 3. Yellow
 4. Black
 5. Red

Nested Ordered List Blue and Last Ordered List Green

We need to be specific enough so that the rules apply only to the elements we want.

Change your `stylesheets/main.css` file to have the following content (and only that):

```

1  ol li {
2      color: green;
3  }
4  li ol li {
5      color: blue;
6  }

```

(the above code snippet online) Reload your index.html page (make sure that the <link> references the file stylesheets/main.css) and see the result. It should be the expected one.

The first rule:

```

1  ol li {
2      color: green;
3  }

```

(the above code snippet online)

makes sure that all ordered list items have green color, but the second rule:

```

1  li ol li {
2      color: blue;
3  }

```

(the above code snippet online)

overrides the first one, for the cases in which the ordered list is part of another list item (which is the case for the “Optional:” ordered list in our page).

Now, change your stylesheets/main.css file and make sure it has the following content and only this:

```

1  ol li {
2      color: green;
3  }
4  ul ol li {
5      color: blue;
6  }

```

(the above code snippet online)

Save it and reload your index.html page. Did anything change? No, This is because the rule:

```

1  ul ol li {
2      color: blue;
3  }

```

(the above code snippet online)

matches any li that is contained inside an ol item that is contained inside a ul item. The containment relationship defined by this rule does not imply **direct** containment. The containing part can be any ancestor, in any level, of the contained part. In our case ul element is a not a direct parent of the ol element, but the rule still applies, because it doesn’t have to be.

Multiple selectors

We will finish our first encounter with CSS selectors, by using multiple selectors to apply the same style properties.

Change your `index.html` file to have the following content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
6          <title>Multiple Selectors</title>
7      </head>
8
9      <body>
10         <h1>Extract 1</h1>
11
12         <p>
13             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\por incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris \nisi ut aliquip ex ea commodo consequat.
14             Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolor\eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit \anim id est laborum.
15         </p>
16
17         <h3>Another Extract</h3>
18
19         <p>
20             But I must explain to you how all this mistaken idea of denouncing pleasure\and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it? But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant \
```

```
41 pleasure?  
42     </p>  
43  
44     </body>  
45  
46 </html>
```

(the above code snippet online)

The question here is how can we apply the color blue to the content of `<h1>` and `<h3>` tags? One can definitely do this:

```
1 h1 {  
2     color: blue;  
3 }  
4 h3 {  
5     color: blue;  
6 }
```

(the above code snippet online)

But this becomes unmaintainable, since any decision to change the color from blue to red, for example, would have to be done to multiple places in your CSS code. There is a better way:

Change the content of your `stylesheets/main.css` as follows:

```
1 h1, h3 {  
2     color: blue;  
3 }
```

(the above code snippet online)

and reload your page. You will see that you have the desired result:

Extract 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Another Extract

But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it? But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant pleasure?

Multiple Selectors with Same Style Properties

As you can see, we can separate with , one CSS selector from the other and still apply the same style properties.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create an HTML page as the one below:

Politics

Fuming over Ryan, some conservative voices turn on the Freedom Caucus

Rep. Mark Meadows (R-N.C.) has been able to count on his Facebook page for stalwart support during his long-running battle with the House Republican leadership, including a successful effort to oust House Speaker John A. Boehner (R-Ohio).

“Keep up the great work,” read a comment posted last week. “We the people thank you for ridding us of John Boehner!”

David Weigel contributed to this report.

Task: First Encounter With CSS Selectors

1. The actual content of your HTML page should be this:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
6          <title>Task for CSS Selectors</title>
7      </head>
8
9      <body>
10
11         <h6>Politics</h6>
12
13         <h1>Fuming over Ryan, some conservative voices turn on the Freedom Caucus</h1>
14     <br>
15
16         <p>
17             Rep. Mark Meadows (R-N.C.) has been able to count on his Facebook page for\
18             stalwart support during his long-running
19                 battle with the House Republican leadership, including a successful effort\
20             to oust House Speaker John A. Boehner (R-Ohio).
21         </p>
22
23         <p>
24             “Keep up the great work,” read a comment posted last week. “We the people \
25             thank you for ridding us of John Boehner!”
26         </p>
27
```

```
28      <footer>
29          <h6>David Weigel contributed to this report.</h6>
30      </footer>
31
32      </body>
33
34      </html>
```

gist

2. Send your work to your mentor for approval.

Chapter 13 - More Advanced CSS Selectors

Summary

In this chapter we improve our knowledge in CSS Selectors. With some advanced techniques we are able to style documents like this:

| Company Name | INVOICE#_____ | | |
|---|--------------------|-----------|-----------------------|
| | INVOICE DATE:_____ | | |
| BILL TO: | | | |
| Street Address | Name | | |
| City, ST ZIP CODE | Company Name | | |
| Phone [(000) 00 - 0000] | Street Address | | |
| Fax [(000) 00 - 0000] | Phone | | |
| Payment Terms: Invoice Date + 30days | | | |
| LINE | QTY ORDERED | PRICE/EXT | PRICE |
| 1 | 1,000.00 | \$1.50 | \$1,500.00 |
| 2 | 3,000.00 | \$4.00 | \$12,000.00 |
| 3 | 3,000.00 | \$2.00 | \$6,000.00 |
| 4 | 500.00 | \$1.00 | \$500.00 |
| 5 | 800.00 | \$2.00 | \$1,600.00 |
| | | | Subtotal: \$21,600.00 |
| | | | Tax Rate: 23% |
| | | | Total: \$26,568.00 |
| Make all checks payable to: <u>Company Name</u> | | | |
| Thank you for your business. | | | |

Web Page with Advanced CSS Selectors

Learning Goals

1. Learn about the `id` attribute.
2. Learn how to select elements using the `id` attribute.
3. Learn about the `class` attribute.
4. Learn how to select elements using the `class` attribute.
5. Learn how you can combine tag name and class selection in your CSS rules.
6. Learn how you can have multiple classes on the same element.

7. Learn how you can be more specific by selecting elements that have more than one class applied.

More Advanced CSS Selectors

We have learned in the previous chapters about how we can CSS select various elements of our HTML document:

- Using the HTML tag name to select one or more HTML elements with the particular tag name. For example:

```
1 p {  
2   font-size: 16px;  
3 }
```

(the above code snippet online)

which selects all paragraphs in our HTML page and applies the same font size.

- Using the HTML tag name to select one or more HTML elements with different tag names in order to apply same rules. For example:

```
1 p, h3 {  
2   font-size: 16px;  
3 }
```

(the above code snippet online)

which selects all ps and h3s HTML elements and applies the same font size.

- Using containing-contained relationship to be more specific about which elements to select. For example:

```
1 li {  
2   font-size: 16px;  
3 }  
4 ul li {  
5   color: blue;  
6 }  
7 ol li {  
8   color: green;  
9 }
```

(the above code snippet online)

which may be selecting the font size 16px for all the li elements, but make sure that lis that are contained inside ul have text color blue, whereas lis that are contained inside ol have text color green

id attribute

In HTML documents I can use a special attribute which has the name `id`. It is special because when used, it should have a unique value through out the whole HTML elements in your document. In other words, no 2 or more elements in your document should have the same `id` value.

Here is an HTML extract of a `p` element that has an `id` attribute:

```
1 <p id="main-paragraph">
2   Lorem ipsum...
3 </p>
```

(the above code snippet online)

The `id` has the value `main-paragraph`. Pretty straightforward.

Note: The values of the `id` should be following a specific style:
* Lower case letters
* If multiple words, then we separate the one word from the other using the dash - character. Occasionally, you may see the character underscore `_` to be used, but my suggestion here is that you get used to the dash - character.

Important: You should be very good with your HTML and CSS code styling. It is important to make your code easy to read and maintainable. We are suggesting that you start reading, little-by-little, the document: [Google HTML and CSS Style Guide](#).
* If you want to read about why styling matters, you can read this: [Why Coding Style Matters](#).

Here is an HTML extract that has 2 `p` elements with different ids.

```
1 <p id="introduction">
2   We are going to talk about....
3 </p>
4 <p id="main-subject-text">
5   The number of people...
6 </p>
```

(the above code snippet online)

The **very important** point that you need to understand here is that you should not reuse the same `id` inside the same HTML document. So, this is not allowed:

```
1 <p id="a-paragraph">  
2   We are going to talk about....  
3 </p>  
4 <p id="a-paragraph">  
5   The number of people...  
6 </p>
```

(the above code snippet online)

When saying *it is not allowed*, this does not mean that your browser will not display the HTML page. But a lot of tools, like CSS and JavaScript, rely on the fact that ids are unique and if they are not, you will not have the expected results. Your document might not be styled correctly. Your JavaScript code might not be functioning as expected.

CSS Selecting elements by id

Having learned about ids, we can now say that you can use them to select elements in your CSS rules.

Assuming that we have the following HTML document (create it and save it with the name `index.html`):

```
1 <!DOCTYPE html>  
2 <html>  
3   <head>  
4     <meta charset="utf-8">  
5     <title>2 paragraphs styled</title>  
6   </head>  
7   <body>  
8  
9     <p id="introduction">  
10       Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\\  
11       por incididunt ut labore et dolore  
12         magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco la\\  
13       boris nisi ut aliquip ex ea commodo  
14         consequat.  
15   </p>  
16  
17     <p id="main-text">  
18       Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium \\  
19       doloremque laudantium, totam rem aperiam,  
20         eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vita\\  
21       e dicta sunt explicabo. Nemo enim ipsam  
22         voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia conse\\  
23       uuntur magni dolores eos qui ratione voluptatem  
24         sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit \\  
25       amet, consectetur, adipisci velit, sed quia non
```

```
26      numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam,  
27      quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut alii  
28      quid ex ea commodi consequatur? Quis autem  
29      vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum  
30      fugiat quo voluptas nulla pariatur?  
31      </p>  
32      </body>  
33  </html>
```

(the above code snippet online)

we can now apply the following CSS rules in order to style the paragraphs differently:

(Save these rules inside a file with name `main.css`. And reference the file from within the `index.html`)

```
1 #introduction {  
2     font-size: 24px;  
3 }  
4  
5 #main-text {  
6     font-size: 16px;  
7 }
```

(the above code snippet online)

The `#` symbol together with the actual `id` value that we want to target, is used to CSS select HTML elements by their id.

The result of applying the above rules to our HTML document will be the following:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Selecting by id first example

The class attribute

Another tool in our tool belt when CSS selecting, is the `class` attribute. We assign this attribute to our elements and then we can select elements in our CSS rules by using the name of class preceded by the `.` symbol. A class name can be reused in many HTML elements, and, in fact, that's its real power. Because we can set a particular class to many different elements and change their style all at once by specifying CSS rules for the particular class.

Here is an example of using classes inside our HTML document:

Note: Make sure that you create the css file that is referenced inside the HTML document

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Table styled with classes</title>
6          <link rel="stylesheet" href="stylesheets/table-with-classes.css" type="text/c\
7 ss">
8      </head>
9
10     <body>
11         <table>
12             <tr>
13                 <th>Name</th>
14                 <th>Email</th>
15                 <th>Phone</th>
16                 <th>Address</th>
17             </tr>

```

```
18
19      <tr class="odd">
20          <td>Maud Adams</td>
21          <td>m.adams@mailinator.com</td>
22          <td>(541) 754-3010</td>
23          <td>795 E DRAGRAM, TUCSON AZ 85705</td>
24      </tr>
25
26      <tr class="even">
27          <td>Angelica Angurbash</td>
28          <td>a.angurbash@mailinator.com</td>
29          <td>202-555-0195</td>
30          <td>1731 Bunker Hill Road Northeast </td>
31      </tr>
32
33      <tr class="odd">
34          <td>Theoni_V._Aldredge</td>
35          <td>theoni.alredge@mailinator.com</td>
36          <td>207-555-0145</td>
37          <td>1900 Silver Cross Boulevard </td>
38      </tr>
39
40      <tr class="even">
41          <td>Kathleen_Lockhart</td>
42          <td>kathleen.lockhart@mailinator.com</td>
43          <td>207-555-0155</td>
44          <td>7105 Bailey Creek Circle Southeast </td>
45      </tr>
46
47      <tr class="odd">
48          <td>Alice_Guy-Blach</td>
49          <td>alice.guy-blach@mailinator.com</td>
50          <td>202-555-0127</td>
51          <td>3301 New Mexico Avenue Northwest #236 </td>
52      </tr>
53  </table>
54 </body>
55 </html>
```

(the above code snippet online)

The CSS file that can be used to style the document should contain the following content:

```

1  th {
2      background-color: maroon;
3      color: white;
4      font-size: 16px;
5  }
6
7  .odd {
8      background-color: lightblue;
9      color: black;
10 }

```

(the above code snippet online)

And the final result will be:

| Name | Email | Phone | Address |
|--------------------|----------------------------------|----------------|---------------------------------------|
| Maud Adams | m.adams@mailinator.com | (541) 754-3010 | 795 E DRAGRAM, TUCSON AZ 85705 |
| Angelica Angurbash | a.angurbash@mailinator.com | 202-555-0195 | 1731 Bunker Hill Road Northeast |
| Theoni_V._Aldredge | theoni.alredge@mailinator.com | 207-555-0145 | 1900 Silver Cross Boulevard |
| Kathleen_Lockhart | kathleen.lockhart@mailinator.com | 207-555-0155 | 7105 Bailey Creek Circle Southeast |
| Alice_Guy-Blach | alice.guy-blach@mailinator.com | 202-555-0127 | 3301 New Mexico Avenue Northwest #236 |

Table styling with class selector

What did you actually do?

1. We flagged the odd rows of the table with the class `odd` and the even rows of the table with the class `even`.

```

9   <body>
10  <table>
11    <tr>
12      <th>Name</th>
13      <th>Email</th>
14      <th>Phone</th>
15      <th>Address</th>
16    </tr>
17
18    <tr class="odd">
19      <td>Maud Adams</td>
20      <td>m.adams@mailinator.com</td>
21      <td>(541) 754-3010</td>
22      <td>795 E DRAGRAM, TUCSON AZ 85705</td>
23    </tr>
24
25    <tr class="even">
26      <td>Angelica Angurbash</td>
27      <td>a.angurbash@mailinator.com</td>
28      <td>202-555-0195</td>
29      <td>1731 Bunker Hill Road Northeast </td>
30    </tr>
31
32    <tr class="odd">
33      <td>Theoni_V._Aldredge</td>
34      <td>theoni.aldredge@mailinator.com</td>
35      <td>207-555-0145</td>
36      <td>1900 Silver Cross Boulevard </td>
37    </tr>
38
39    <tr class="even">
40      <td>Kathleen_Lockhart</td>
41      <td>kathleen.lockhart@mailinator.com</td>
42      <td>207-555-0155</td>
43      <td>7105 Bailey Creek Circle Southeast </td>
44    </tr>
45
46    <tr class="odd">
47      <td>Alice_Guy-Blach</td>
48      <td>alice.guy-blach@mailinator.com</td>
49      <td>202-555-0127</td>
50      <td>3301 New Mexico Avenue Northwest #236 </td>
51    </tr>
52  </table>
53 </body>
54 </html>

```

Flagged Table Rows with Odd and Even Classes

2. We then decided to style the odd rows with a particular background color and text color, with the rule:

```

1   .odd {
2     background-color: lightblue;
3     color: black;
4   }

```

(the above code snippet online)

Combining class selection with tag name selection

Sometimes there are cases that you can be more specific in your CSS selection with the help of both the class and the tag name.

Let's see an example:

Suppose that we have the following HTML document:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Class and Tag Name Selection Together</title>
6      <link rel="stylesheet" href="stylesheets/class-and-tag-name.css" type="text/c\
7 ss">
8    </head>
9
10   <body>
11
12     <h1>Announcement</h1>
13
14     <h2 class="text-danger">Must Read!</h2>
15
16     <p>
17       Today, we are expecting strong weather phenomena...
18     </p>
19
20     <p class="text-danger">
21       Inform everybody in your neighborhood.
22     </p>
23
24   </body>
25 </html>

```

(the above code snippet online)

As you can see above, we are using the class `text-danger`. We want to make sure that the elements that have this class have text color red. In order to do that, we have the following class rule inside the accompanying CSS file:

```
1 .text-danger {  
2     color: red;  
3 }
```

(the above code snippet online)

But, on top of that, we want to be more specific for the last paragraph. We want this paragraph to have italic font style. If we just do:

```
1 p {  
2     font-style: italic;  
3 }
```

(the above code snippet online)

this will affect also the first paragraph. We have to be more specific. If we just use the following rule:

```
1 .text-danger {  
2     font-style: italic;  
3 }
```

(the above code snippet online)

then again, we are going to affect the h2 header too.

In order to specify the last paragraph we can use the following rule:

```
1 p.text-danger {  
2     font-style: italic;  
3 }
```

(the above code snippet online)

As you can see, we are saying “the paragraph p that has the class .text-danger”, by concatenating the class selector next to the tag name selector.

The final page should look like this:

Announcement

Must Read!

Today, we are expecting strong weather phenomena...

Inform everybody in your neighborhood.

Tag name and CSS selector together

Multiple classes on same element

Another property of the class attribute is that can have multiple values. Hence, the HTML element can have multiple classes. The class values are separated by blank spaces. For example:

```
1 <p class="text-danger small">
2   Yesterday, we visited the Natural History Museum....
3 </p>
```

(the above code snippet online)

In the above example, the paragraph has 2 classes: `text-danger` and `small`. This means that it inherits, automatically, the CSS rules that apply to both classes.

Let's see an example of that. Suppose that we have the following HTML document:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Two Classes on Same Element</title>
6     <link rel="stylesheet" href="stylesheets/two-classes-on-same-element.css" typ\ 
7 e="text/css">
8   </head>
9
10  <body>
```

```
11
12     <h1>Announcement</h1>
13
14     <p class="text-danger">
15         We are expecting...
16     </p>
17
18     <p class="text-danger small">
19         Inform your neighbours.
20     </p>
21
22     </body>
23 </html>
```

(the above code snippet online)

As you can see, we are using 2 classes here. The `text-danger` and the `small`. Whereas the first paragraph has the class `text-danger` only, the second one has both classes applied. We want the `text-danger` class to have text color red and we want the `small` class to have font size quite small, like 9px.

Let's see how the CSS file should look like:

```
1 .text-danger {
2     color: red;
3 }
4
5 .small {
6     font-size: 9px;
7 }
```

(the above code snippet online)

Pretty simple according to what we have learned so far. The `text-danger` class elements are ruled to have the `color` property equal to red. The `small` class elements are ruled to have the `font-size` property equal to 9px. If one applies both classes to an element, then all properties apply together.

The final result of applying this to our HTML page will be:

Announcement

We are expecting...

Inform your neighbours.

Applying 2 classes on same HTML element

Finally, we will close this chapter by saying that when some elements have multiple classes applied, then we can specify CSS rules that require both classes to be present. For example, if we enhance the previous CSS rules by adding this:

```
1 .text-danger.small {  
2     font-weight: bold;  
3 }
```

(the above code snippet online)

then we are essentially say that all the HTML elements that have both `.text-danger` and `.small` classes, should be printed bold. Take the following example:

```
1 <!DOCTYPE html>  
2 <html>  
3   <head>  
4     <meta charset="utf-8">  
5     <title>Two Classes on Same Element 2</title>  
6     <link rel="stylesheet" href="stylesheets/two-classes-on-same-element.css" typ\  
7 e="text/css">  
8   </head>  
9  
10  <body>
```

```
11
12     <h1>Announcement</h1>
13
14     <p class="text-danger">
15         We are expecting...
16     </p>
17
18     <p class="text-danger small">
19         Inform your neighbours.
20     </p>
21
22     <p class="small">
23         Lore ipsum...
24     </p>
25
26     </body>
27 </html>
```

(the above code snippet online)

With the stylesheets/two-classes-on-same-element.css file holding the rules:

```
1 .text-danger {
2     color: red;
3 }
4
5 .small {
6     font-size: 9px;
7 }
8
9 .text-danger.small {
10    font-weight: bold;
11 }
```

(the above code snippet online)

the result will be this:

Announcement

We are expecting...

Inform your neighbours.

Lore ipsum...

Requiring both classes inside the CSS rule

As you can see, only the paragraph that has both `text-danger` and `small` classes is printed with bold font.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create an invoice template like the one below.

| | | | |
|--|----------------|--------------------|-------------|
| Company Name | | INVOICE#_____ | |
| | | INVOICE DATE:_____ | |
| BILL TO: | | | |
| Street Address | Name | | |
| City, ST ZIP CODE | Company Name | | |
| Phone [(000) 00 - 00000] | Street Address | | |
| Fax [(000) 00 - 00000] | Phone | | |
| Payment Terms: Invoice Date + 30days | | | |
| LINE QTY ORDERED PRICE/EXT PRICE | | | |
| 1 | 1,000.00 | \$1.50 | \$1,500.00 |
| 2 | 3,000.00 | \$4.00 | \$12,000.00 |
| 3 | 3,000.00 | \$2.00 | \$6,000.00 |
| 4 | 500.00 | \$1.00 | \$500.00 |
| 5 | 800.00 | \$2.00 | \$1,600.00 |
| Subtotal: | | | \$21,600.00 |
| Tax Rate: | | | 23% |
| Total: | | | \$26,568.00 |
| Make all checks payable to: <u>Company Name</u> | | | |
| Thank you for your business. | | | |

Invoice Template

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Invoice Template</title>
6          <link rel="stylesheet" href="stylesheets/task.css" type="text/css">
7      </head>
8
9      <body>
10         <table border="0" cellspacing="0">
11             <tr>
12                 <td rowspan="2" colspan="2" id="company-name">Company Name</td>
13                 <td colspan="2" style="text-align: center;">INVOICE#_____</td>
14             </tr>
15             <tr>
16                 <td colspan="2" style="text-align: center;">INVOICE DATE:_____</td>

```

```
17      </tr>
18
19      <tr>
20          <td colspan="3"></td>
21          <td>BILL TO:</td>
22      </tr>
23
24      <tr>
25          <td colspan="3">Street Address</td>
26          <td>Name</td>
27      </tr>
28
29      <tr>
30          <td colspan="3">City, ST ZIP CODE</td>
31          <td>Company Name</td>
32      </tr>
33
34      <tr>
35          <td colspan="3">Phone [(000) 00 - 00000]</td>
36          <td>Street Address</td>
37      </tr>
38
39      <tr>
40          <td colspan="3">Fax [(000) 00 - 00000]</td>
41          <td>Phone</td>
42      </tr>
43
44      <tr>
45          <td colspan="4">Payment Terms: Invoice Date + 30days</td>
46      </tr>
47
48      <tr>
49          <th>LINE</th>
50          <th>QTY ORDERED</th>
51          <th>PRICE/EXT</th>
52          <th>PRICE</th>
53      </tr>
54
55      <tr>
56          <td>1</td>
57          <td>1,000.00</td>
58          <td>$1.50</td>
59          <td>$1,500.00</td>
60      </tr>
61
62      <tr>
```

```
63      <td>2</td>
64      <td>3,000.00</td>
65      <td>$4.00</td>
66      <td>$12,000.00</td>
67  </tr>
68
69  <tr>
70      <td>3</td>
71      <td>3,000.00</td>
72      <td>$2.00</td>
73      <td>$6,000.00</td>
74  </tr>
75
76  <tr>
77      <td>4</td>
78      <td>500.00</td>
79      <td>$1.00</td>
80      <td>$500.00</td>
81  </tr>
82
83  <tr>
84      <td>5</td>
85      <td>800.00</td>
86      <td>$2.00</td>
87      <td>$1,600.00</td>
88  </tr>
89
90  <tr>
91      <td colspan="3">Subtotal:</td>
92      <td>$21,600.00</td>
93  </tr>
94
95  <tr>
96      <td colspan="3">Tax Rate:</td>
97      <td>23%</td>
98  </tr>
99
100 <tr>
101     <td colspan="3">Total:</td>
102     <td>$26,568.00</td>
103 </tr>
104
105 <tr>
106     <td colspan="3">Make all checks payable to:</td>
107     <td id="company-name-on-footer">Company Name</td>
108 </tr>
```

```
109
110     </table>
111
112     <p>Thank you for your business.</p>
113
114 </body>
115
116 </html>
```

(the above code snippet online)

1. You will need to pay attention to the following:
 1. Do not write CSS rules inside the HTML document
 2. Use a background color for the body
 3. Use the font family “Arial, Helvetica, sans-serif” for the body
 4. The line items of the invoice have higher height than the other rows of the invoice.
You will need to adjust the padding property for those lines.
 5. Use id styling and class styling.
 6. See that the table needs to occupy the whole width of the browser window.
 7. Here is a list of CSS properties that might be handy to solve this exercise:
 1. `background-color`, to set the background color
 2. `font-family`, to set the font set
 3. `text-align`, to align the text within a cell. Takes the values `right`, `left`, `center`, `justify`
 4. `font-size`, to set the font size. Try values like `16px`, or `12px` or similar.
 5. `width`, to specify the width of an element with regards to its parent. Try `100%` if you want to occupy the whole width
 6. `font-weight`, to set the font weight to value like `normal`, `bold`, `bolder`, `light` e.t.c.
 7. `padding-top`, to set the free space between the text/content and its border, in the top direction. Try to set that in pixels, like the `font-size`.
 8. `padding-bottom`, similar to `padding-top` but for the bottom side of the text/content.
 9. `padding-left`, similar to `padding-top` but for the left side of the text/content.
 10. `padding-right`, similar to `padding-top` but for the right side of the text/content.
 11. `border-bottom` works similar to `border` but attaches a border only at the bottom of an area.
 12. `border-top`, similar to `border` but for the top side of the area.
 13. `border-left`, similar to `border` but for the left side of the area.
 14. `border-right`, similar to `border` but for the right side of the area.
 15. For the color values, you can use values like:
 1. `lightblue`
 2. `darkblue`
 3. `white`
 4. `red`
 5. `maroon`
 6. `black`
 7. `lightgray`

Chapter 14 - Talk About Color

Summary

In this chapter we start talking about color. Color is fundamental element in order to make our HTML pages looking nice. We will be able to define color in various formats and build pages like this:

| Color Name | Color Hex Value | Color RGB Value | Color Name |
|---------------|-----------------|------------------|------------|
| Medium Purple | #9370DB | rgb(147,112,219) | color |
| Rosy Brown | #BC8F8F | rgb(188,143,143) | color |
| Crimson | #DC143C | rgb(220,20,60) | color |
| Moccasin | #FFE4B5 | rgb(255,228,181) | color |
| Beige | #F5F5DC | rgb(245,245,220) | color |

Web Page With Colors

Learning Goals

1. Learn to specify colors by name.
2. Learn to specify colors by hex values of red, green and blue.
3. Learn to specify colors by dec values of red, green and blue.
4. Learn about transparency and opacity.

Let's Talk about Color

Using colors is a must when you design a Web page. If you do not use colors, the browser uses black color for text and white for background. Very boring page colors of course and if we want to have live, nice-looking cool pages, we need to start using colors.

There are many CSS style properties that use colors and you can definitely set the color for anything that can be visually displayed as part of the content of your Web page.

Text Color and Background Color

The two most important color properties are:

1. color which specifies the color of text.
2. background-color which specifies the color of the background.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Invoice Template</title>
6          <link rel="stylesheet" href="stylesheets/basic-color-attributes.css" type="text/css">
7      </head>
8
9
10     <body>
11         <p>
12             Introduction to Computer Programming is a series of video lectures.....
13         </p>
14     </body>
15
16 </html>
```

(the above code snippet online)

If you see the above example, the css of which is:

```

1  p {
2      padding-top: 10px;
3      padding-bottom: 10px;
4      padding-left: 10px;
5      padding-right: 10px;
6
7      color: white;
8      background-color: darkblue;
9
10     font-size: 20px;
11 }
```

(the above code snippet online)

you will see that you have chosen as background color darkblue and foreground color white.

The visual result should be something like this:

Important Always try the code that the chapter is using in order to explain to you the concepts. This is the only way you will learn programming. Just reading is not enough. Also, you need to stop copying/pasting the code to your editor. You HAVE to type everything from the beginning. Copying/pasting will not differ from just reading the code. You need to type the code text yourself. Do

mistakes and correct them. Otherwise, you will never become a good developer.

Introduction to Computer Programming is a series of video lectures....

Specifying the color value

We have already seen example on how we can specify the color value. We have used some standard names like red, blue and white.

There are various ways one can specify the color value.

By name

You can use names like the following:

- blue
- blueviolet
- cadetblue
- darkred
- red
- yellow e.t.c.

There are 140 color names supported by all browsers. The following Web Page lists all of them:

[List of Colors Supported by All Browsers](#)

By Color Value

Colors can be set using the combination of RED, GREEN and BLUE colors.

One can use the hexadecimal value of a color. The hexadecimal has 3 parts. The first part refers to red value. The second part refers to green value and the third part refers to blue value. We also prefix the value with the # symbol.

All 3 parts should have a value in the range 00..#FF.

Take for example the value:

#000000

This has 00 for red, 00 for green and 00 for blue. In other words, we have absolute absence of all three color components. Which color is that?

It is the black color!

#000000

Let's try this number here:

```
#FF0000
```

This has the value FF for red, 00 for green and 00 for blue. This means that we have the maximum value for red, but the absence of the other 2 color components. What color will that be?

It will be red:

```
#FF0000
```

With this technique we have 16M different colors. Huge number of colors that we can create and use in our web pages. Let's try that in a real web page. Here is the code of our HTML page again:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Invoice Template</title>
6          <link rel="stylesheet" href="stylesheets/basic-color-attributes.css" type="text/css">
7      </head>
8
9
10     <body>
11         <p>
12             Introduction to Computer Programming is a series of video lectures.....
13         </p>
14     </body>
15
16 </html>
```

(the above code snippet online)

We change the content of the `stylesheets/basic-color-attributes.css` as follows:

```

1  p {
2      padding-top: 10px;
3      padding-bottom: 10px;
4      padding-left: 10px;
5      padding-right: 10px;
6
7      color: #7CFC00;
8      background-color: #4682B4;
9
10     font-size: 20px;
11 }
```

(the above code snippet online)

The visual result will be this:

Introduction to Computer Programming is a series of video lectures....

Colors by hex values

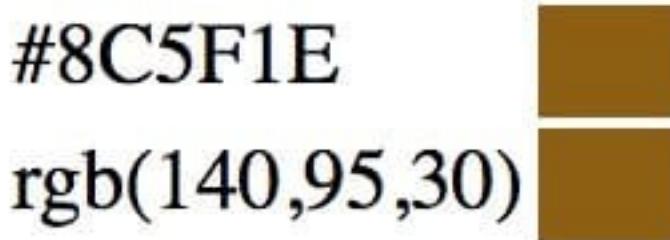
Please, note that you can use the notation `rgb(<dec red>, <dec green>, <dec blue>)` instead of using the `#<hex red><hex green><hex blue>`. The `rgb` notation uses the integer numbers 0 up to 255 for the components. With 0 meaning absence and 255 full presence.

Here is an HTML fragment that sets the background of two cells to the same color, but using different technique:

```
1 <table>
2   <tr><td>#8C5F1E</td><td style="background-color: #8C5F1E;"></td></tr>
3   <tr><td>rgb(140,95,30)</td><td style="background-color: rgb(140,95,30);"></td><\n4 /tr>
5 </table>
```

(the above code snippet online)

with the final visual result to be:



Same Color Defined With Two Different Techniques

Note: There are a lot of resources on the Internet that can give you HTML color names and their mapping to HEX values. * [Convert Names to HEX](#)

Also you can find HEX to decimal converters [like this](#).

Did you know that you can create a different level of gray shade by giving the same value to each one of red, green and blue components? See the following HTML fragment which shades 5 table cells with different shades of gray.

```

1  <table>
2    <tr><td>rgb( 5,      5,      5)</td><td style="background-color: rgb( 5, 5, 5);"> \
3      &nbsp;</td></tr>
4    <tr><td>rgb( 15,    15,    15)</td><td style="background-color: rgb( 15, 15, 15);"> \
5      &nbsp;</td></tr>
6    <tr><td>rgb( 55,    55,    55)</td><td style="background-color: rgb( 55, 55, 55);"> \
7      &nbsp;</td></tr>
8    <tr><td>rgb(105, 105, 105)</td><td style="background-color: rgb(105,105,105);"> \
9      &nbsp;</td></tr>
10   <tr><td>rgb(205, 205, 205)</td><td style="background-color: rgb(205,205,205);"> \
11     &nbsp;</td></tr>
12 </table>
```

(the above code snippet online)

RGB Color With Transparency

There are times that you want to specify a color alongside a transparency. This means that the color applied allows part of the background color to go through it.

For this purpose we are using the function `rgba(<dec red part>, <dec green part>, <dec blue part>, <opacity>)`.

For example, the color `rgba(0, 0, 0, 0.5)` is a black color that is 50% transparent (or 50% opaque). Or, the value `rgba(255, 255, 255, 0.1)` is a white color that is 90% transparent (or 10% opaque).

Try the following HTML page:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Background Color with Transparency</title>
6      <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7    </head>
8
9    <body>
10      <h1>Background Color with Transparency</h1>
11    </body>
12 </html>
```

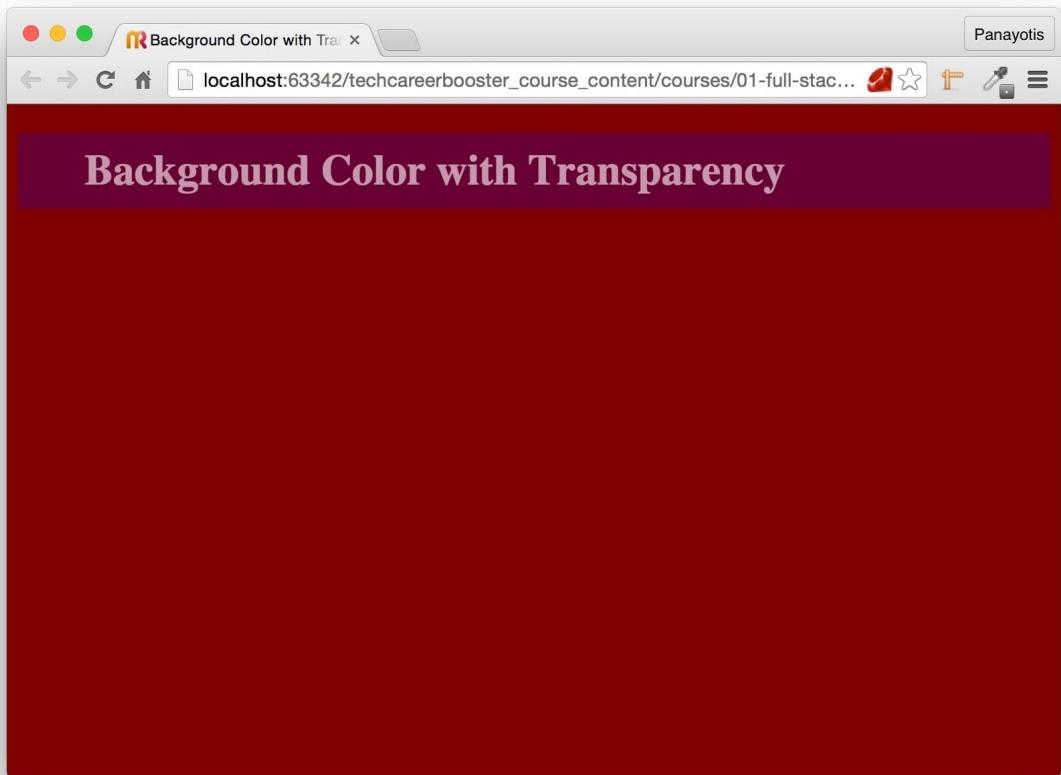
(the above code snippet online)

with the following CSS file (in `stylesheets/main.css`):

```
1 body {  
2     background-color: maroon;  
3 }  
4  
5 h1 {  
6     background-color: rgba(0, 0, 255, 0.2);  
7     padding: 10px 50px;  
8     color: rgba(255, 255, 255, 0.6);  
9 }
```

(the above code snippet online)

Save those files and load the page in your browser. You will see the following page:



Page with Background Colors Using Transparency

As you can see here, both the background color and the color of the `h1` element are transparent, letting some of the background color of the body being visible to the user eye.

Play with the opaque value. Set it to 1.0. Set it to 0.1. Set it to 0.0. Inspect how the various values affect the final result.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create the following page using a table and cells.

| Color Name | Color Hex Value | Color RGB Value | Color Name |
|---------------|-----------------|------------------|------------|
| Medium Purple | #9370DB | rgb(147,112,219) | color |
| Rosy Brown | #BC8F8F | rgb(188,143,143) | color |
| Crimson | #DC143C | rgb(220,20,60) | color |
| Moccasin | #FFE4B5 | rgb(255,228,181) | color |
| Beige | #F5F5DC | rgb(245,245,220) | color |

Table with color cells

2. In order to color the text of the first column you should use the name value for the color.
3. In order to color the text of the second column you should use the hex value of the color.
4. In order to color the text of the third column you should use the rgb value of the color.
5. Use any method you want to correctly color the background of the fourth column.
6. The last row, first three columns, have a background color that make the Beige color appear more clearly. Pick up any background color that you want for these three columns. We are using darkblue here.
7. For this exercise only, you should not use a separate CSS file. You have to use the style attribute.

Chapter 15 - Formatting Parts Of Text

Summary

In this chapter we start learning how we can style specific parts of our text. You will build a page like this:

Making Way €14.99

Theo Dorgan

Early as Clare had woken, Tom had woken earlier, resolved. He'd filled up his water tanks, been to the bustling market for fruit and vegetables, the supermarket for other provisions, the harbour office to pay his dues. Everything stowed away, he was thinking about Clare as he tidied his lines, unhooked and stowed the sail cover, ran some basic engine checks, hosed down the deck and cockpit.

Parting, the night before, he'd been on the point of asking: will we see each other tomorrow? When he'd realised, simply, there was no need to ask. Now, shaving carefully, he wondered about this. Women liked Tom, he knew this, and he liked women. He often felt, with men, that he was acting some tedious part, some cartoon version of being a man. Not with women. Some men, he amended the thought, some men. It was true, he had some good male friends, and he asked himself, as the thought jinked away from him, what those men had in common. They didn't bluster, he decided, they were brave, mostly; brave in their feelings, he amended this, forceful and tactful. And they liked women, and women liked them. Women, a sardonic young nephew had informed him recently, are the beta version. He'd been ready to explain this, in the lordly way of young techies, been disconcerted when Tom had laughed and punched him on the shoulder, told him there was hope for him yet. Remembering this now, Tom laughed again, thinking of Clare with true appreciation. Not a word wasted, the phrase came unbidden, not a word wasted. Then, finally, clarity at last as he held his head under the cold tap:

What is this about? What next?

He had a sudden, searing urge to see her. Now. "Come on", he muttered, shoving his feet into deck shoes, tugging a T-shirt over his head, "come on." Then, "steady now lad, steady, steady." He felt something that was not quite an erotic electricity course through him, a charge... - See more at: <http://www.drb.ie/new-books/making-way#sthash.eIvAFgBW.dpuf>

Styling Parts of Text

Learning Goals

1. Learn how you can use the `` tag in order to style parts of your text.
2. Learn how you can use the following HTML formatting tags:
 1. `<i>` and ``
 2. `` and ``
 3. `<small>`
 4. `<mark>`
 5. ``

6. <sub>
7. <sup>

Formatting parts of text

We have learned how we can apply basic style to our HTML elements. But how do we style parts of text that is not contained in any particular HTML element?

For example, how can we create the following effect?

The 17-year story behind Rubio's knockdown of Bush

The senator's confrontation with *Jeb Bush* in Wednesday's debate had long been brewing. Their presidential campaigns put the old Florida allies on a path to collide.

Making Part of Text Italic

As you can see in this image, part of the text, the name Jeb Bush, is styled with italic font. We haven't learned so far, any way we can do that.

But of course, there is one, and not only one, as always.

span tag

One way we can do that is by including the part of the text that we want to style inside a tag. Then, we can style the element in any way we like, using standard CSS rules.

Hence, the above result can be achieved with the following combination of HTML and CSS code:

The HTML first:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Span Element</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
```

```

10   <h1>The 17-year story behind Rubio's knockdown of Bush</h1>
11
12   <p>
13     The senator's confrontation with <span class="italic">Jeb Bush</span> in We\
14 dnesday's debate had long been brewing. Their presidential campaigns put the old \
15 Florida allies on a path to collide.
16   </p>
17
18 </body>
19 </html>
```

(the above code snippet online)

And the CSS next:

```

1 .italic {
2   font-style: italic;
3 }
```

(the above code snippet online)

As you can see from the HTML code, we have included the name Jeb Bush inside a `span` tag. And then we styled the `span` element using the class `italic`.

So, `` elements are very useful, because they do not change the structure of the document, but they can certainly be used to style parts of the document that cannot be, otherwise, selected.

HTML formatting tags

Another way you can accomplish the same effect, is to use the HTML tag `<i>` (from `italic`) or the HTML tag `` (from `emphasize`).

**<i> and **

They turn their content to `italic` style. However, the ``, gives also a semantic emphasized importance to the content that it encloses. This is usually taken into consideration by screen readers that read aloud the content of the page to computer users that have disabilities related to their vision.

See the following HTML extract:

```

1 <p>
2   The next word is <i>italic</i>. And the next word is <em>emphasized</em>.
3 </p>
```

(the above code snippet online)

which should be displayed as:

The next word is *italic*. And the next word is *emphasized*.

So, this is the equivalent effect as if you had `style="font-style: italic;"`.

Let's see some other HTML formatting tags:

** and **

They turn their content to bold. However, the `` tag gives also a semantic property of "strong" to its content and might be pronounced differently by screen readers. See the following HTML extract:

```
1 <p>
2   The next word is <b>bold</b>. And the next word is <strong>strong</strong>.
3 </p>
```

(the above code snippet online)

which should be displayed as:

`<p> The next word is bold. And the next word is strong. </p>`

The next word is **bold**. And the next word is **strong**.

So, this is the equivalent effect as if you had `style="font-weight: bold;"`.

<small>

It turns its content to small font size.

See the following HTML extract:

```
1 <p>
2   The next phrase is small: <small>This is a small phrase.</small>
3 </p>
```

(the above code snippet online)

which should be displayed as:

The next phrase is small: This is a small phrase.

Small Text Thanks To `<small>` tag

This is equivalent to setting a smaller size value to the `font-size` property.

<mark>

It highlights its content with yellow background color. As if the reader has used a highlighter to mark part of the text.

See the following HTML extract:

```
1 <p>
2   The next phrase is marked/highlighted: <mark>This is a marked phrase.</mark>
3 </p>
```

(the above code snippet online)

which should be displayed as:

The next phrase is marked/highlighted: **This is a marked phrase.**

Marked Text Thanks to <mark> tag

This is equivalent to setting the background-color property.

It makes the text look as if somebody has deleted it.

See the following HTML extract:

```
1 <p>
2   The next phrase is deleted: <del>This is a deleted phrase.</del>
3 </p>
```

(the above code snippet online)

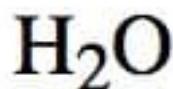
which should be displayed as:

The next phrase is deleted: ~~This is a deleted phrase.~~

This is equivalent to setting the text-decoration property with value line-through.

<sub>

It makes the text printed below the base line of the main text. See for example this:



Subscript Example

Do you see how the number 2 has been printed? In order to produce the above result, you have to enclose the number 2 into a `<sub>` element (from subscript). Here is the HTML extract that does this:

```
1 H<sub>2</sub>0
```

(the above code snippet online)

Important: This is a tag which does not have an equivalent approach in CSS. And this is correct, because this tag has nothing to do with the styling of your text/document. So, don't look for CSS equivalents.

`<sup>`

Similar to `<sub>`, you can print part of your text elevated above the baseline. This is a superscript and can be achieved with the `<sup>` tag. For example, see the following:

January 1st

Superscript Example

In order to do that you need to write the following HTML code:

```
1 January 1<sup>st</sup>
```

(the above code snippet online)

Again, do not look for CSS alternatives. `<sup>` has nothing to do with styling.

Applying multiple formatting

There are times that we want to apply more than one HTML tag formatting. Can we do that?

For example: Can I have part of the text being marked and small at the same time? Like this?

`<p> Next phrase is both marked and small: <mark><small>Phrase marked and small</small></mark></p>`

Next phrase is both marked and small: Phrase marked and small

Marked And Small

Easy! You only have to include it in both `<mark>` and `<small>` tags. Nesting one tag inside the other. Like a Russian doll. Make sure that tags close in the reverse order they have been opened.

```

1  <p>
2      Next phrase is both marked and small: <mark><small>Phrase marked and small</small>
3  </mark>
4  </p>
```

(the above code snippet online)

Closing note

If you can do things with CSS styling and outside the HTML document, then prefer this method. On the other hand, it will not harm anybody to use, for example, semantic HTML tags like `` in order to emphasize the content of your document. Nevertheless, you can always apply rules to change the style of the formatting tags themselves. See for example the following page:

The 17-year story behind Rubio's knockdown of Bush

The senator's confrontation with ***Jeb Bush*** in Wednesday's debate had long been brewing. Their presidential campaigns put the old Florida allies on a path to collide.

Change the default style of em tag

This has been created with the following HTML code:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Change the em tag style</title>
6          <link rel="stylesheet" href="stylesheets/change_em_style.css" type="text/css">
7      </head>
8
9      <body>
10         <h1>The 17-year story behind Rubio's knockdown of Bush</h1>
11
12         <p>
13             The senator's confrontation with <em>Jeb Bush</em> in Wednesday's debate ha\
14             d long been brewing.
15             Their presidential campaigns put the old Florida allies on a path to collid\
16             e.
```

```
17      </p>
18  </body>
19
20 </html>
```

(the above code snippet online)

and the following CSS code:

```
1  em {
2      font-weight: bold;
3      font-style: italic;
4 }
```

(the above code snippet online)

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

- (1) Create the following HTML page (for your convenience, parts of the text are given below, just copy / paste them in your editor):

Making Way €14.99

Theo Dorgan

Early as Clare had woken, Tom had woken earlier, resolved. He'd filled up his water tanks, been to the bustling market for fruit and vegetables, the supermarket for other provisions, the harbour office to pay his dues. Everything stowed away, he was thinking about Clare as he tidied his lines, unhooked and stowed the sail cover, ran some basic engine checks, hosed down the deck and cockpit.

Parting, the night before, he'd been on the point of asking: will we see each other tomorrow? When he'd realised, simply, there was no need to ask. Now, shaving carefully, he wondered about this. Women liked Tom, he knew this, and he liked women. He often felt, with men, that he was acting some tedious part, some cartoon version of being a man. Not with women. Some men, he amended the thought, some men. It was true, he had some good male friends, and he asked himself, as the thought jinked away from him, what those men had in common. They didn't bluster, he decided, they were brave, mostly; brave in their feelings, he amended this, forceful and tactful. And they liked women, and women liked them. Women, a sardonic young nephew had informed him recently, are the beta version. He'd been ready to explain this, in the lordly way of young techies, been disconcerted when Tom had laughed and punched him on the shoulder, told him there was hope for him yet. Remembering this now, Tom laughed again, thinking of Clare with true appreciation. Not a word wasted, the phrase came unbidden, not a word wasted. Then, finally, clarity at last as he held his head under the cold tap:

What is this about? What next?

He had a sudden, searing urge to see her. Now. "Come on", he muttered, shoving his feet into deck shoes, tugging a T-shirt over his head, "come on." Then, "steady now lad, steady, steady." He felt something that was not quite an erotic electricity course through him, a charge... - See more at: <http://www.drb.ie/new-books/making-way/#sthash.eIiFAFgBW.dpuf>

Formatting Parts of Text - Task

```
1 Making Way 14.99
2
3 Theo Dorgan
4
5 Early as Clare had woken, Tom had woken earlier, resolved. He'd filled up his wat\
6 er tanks, been to the bustling market for fruit and vegetables,
7 the supermarket for other provisions, the harbour office to pay his dues. Everyth\
8 ing stowed away, he was thinking about Clare as he tidied his lines,
9 unhooked and stowed the sail cover, ran some basic engine checks, hosed down the \
10 deck and cockpit.
11
12 Parting, the night before, he'd been on the point of asking: will we see each oth\
13 er tomorrow? When he'd realised, simply, there was no need to ask.
14 Now, shaving carefully, he wondered about this. Women liked Tom, he knew this, an\
15 d he liked women. He often felt, with men, that he was acting some
16 tedious part, some cartoon version of being a man. Not with women. Some men, he a\
17 mended the thought, some men. It was true, he had some good male friends,
```

18 and he asked himself, as the thought jinked away from him, what those men had in \
19 common. They didn't bluster, he decided, they were brave, mostly;
20 brave in their feelings, he amended this, forceful and tactful. And they liked w\|
21 omen, and women liked them. Women, a sardonic young nephew had
22 informed him recently, are the beta version. He'd been ready to explain this, in \
23 the lordly way of young techies, been disconcerted when Tom had
24 laughed and punched him on the shoulder, told him there was hope for him yet. Rem\|
25 embering this now, Tom laughed again, thinking of Clare with true
26 appreciation. Not a word wasted, the phrase came unbidden, not a word wasted. The\|
27 n, finally, clarity at last as he held his head under the cold tap:
28
29 What is this about? What next?
30
31 He had a sudden, searing urge to see her. Now. "Come on", he muttered, shoving h\|
32 is feet into deck shoes, tugging a T-shirt over his head,
33 "come on." Then, "steady now lad, steady, steady." He felt something that was not\|
34 quite an erotic electricity course through him, a charge...
35 - See more at: <http://www.drb.ie/new-books/making-way#sthash.e1FAFgBW.dpuf>

(the above code snippet online)

(2) Watch out the following points:

Making Way €14.99

Theo Dorgan

Early as Clare had woken, Tom had woken earlier, resolved. He'd filled up his water tanks, been to the bustling market for fruit and vegetables, the supermarket for other provisions, the harbour office to pay his dues. Everything stowed away, he was thinking about Clare as he tidied his lines, unhooked and stowed the sail cover, ran some basic engine checks, hosed down the deck and cockpit.

Parting, the night before, he'd been on the point of asking: will we see each other tomorrow? When he'd realised, simply, there was no need to ask. Now, shaving carefully, he wondered about this. Women liked Tom, he knew this, and he liked women. He often felt, with men, that he was acting some tedious part, some cartoon version of being a man. Not with women. Some men, he amended the thought, some men. It was true, he had some good male friends, and he asked himself, as the thought jinked away from him, what those men had in common. They didn't bluster, he decided, they were brave, mostly; brave in their feelings, he amended this, forceful and tactful. And they liked women, and women liked them. Women, a sardonic young nephew had informed him recently, are the beta version. He'd been ready to explain this, in the lordly way of young techies, been disconcerted when Tom had laughed and punched him on the shoulder, told him there was hope for him yet. Remembering this now, Tom laughed again, thinking of Clare with true appreciation. Not a word wasted, the phrase came unbidden, not a word wasted. Then, finally, clarity at last as he held his head under the cold tap:

What is this about? What next?

He had a sudden, searing urge to see her. Now. "Come on", he muttered, shoving his feet into deck shoes, tugging a T-shirt over his head, "come on." Then, "steady now lad, steady, steady." He felt something that was not quite an erotic electricity course through him, a charge... - See more at: <http://www.drb.ie/new-books/making-way/#sthash.eIxFAGBW.dpuf>

Formatting Parts of Text - Task - Points to be Careful With

- (3) Use **only** HTML formatting tags. Do not use CSS styling.

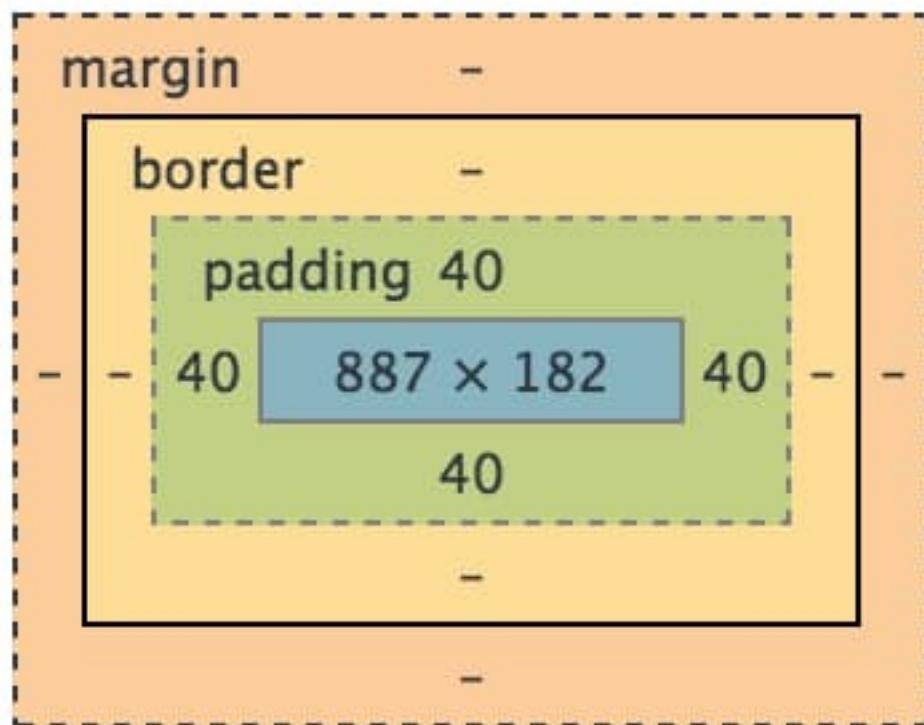
Chapter 16 - CSS Box Model

Summary

In this chapter you will learn about the CSS box model, which is very important in order for you to understand how elements are put and sized on the page.

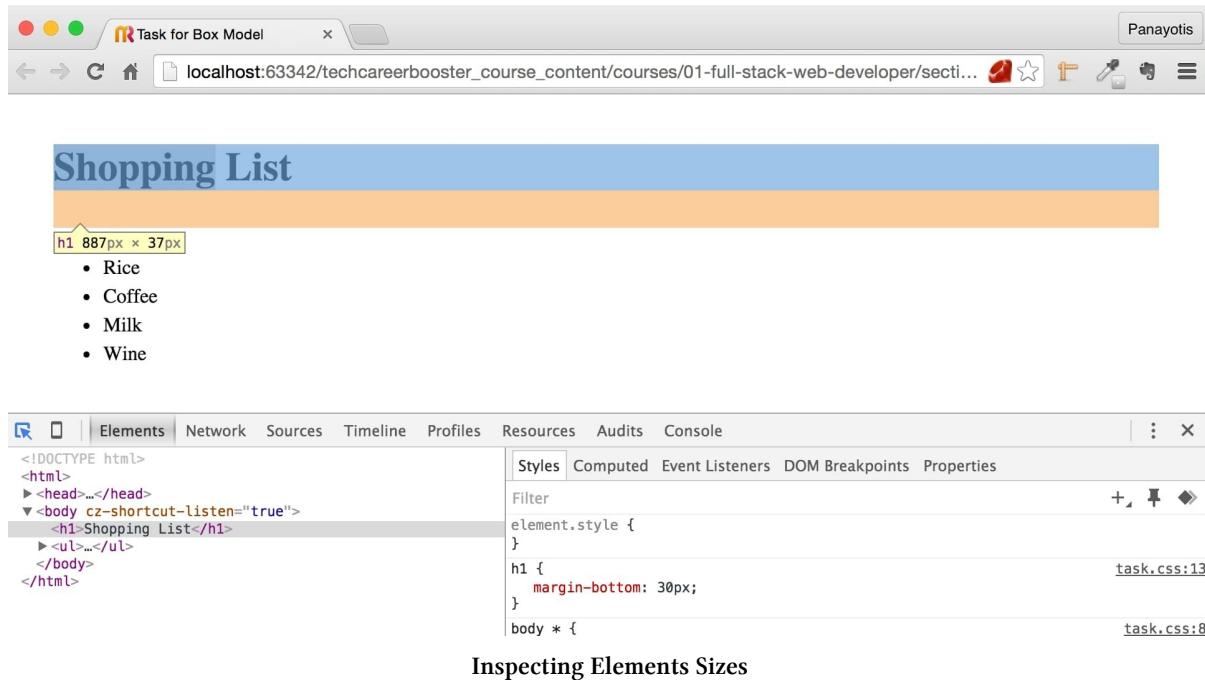
You will also learn how to use Developer Tools, a necessary tool to every Web developer. This will allow you to inspect metrics and properties of various elements on your HTML page.

So, you will be able to see the CSS box model of each element like this:



[Visualize CSS Box Model](#)

and inspect properties and their sizes like this:



Learning Goals

1. Learn about the content area.
2. Learn about the padding.
3. Learn about the border.
4. Learn about the margin.
5. Learn how to set the size of padding, margin, border, using padding, margin, border.
6. Learn how to set the size of padding and margin, using side specific CSS properties, like padding-top.
7. Learn to use Chrome (or other browser) Developer Tools to identify the box model of the selected element.
8. Learn to use Chrome (or other browser) Developer Tools to inspect an element and its style properties.
9. Learn about the box-sizing property and the two basic values content-box and border-box.
10. Learn how interpretation of width is affected by the value of the box-sizing property.
11. Learn about line height and how browser reserves space to accommodate ascenders and descenders characters.
12. Learn about the inheritance (or absence of it) for style properties.
13. Learn how you can specify CSS rules that match all the elements in a document, using the * selector.

CSS Box Model

Before we proceed in more advanced techniques in authoring our HTML page, it is very important to understand the CSS Box Model. This is how browser models how an area in your HTML page is being drawn.

Let's start with an empty page. We will build on that while we will be learning various concepts in the CSS box model area.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>CSS Box Model</title>
6     <link rel="stylesheet" href="stylesheets/index.css">
7   </head>
8
9   <body>
10
11
12 </body>
13
14 </html>
```

(the above code snippet online)

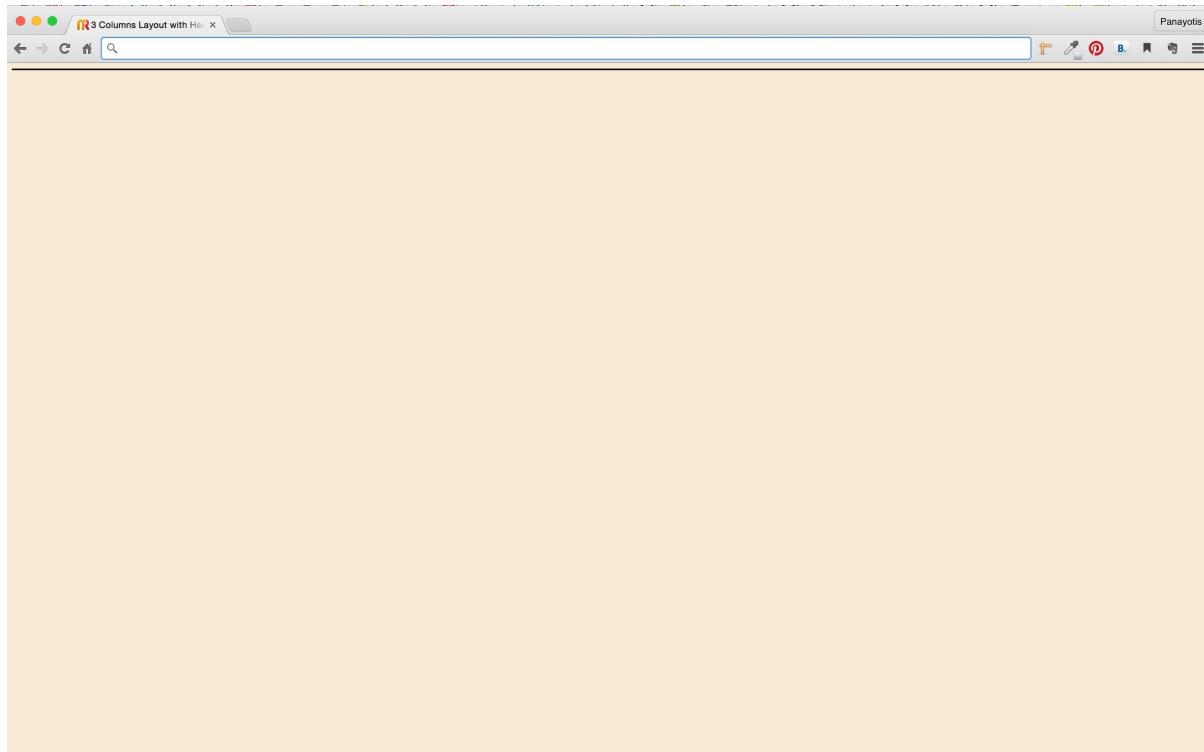
Boundaries of Body

Our HTML page has a <body>. Let's see what are its boundaries. Add the following to your CSS file:

```
1 body {
2   background-color: antiquewhite;
3   border: 1px solid black;
4 }
```

(the above code snippet online)

Save all files and reload your page. You should see this one below:



Body with Border and Background

As you can see above, there is a problem with the border. One might expect a border around the edges of the browser window. But this is not the case. The border surrounds the body according to its height. And the height of the body is 0, because it does not have any content. Since we don't have any content inside our body element, the box that is supposed to be around it, collapses into a single line.

Note: Since the height of the body is 0, why does the background color of the body expand to the whole viewport? See at the end of the chapter about this.

Let's prove that if we add some content to the body, then border will not collapse to one line. We will add some content inside the body and we will refresh our page.

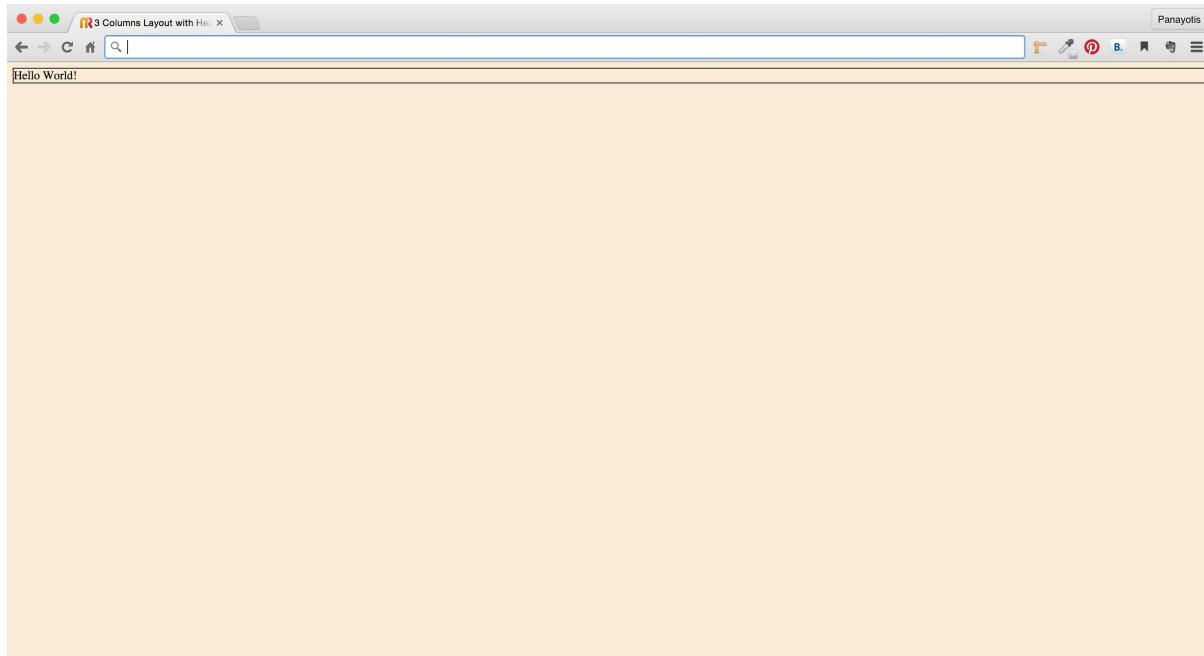
Visible Border Box

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>CSS Box Model</title>
6          <link rel="stylesheet" href="stylesheets/index.css">
7      </head>
8
9      <body>
10
```

```
11     Hello World!
12
13     </body>
14
15 </html>
```

(the above code snippet online)

What we see is the following:



Body with some content

Now things are getting better. We can clearly see the border around the content of the `<body>` element.

Padding Area

Padding is the blank area that is between the content and the border of an element. So, in our particular example is the blank area around the “Hello World!” and the borders. One way we can understand where the padding is, is that we can change its size and refresh our page.

Let's do that. In our CSS file make sure the body selector has the following rules:

```
1 body {
2     background-color: antiquewhite;
3     border: 1px solid black;
4
5     padding-top: 30px;
6 }
```

(the above code snippet online)

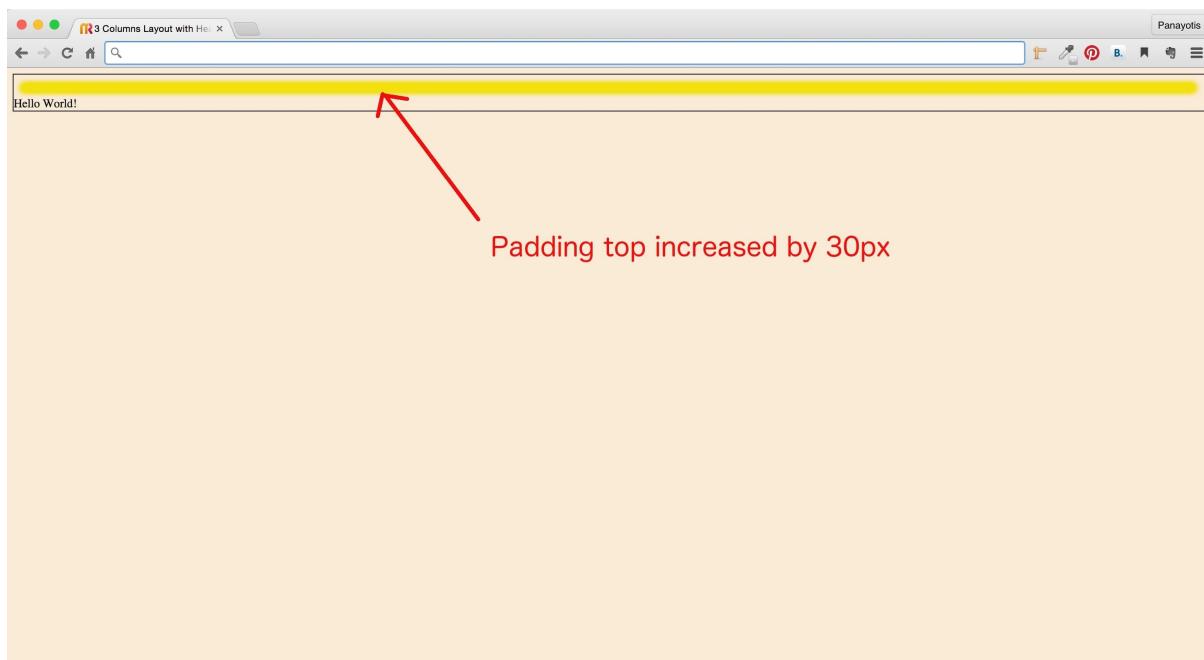
The new style property here is the padding-top. As its name implies, it controls the size of the top side of the padding. Whatever it is above the content of the element up and until the border starts.

If you refresh your page you will see the following:



Body with Big Padding Top

As you can see the top padding has been increased by 30px.



Padding Top Marked

Note that you can separately control the padding on other sides: `padding-left`, `padding-bottom`, `padding-right`. Also, you can always use the `padding` property which can control all the sides:

1. `padding: 10px;`, when we specify 1 value, then all sides are going to have the same padding.
2. `padding: 10px 20px;`, when we specify 2 values, then first size refers to top and bottom margins and second size refers to right and left margins.
3. `padding: 10px 20px 30px;`, when we specify 3 values, then first goes to top, second to right and left and third size goes to bottom.
4. `padding: 10px 20px 30px 40px;`, when we specify 4 values, then first goes to top, second goes to right, third goes to bottom and last goes to left (like going around in a clockwise direction)

Margin Area

Besides the content, padding and border you might have noticed that there is a blank area outside the border, around the box, reaching the edges of the window. This is the margin. In fact, the margin, by default has the same size on all sides. So, even if you see a lot of blank space below the bottom border, it is only a small number of pixels that are part of the bottom margin.

In fact, by default, the `body` element has 8pixels size on all sides of the element.



Body Margin

In order for you to visualize the margin, I have highlighted that in the above picture. Is that clear?

Again, margin is the area that surrounds the border of the element.

CSS Box Model again

As far as the area that an element occupies, we have learned about (inside out):

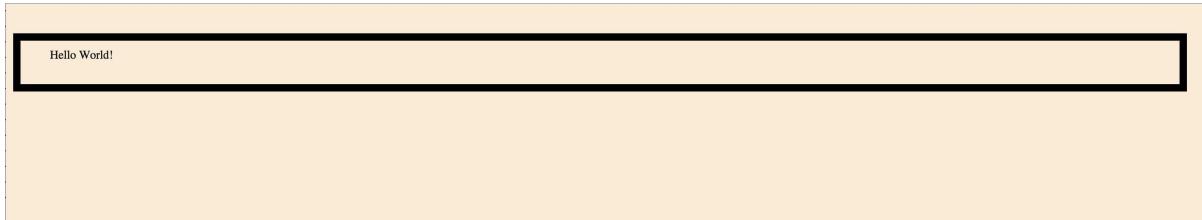
- content
- padding, which surrounds content
- border, which surrounds padding
- margin, which surrounds border

Let's change our CSS to be as follows:

```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 40px 30px 20px 10px;  
9 }
```

(the above code snippet online)

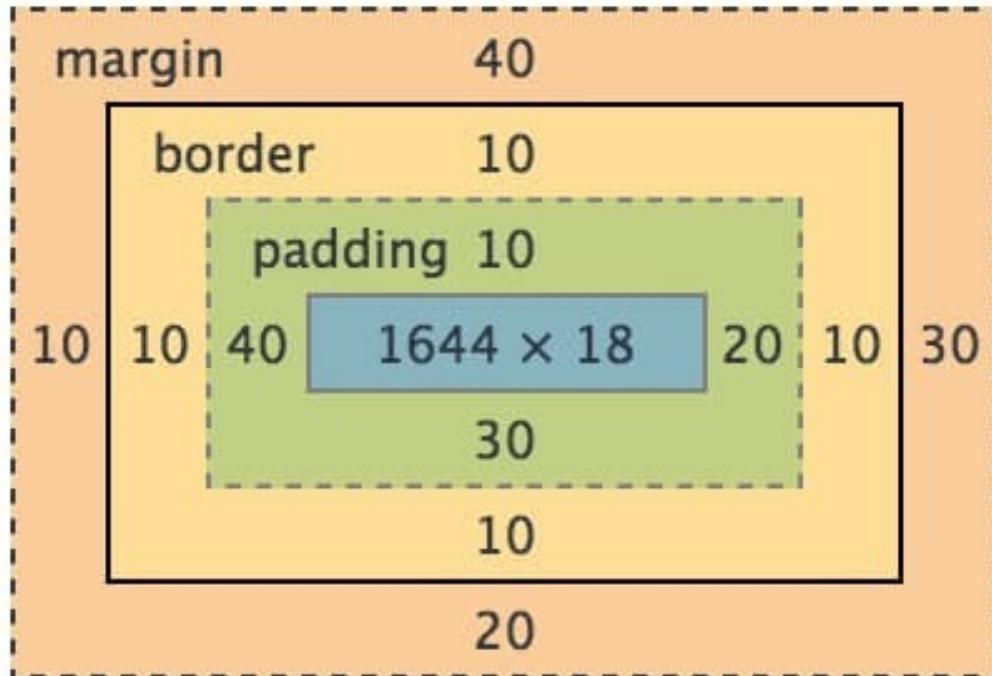
and reload our page. It will look like this:



CSS Box Model with big border

Can you identify the various areas?

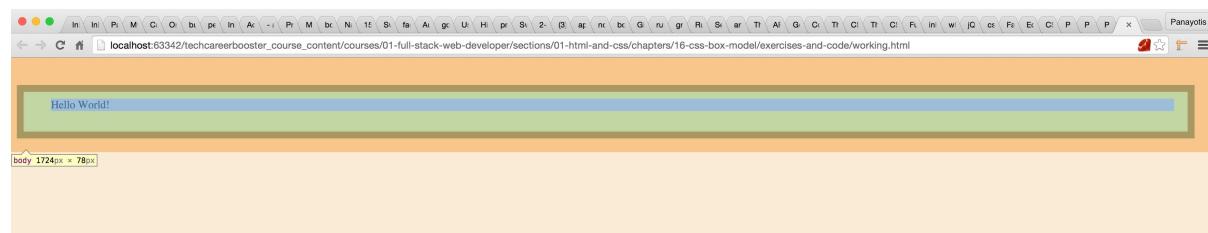
This is what you should be having in your mind for this particular HTML page:



Box Model Example

What's in the middle is the width and height, which is dynamically calculated and would probably be different on your end. It depends on the size of your browser window. But the rest of the metrics should be the same.

And this is another view:



Another view of the box model

From inside out

- the light blue area is the content area,
- the light green area is the padding area (see how the different padding sides have their own size according to our CSS rule)
- the dark brown area is the border, which has equal size on all sides

- the orange external area is the margin (see how the side sizes differ according to our CSS rule)

Use Chrome Developer Tools

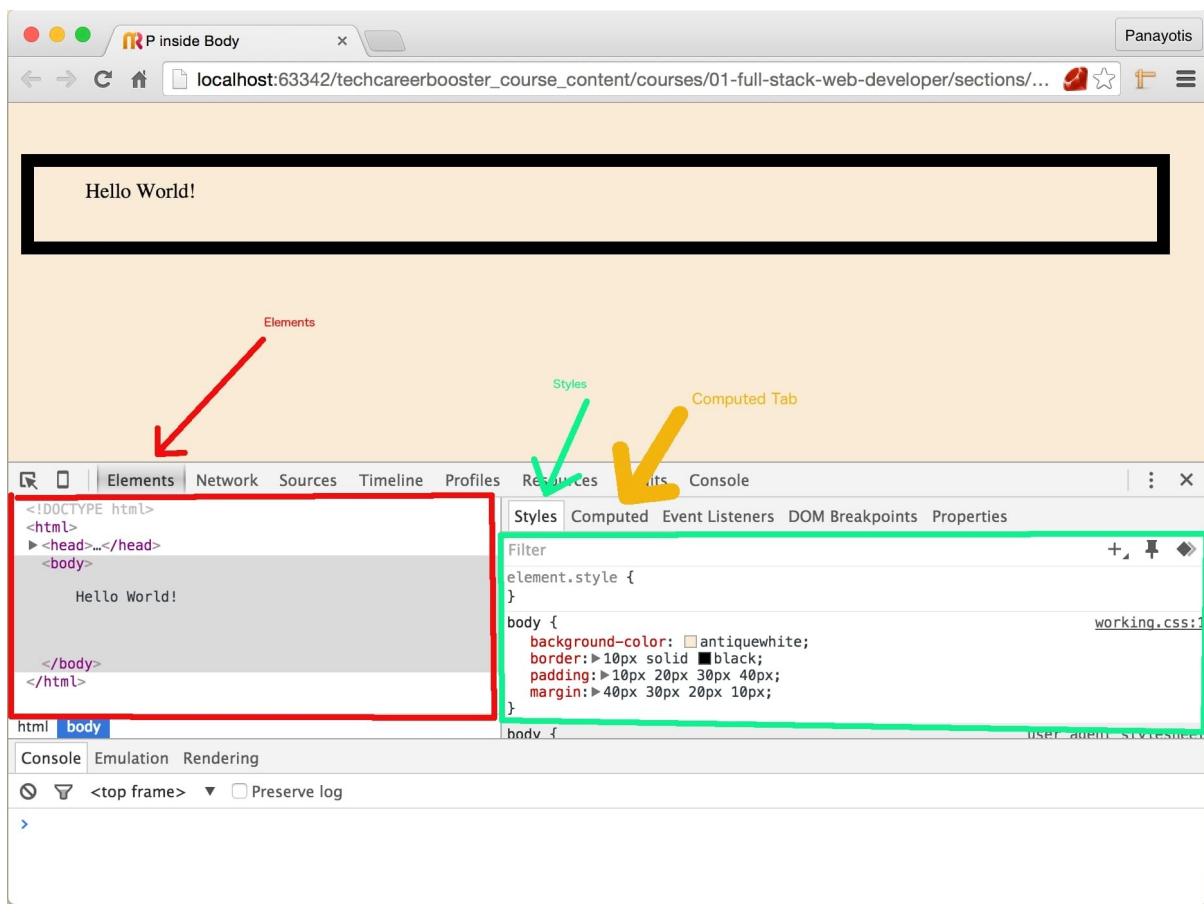
This is a good point that we start learning about Google Developer Tools. When you have a page loaded on Chrome and

- you click on `<kbd>cmd+alt+I</kbd>` combination of keys if you are on Mac, OR
- you click on `<kbd>F12</kbd>` if you are on Windows or Linux OR
- pick up an element on the page, right click on that and select, from the menu, “Inspect Element”

you will start the Chrome developer tools.

This is a very useful tool. We will start using it in order to inspect the actual CSS Box Model of various elements on our page.

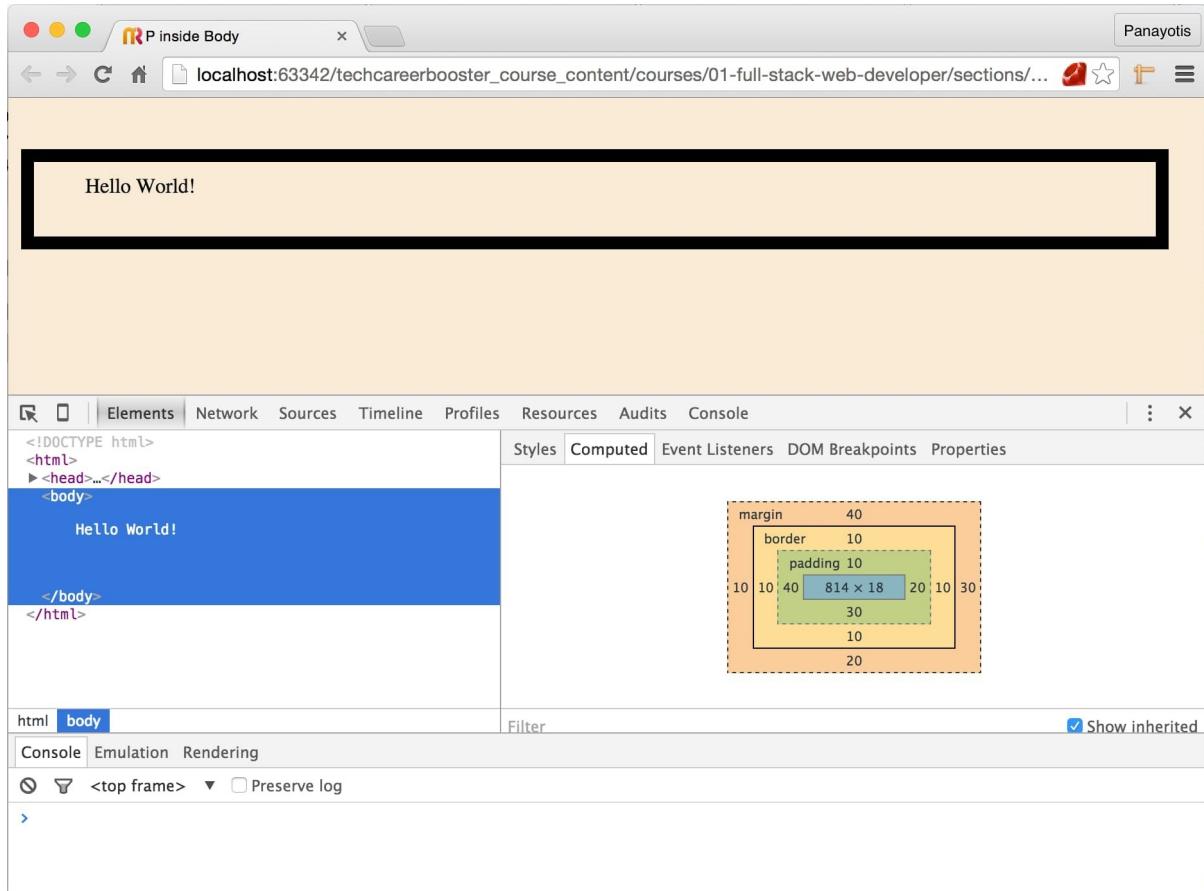
As you can see here...



...in the left area of the developer tools I can see the HTML elements of my page. I can click to select specific elements. On the screenshot, you can see the `<body>` element selected. Further

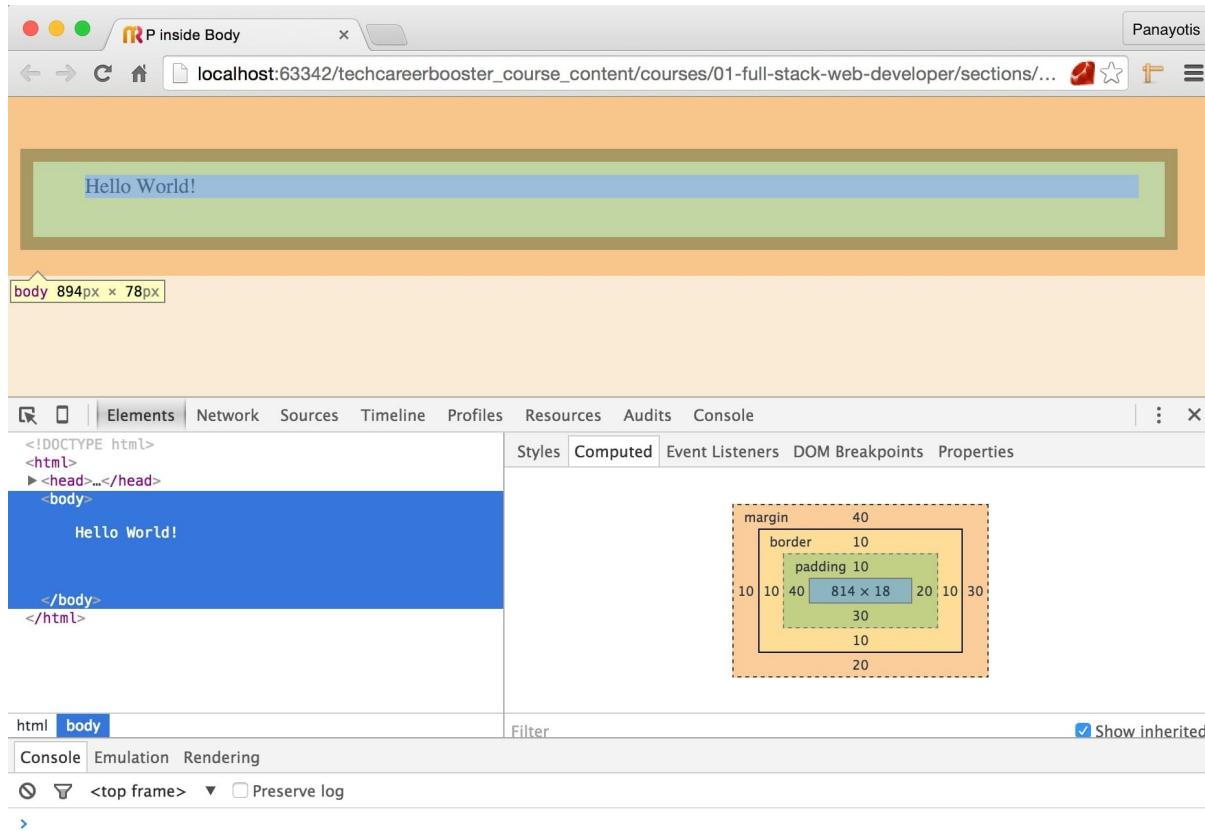
on, on the right, there is an area where I can see the styles of the selected elements. Hence, on this particular screenshot above, you can see the styles of the `<body>` element. Also, there is the “Computed” tab that I am suggesting you to click in order to see the values of all the CSS properties of the selected element, alongside its CSS Box Model.

Here is what the computed tab would look like for our case.



Developer Tools Computed Tab

Also, when you keep your mouse on top of an HTML element, in the “Elements” area, then Chrome highlights the box model on the HTML page itself. See, how I am putting my mouse over the `<body>` element and how Chrome highlights the box model inside the HTML page:



Developer Tools Highlight Box Model Inside HTML Page

Developer Tools is a fantastic tool and absolutely necessary for you to harness. It will be one of your first assistants in your Web developer career. So, spend some time to get acquainted with.

In the following notes of this chapter, as well as in the following chapters we will make heavy use of it.

Note: Firefox and Safari also have developer tools, similar to Chrome.

Width and Box Sizing

As you saw in our previous examples, the width of the body is automatically calculated. But we can still set the width to a specific size. Let's try that. Change the CSS rules as follows:

```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 40px 30px 20px 10px;  
9  
10    width: 400px;  
11}
```

(the above code snippet online)

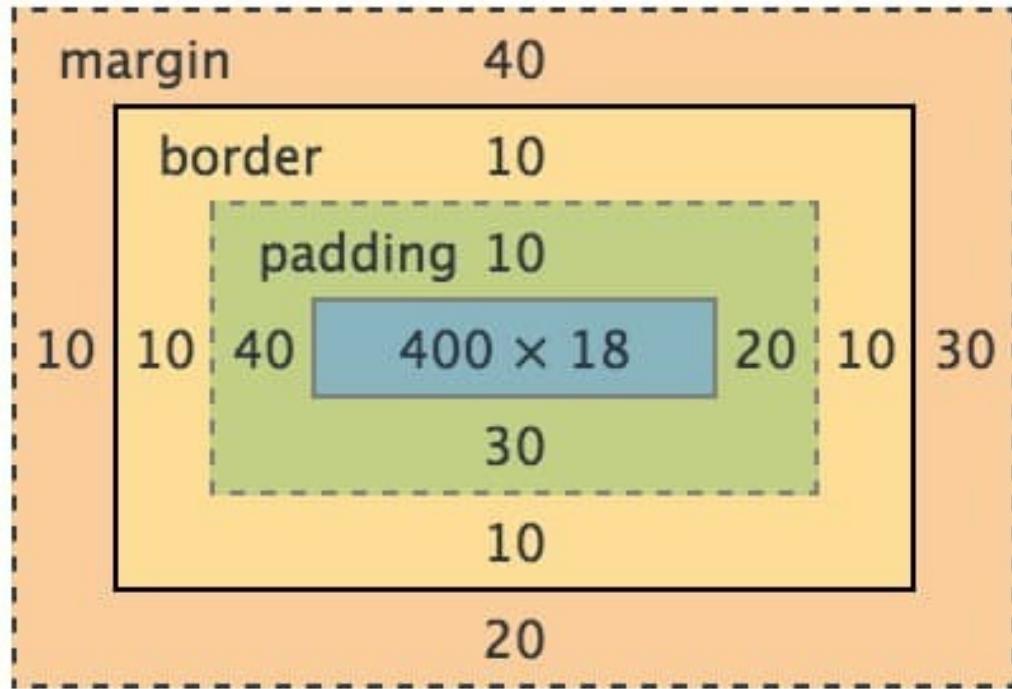
If you reload your page, the result will be something like this:



Body with Specific Width

I have highlighted the box areas, using developer tools, for better understanding.

And another view of this is the following:



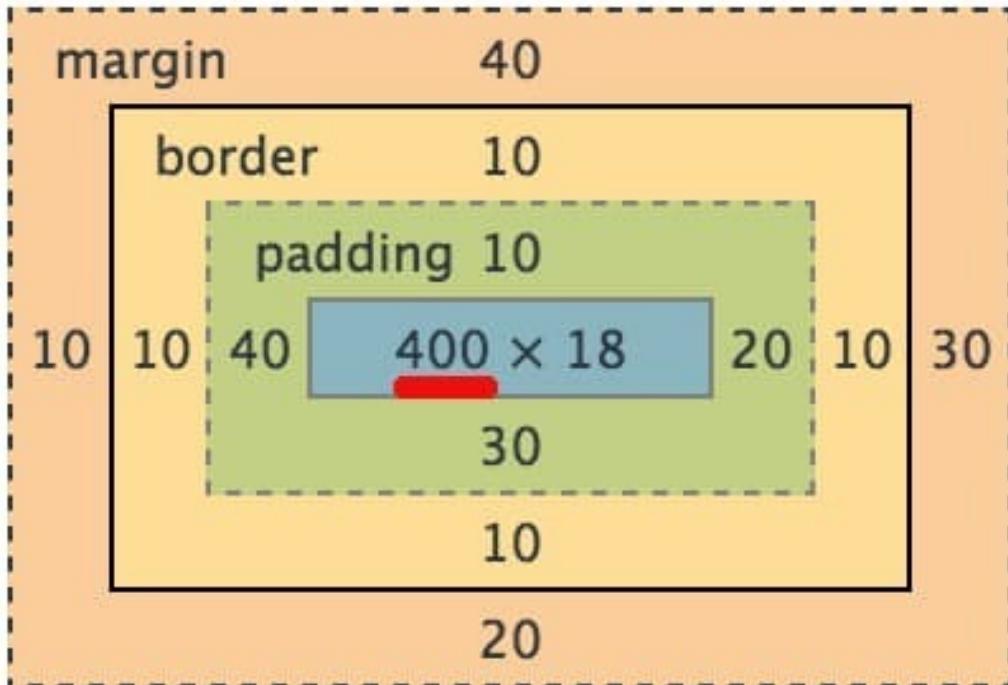
Body Width Another View

There is a small but critical difference between these two views. The first one says that the width of the `<body>` element is 480px.



Body With Specific Width

Whereas the second view mentions that the width is 400px exactly as we have set it up.



Body With Specific Width Another View

Where does this 480px come from? And what does it represent? It represents the **width content + padding-left size + padding-right size + border-left size + border-right size**. If you do the calculations for the example above, you will verify how this 480px is calculated.

Important By default, when you set the width of an element, you specify the width of its content, excluding padding and border sizes.

Let's do something else to verify this rule. Let's change the `width` property from `400px` to `100%`. When we specify the width of an element by percentage, we refer to the actual width of the parent/containing element. Hence, if we specify the width of the body to be `100%` we actually say that we want the width to be equal to the width of the `html` (whole document) element. So, the new CSS rules are:

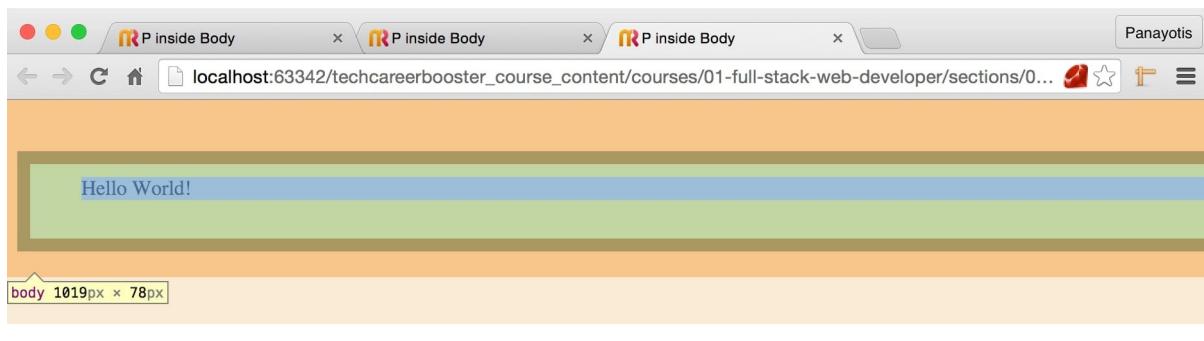
```

1 body {
2     background-color: antiquewhite;
3
4     border: 10px solid black;
5
6     padding: 10px 20px 30px 40px;
7
8     margin: 40px 30px 20px 10px;
9
10    width: 100%;
11 }

```

(the above code snippet online)

If you reload your page again, you will see this:



Hmmmm. Things are not good. Having 100% value for the `width` of `body`, browser draws the content (Hello World!) and then reserves all the rest of the window width for this content. Then browser draws the padding, border and margin, which, unfortunately are clipped and they are not displayed.

Generally, when the `width` refers to the content width rather than to the actual area that the element occupies, things and calculations become tricky and we start feeling uncomfortable with.

There is one way we can overcome this and make sure that when we specify the `width` of an element, we actually refer to the whole area that it occupies. This will make sure that the areas and elements will be put on the page in a much more intuitive way.

We need to specify the `box-sizing` property. The `box-sizing` tells the browser, how it should be interpreting the `width` and `height` properties. The default value of this property is `content-box` and it is what we have experienced until now. The other value that will make things easier to deal with is the value `border-box` and instructs the browser to interpret `width` and `height` taking into account both padding and border sizes.

Note that other properties that are affected by this are:

- `min-width` and
- `min-height`

Important: In any case, the `margin` is not taken into account.

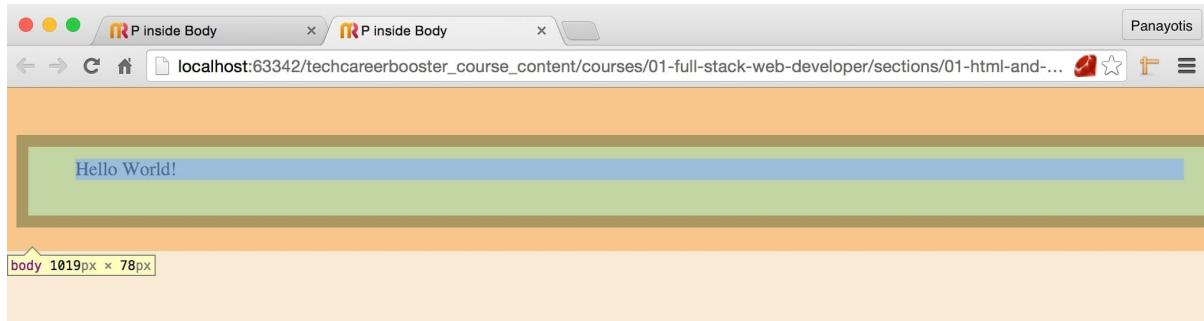
But still, with `border-box` setting on `box-sizing` property things are much easier. Let's see that:

Let's change our CSS to be as follows:

```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 40px 30px 20px 10px;  
9  
10    width: 100%;  
11  
12    box-sizing: border-box;  
13 }
```

(the above code snippet online)

We have added the `box-sizing: border-box;` property. Does that make the things work as expected? If you reload your page you will see this:



Hmmmm. Not what we really wanted. The problem is the left margin. If you have left margin different from 0 together with `box-sizing` set to `border-box` which does not take into account the margin, then the result is not really what we would expect. Try to remove the left margin to see the difference:

```

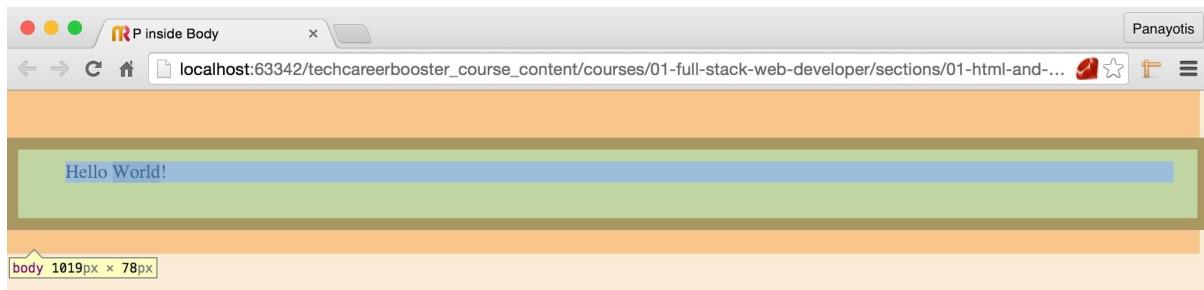
1 body {
2     background-color: antiquewhite;
3
4     border: 10px solid black;
5
6     padding: 10px 20px 30px 40px;
7
8     margin: 40px 30px 20px 0;
9
10    width: 100%;
11
12    box-sizing: border-box;
13 }

```

(the above code snippet online)

Note: When we specify a size to be 0, like the left margin here: `margin: 40px 30px 20px 0;` we do not specify the unit of measurement.

If you reload this page the result will be this:



Body with Margin Left 0

Things are much better but the right border extends beyond the right margin. Do you see that on the right of the browser window? This is not what we really expected. Browser is a little bit confused. Again, the problem is the right margin that does not “play” well with the `box-sizing` property set to `border-box`. Let’s remove right margin too:

```

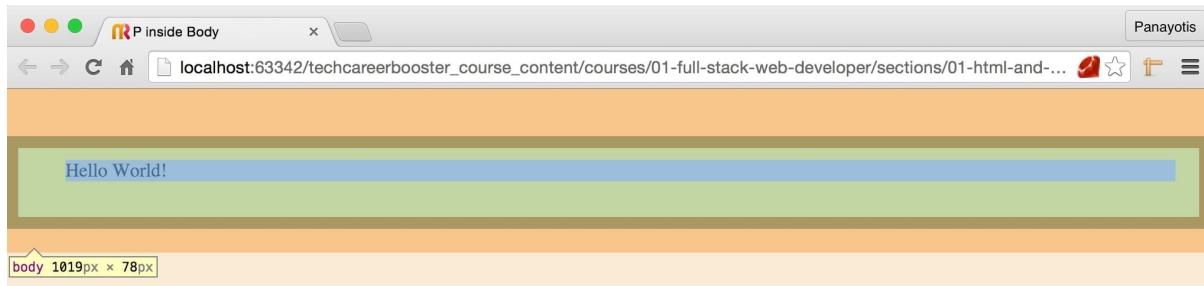
1 body {
2     background-color: antiquewhite;
3
4     border: 10px solid black;
5
6     padding: 10px 20px 30px 40px;
7
8     margin: 40px 0 20px 0;
9
10    width: 100%;
11

```

```
12     box-sizing: border-box;  
13 }
```

(the above code snippet online)

If you reload your page you will see this:



Body Margin Left and Right Zero

That's much better.

The same problems that we saw for the `width` property are applied to `height` property when we use `box-sizing border-box` and we have top and bottom margins.

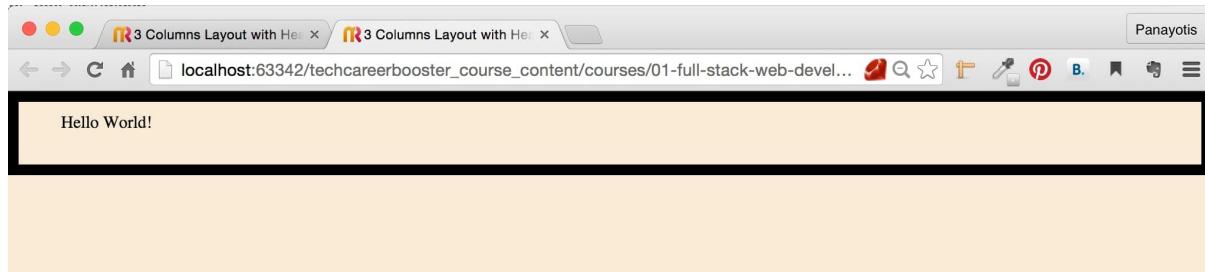
Having said that, we change our CSS file to be:

```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 0 0;  
9  
10    width: 100%;  
11  
12    box-sizing: border-box;  
13 }
```

(the above code snippet online)

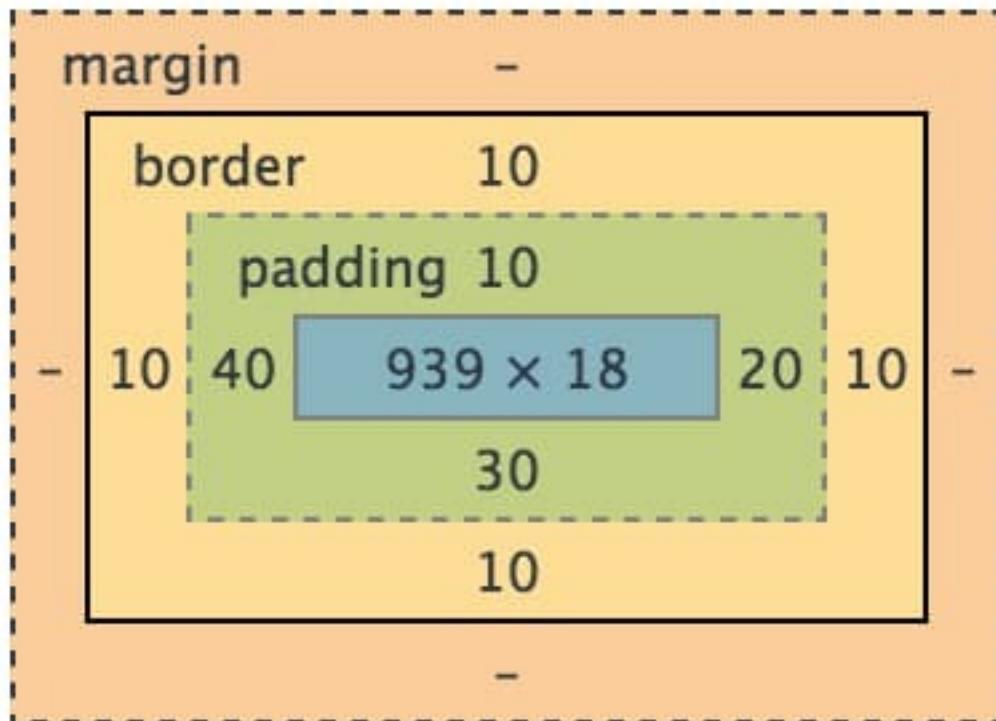
Remember that default margin for body is 8px. That's why we have to 0 it.

If you reload your page, you should see something like this:



Body Margin No

And an alternative view:



Body Margin No Alternative View

What if I really want to have margins around my border? How would I do that?

Then you have to set your margins and remove the width value requirement of 100% completely. It will be automatically calculated.

Let's do that:

```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 10px 10px;  
9  
10    box-sizing: border-box;  
11 }
```

(the above code snippet online)

If you reload the page it will be like this:



Body With Margins But No Width Value

Very nice!

In summary, our recommendation is this:

1. Use `box-sizing: border-box;`. It will make things much more predictable when you set values for `width` and `height`.
2. If you want to use margins, then you should not set the corresponding size properties.

Line Height

There is a CSS property which is called `line-height` and we will now spend some time on that. In order to do that we will work with some text and see how the line height is reserved by the browser in order to print that text on the page.

The `line-height` property has enough value so that the characters of text can fit in. This means that the browser, with `line-height`, reserves enough space to print characters that occupy much height, like characters that are uppercase and have accent (these characters have **ascender portion**), like this: O and characters that expand below the baseline (these characters have **descender portion**), like these jyqq.

Write the following HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Line Height Demonstration</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <p>Text without descenders or ascenders.</p>
12         <p>Text with descenders. Look at the characters "g", "j", "q" and "y": game\
13 , joke, query</p>
14         <p>Text with ascenders. Look at these characters "O&#770;", "A&#771;"</p>
15         <p>Text with ascenders and descenders: gjqyO&#770;A&#771;</p>
16
17     </body>
18
19 </html>
```

(the above code snippet online)

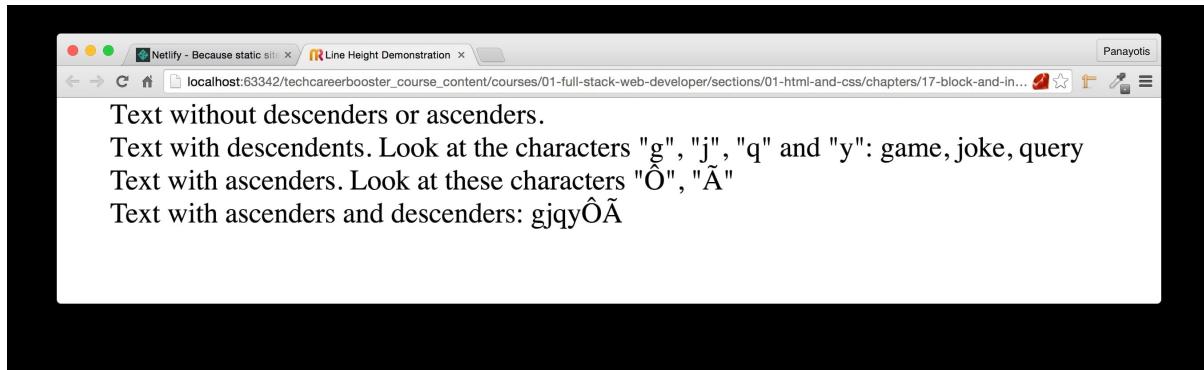
Also, create the `stylesheets/main.css` file, whith the following content:

```
1  * {
2      box-sizing: border-box;
3      margin: 0 20px;
4      padding: 0;
5  }
6
7  body {
8      font-size: 32px;
9  }
```

(the above code snippet online)

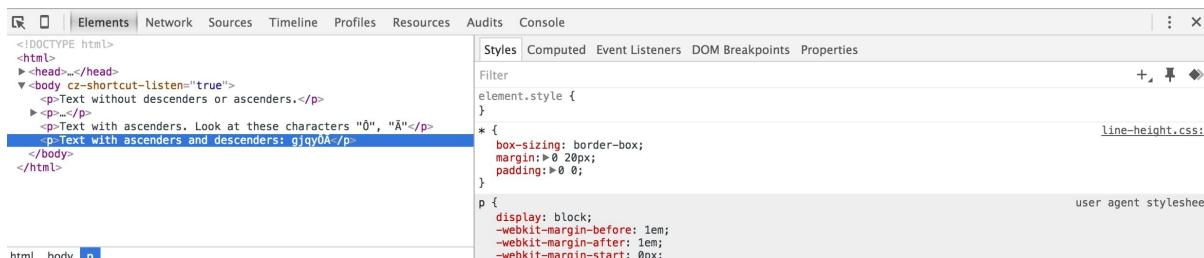
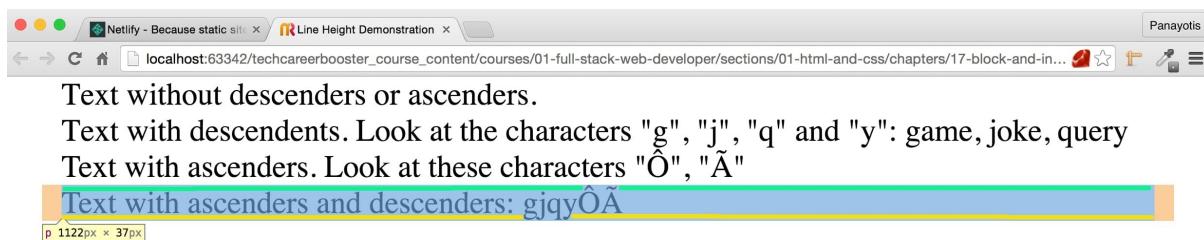
which will make sure that we do not have any top or bottom margin neither padding, and we will have large font size to make things more clear. (Actually, we have removed any left and right padding too, but this is irrelevant to our discussion here.)

Load the `index.html` page on your browser. You should see the following result:



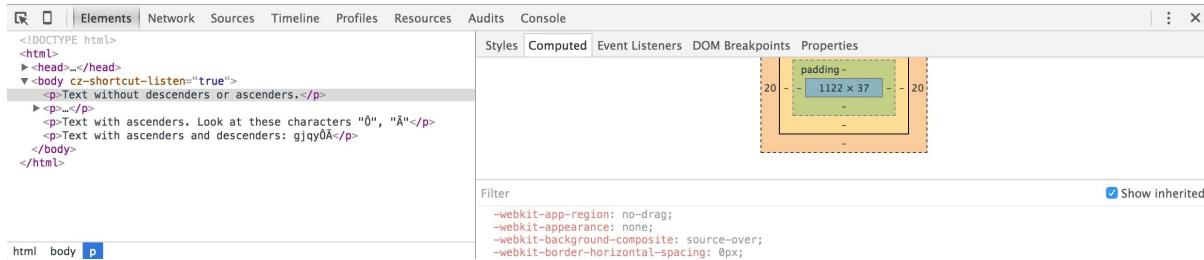
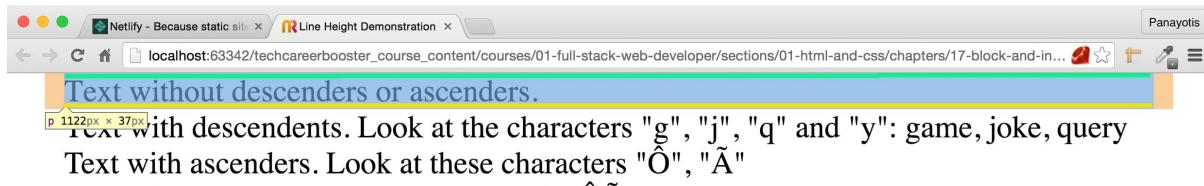
Text with Ascenders and Descenders

As you can see, one line does not start immediately below the previous one. Each line has enough height to accommodate the accents and the descenders. Inspect with the developer tools the last line. Open the developer tools and inside the “Elements” area, put your mouse over the last `<p>` element. Do you see the blue highlight? It depicts the exact line height. And you can see that it is enough to hold the ascenders and descenders of that line.



Line Height Mark Ascenders and Descenders Space

We have marked the ascenders reserved space with the light green color, and the descenders reserved space with the yellow color. This demonstrates how browser reserves the space for these cases and uses it when needed. Note that this space exists and affects the line height even if the text does not have such characters (ascenders/descenders). See how this has been done in the first line:



Text with Ascenders and Descenders First Line

You can see above, the light green color demonstrating the reserved space for ascenders, although this text line does not have any. Same goes for the descenders.

So, keep that in mind, that line height is not equal to the height of the tallest character inside your text, but it is equal to the height that is required to accommodate the tallest character with the font that you are using to write your text. Even if you are not using ascenders neither descenders characters, the reserved space will be there.

Inherited and Non-Inherited values

Let's try another example that contains a `<p>` element inside the `<body>` element. Our CSS is as:

```

1 body {
2   background-color: antiquewhite;
3
4   border: 10px solid black;
5
6   padding: 10px 20px 30px 40px;
7
8   margin: 10px 10px;
9
10  box-sizing: border-box;
11 }
```

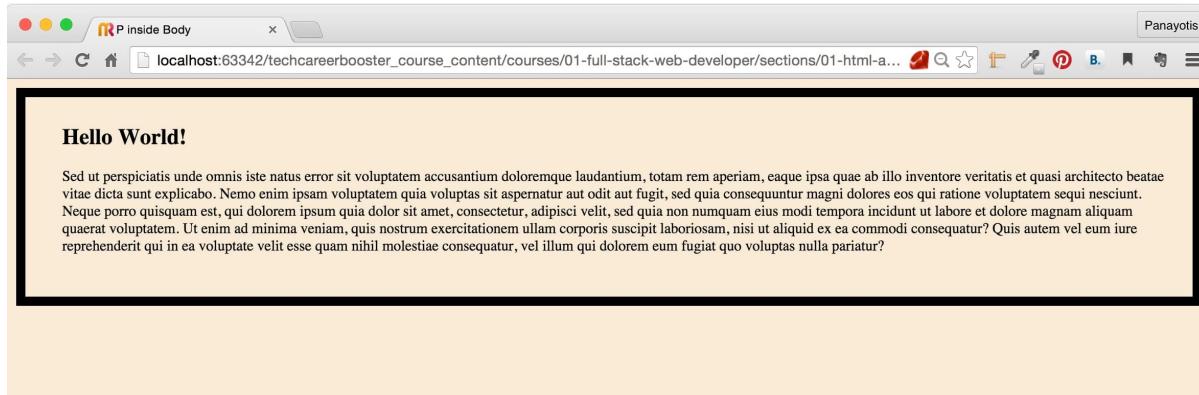
(the above code snippet online)

and our HTML page now is:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>P inside Body</title>
6          <link rel="stylesheet" href="stylesheets/index.css">
7      </head>
8
9      <body>
10         <h2>Hello World!</h2>
11
12         <p>
13             Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium \
14 doloremque laudantium, totam rem aperiam,
15             eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vita\
16 e dicta sunt explicabo. Nemo enim ipsam
17             voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia conseq\
18 uuntur magni dolores eos qui ratione voluptatem
19             sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit \
20 amet, consectetur, adipisci velit, sed quia non
21             numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaer\
22 at voluptatem. Ut enim ad minima veniam,
23             quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut ali\
24 quid ex ea commodi consequatur? Quis autem
25             vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molest\
26 iae consequatur, vel illum qui dolorem eum
27             fugiat quo voluptas nulla pariatur?
28         </p>
29
30     </body>
31
32
33 </html>
```

(the above code snippet online)

If you load this page on your browser you will see something like this:



p Inside body

Let's do the following changes inside our CSS file:

```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 10px 10px;  
9  
10    box-sizing: border-box;  
11  
12    color: darkblue;  
13}  
14  
15 p {  
16     width: 400px;  
17     padding: 10px 10px;  
18     border: 3px solid red;  
19}
```

(the above code snippet online)

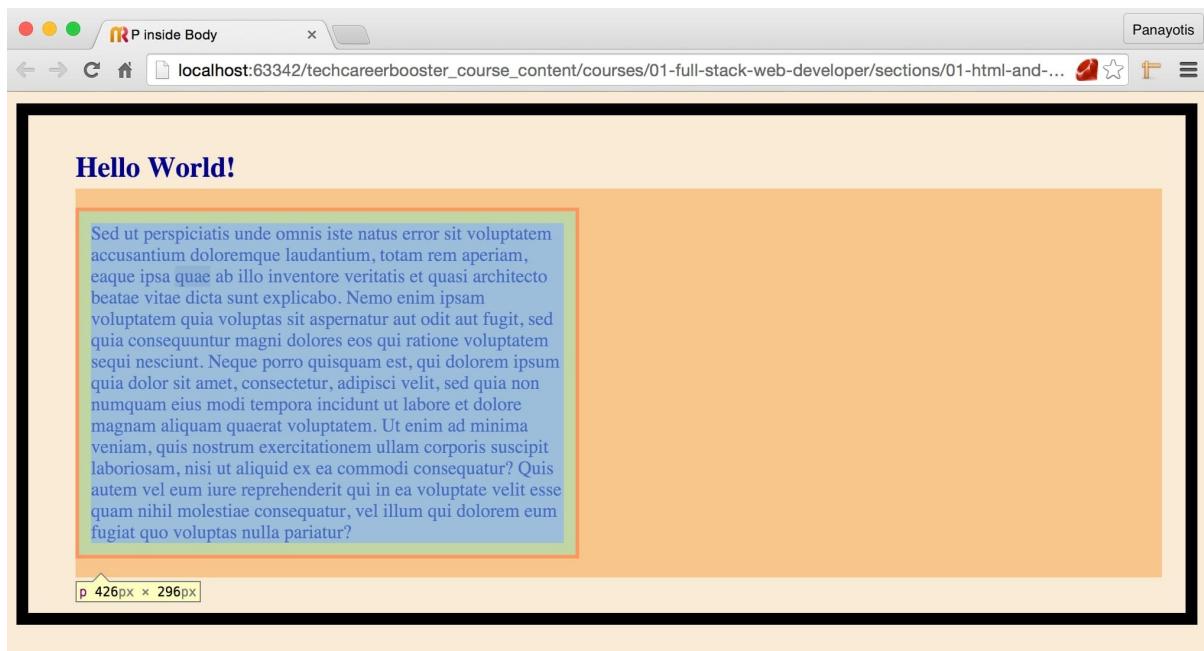
What we have done was to set the color of the text inside body to be darkblue and we have also set some attributes for the p element.

If you reload your page, the result will be:



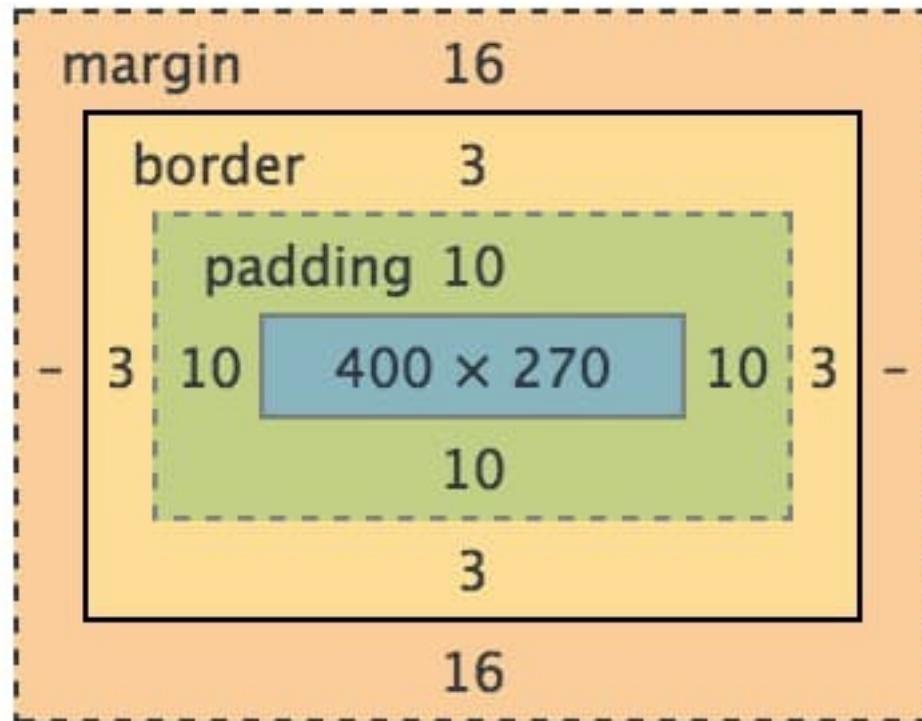
p Inside body With Some Attributes Changed

We can clearly see that the width of the p element has changed. If we get another view that highlights the areas of the p (using developer tools) you will see:



p Inside body Highlight View

Hmmmm. It says that the width is 426px. Let's see another view of its box model, using developer tools again.



p Inside body Box Model View

We can see from the above that the width property refers to the content of the `<p>` element (400px). However, we can see that text color is the same for both elements (`<h1>` and `<p>`) that are inside the `<body>` element.

This small experiment shows that some of the CSS properties have been inherited from the containing element to the contained element, i.e. from `<body>` to `<h1>` and `<p>`, like the `color`, but others, like `margin`, `border` and `box-sizing` have not been inherited.

If you want one element to inherit the CSS property of parent element, then you have to set the value `inherit` to the particular attribute. See this change here:

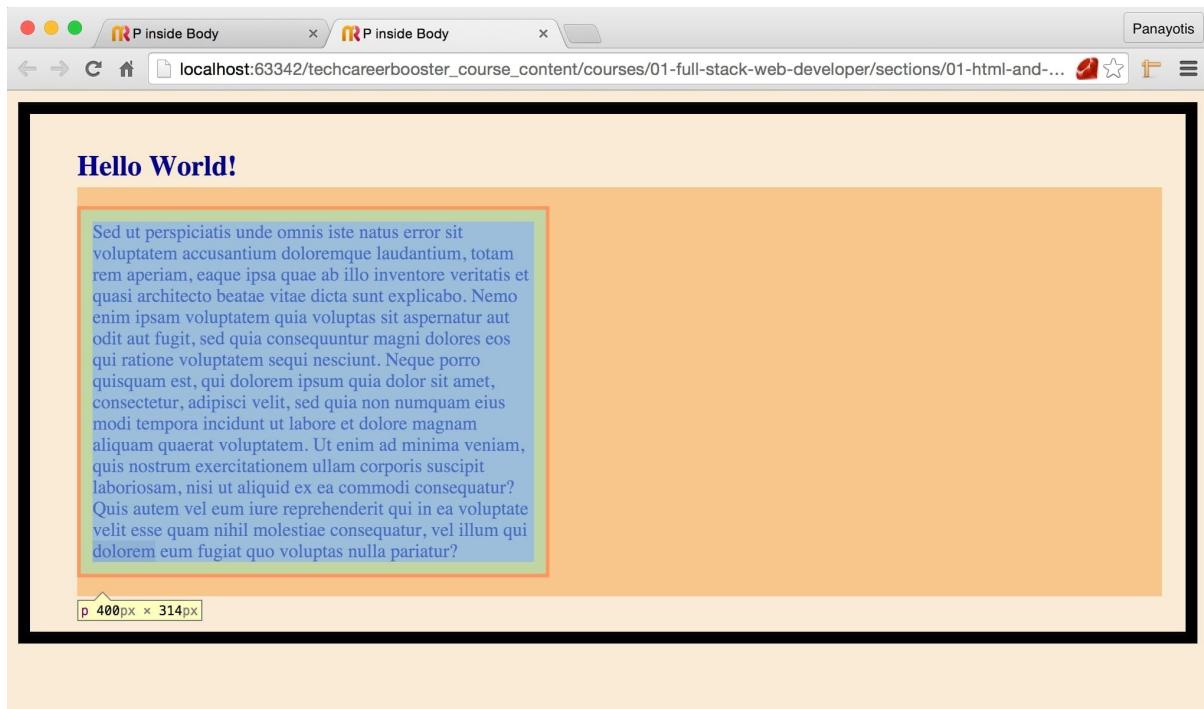
```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 10px 10px;  
9 }
```

```

10    box-sizing: border-box;
11
12    color: darkblue;
13 }
14
15 p {
16    width: 400px;
17    padding: 10px 10px;
18    border: 3px solid red;
19
20    box-sizing: inherit;
21 }
```

(the above code snippet online)

We have added the property `box-sizing` with the value `inherit` to the `<p>` element. Let's reload the page and see how this changes things. You can see now that the actual width of the `<p>` element is 400px meaning that the `<p>` element has inherited the `box-sizing` property from the parent `<body>` element.



Box Sizing Inherited Highlight View

Also, the developer tools can give you the actual properties that are used for the `<p>`. You only have to go to “Computed” tab and see the values of the properties. For example, there, you will see the value of the `box-sizing` property to be `border-box`. Also, the color will be `rgb(0, 0, 139)` which is the `rgb` representation of the `darkblue` color.

```
border-top-width: 3px,  
box-sizing: border-box;  
color: #rgb(0, 0, 139);  
display: block;  
height: 314px;  
...  
...
```

Computed Tab Shows Box Sizing and Color

CSS properties are numerous. Some of them are automatically inherited to contained elements, some not. Usually, properties that change/affect the structure of the page, like margins, paddings, borders e.t.c. are not inherited. Whereas others that affect the format of the content, like color and font-size are automatically inherited. You will learn to identify them by experience. Of course, there are plenty of resources on the Internet that can tell you whether a property is inherited or not.

Applying Rule To All Elements

What if you wanted all your elements to work with the `box-model` value for `box-sizing` property? In that case, you do not have to explicitly set the value to each one of them. You only have to set it to the root element and then specify that all elements will be inheriting this property.

In our example, this would do the trick:

```
1 body {  
2     background-color: antiquewhite;  
3  
4     border: 10px solid black;  
5  
6     padding: 10px 20px 30px 40px;  
7  
8     margin: 10px 10px;  
9  
10    box-sizing: border-box;  
11  
12    color: darkblue;  
13 }  
14  
15 body * {  
16     box-sizing: inherit;  
17 }  
18  
19 p {
```

```

20     width: 400px;
21     padding: 10px 10px;
22     border: 3px solid red;
23 }

```

(the above code snippet online)

Note that we have removed the `box-sizing` setting from the `p` element. And we have added a rule for `body *` which essentially means for any child of the `body` element.

If you reload your page and inspect both `<h1>` and `<p>` elements, you will see that both of them have `box-sizing` property set to `border-box`.

The screenshot shows a browser window with the title "Hello World!". Inside the window, there is a paragraph of Latin text. Below the browser window is the developer tools interface. The element tree shows "html", "body", and "p". The "p" element is highlighted with a red circle. The styles tab in the developer tools shows the following CSS rules:

```

element.style {
}
p {
    width: 400px;
    padding: 10px 10px;
    border: 3px solid red;
}
body * {
}

```

The "body * {" rule includes the following properties:

```

border-right-style: solid;
border-right-width: 3px;
border-top-color: #rgb(255, 0, 0);
border-top-style: solid;
border-top-width: 3px;
box-sizing: border-box;
color: #rgb(0, 0, 139);
display: block;

```

Box Sizing Inherited for p Element

The screenshot shows a browser window with a red box highlighting the text "Hello World!" and a red circle around the "h2" element in the DOM tree. The developer tools interface includes tabs for Elements, Network, Sources, Timeline, and Properties. The Styles tab is active, showing the "user agent stylesheet" rules for the "h2" element. These rules include `display: block;`, `font-size: 1.5em;`, and `color: #00008B;`. To the right, the "Properties" tab displays the inherited properties from the "body" element: `box-sizing: border-box;`, `color: #00008B;`, `display: block;`, `font-size: 24px;`, and `font-weight: bold;`. A red box highlights the `box-sizing` rule in the properties panel.

```

h2 {
    user agent stylesheet
    display: block;
    font-size: 1.5em;
    -webkit-margin-before: 0.83em;
    -webkit-margin-after: 0.83em;
    -webkit-margin-start:
}

```

Box Sizing Inherited for h1 Element

This is because they have inherited that from body thanks to the rule:

```

1 body * {
2     box-sizing: inherit;
3 }

```

(the above code snippet online)

Other ways you can achieve the same trick is the following:

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     background-color: antiquewhite;
7
8     border: 10px solid black;
9
10    padding: 10px 20px 30px 40px;
11
12    margin: 10px 10px;
13
14    color: darkblue;
15 }
16
17 p {
18     width: 400px;
19     padding: 10px 10px;
20     border: 3px solid red;
21 }
```

(the above code snippet online)

You can see here that we have created a very generic rule that applies to all elements, including body:

```
1 * {
2     box-sizing: border-box;
3 }
```

(the above code snippet online)

Hence, we have removed the box-sizing settings from the other elements. Now all the elements have the box-sizing setting to border-box;

Why body background-color covers the whole canvas?

We have seen at the beginning of the chapter that the background-color of the body element covered the whole canvas background color, the whole page. Why was that?

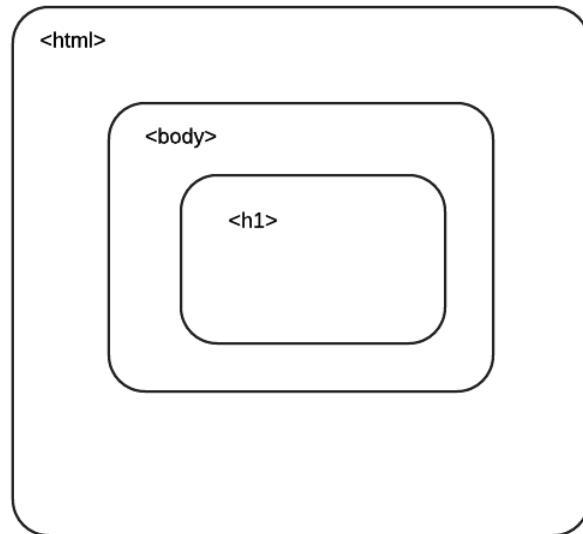
Generally, this does not happen. In other words, the style properties of an element do not propagate to parent elements.

Let's see the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Body with h1 - Background Colors</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         body
11         <h1>Hello World!</h1>
12
13     </body>
14
15 </html>
```

(the above code snippet online)

As you can see in the HTML code above, the hierarchy of elements is as follows:



Hierarchy Of Elements

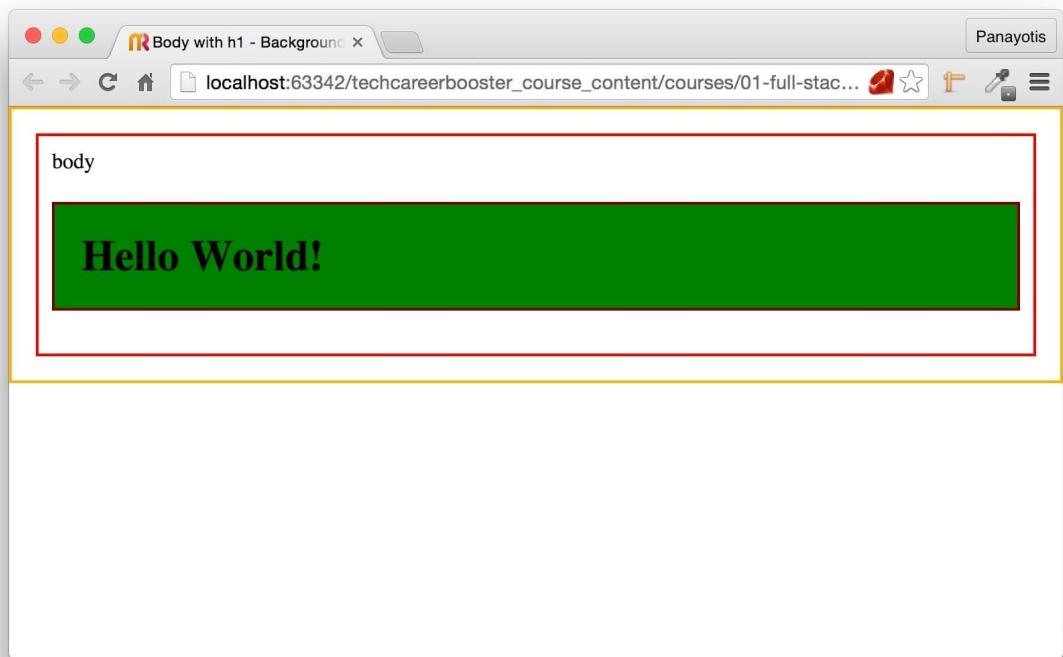
We repeat that the CSS properties do not propagate from child to parent. Hence, if we set the background-color of the h1 element this stays within it.

Try this CSS file:

```
1 html {  
2     padding: 10px;  
3     border: 2px solid #ffb21a;  
4 }  
5 body {  
6     padding: 10px;  
7     border: 2px solid red;  
8 }  
9  
10 h1 {  
11     background-color: Green;  
12     padding: 20px;  
13     border: 2px solid Maroon;  
14 }
```

(the above code snippet online)

It sets the background-color of h1 to green. Also it applies some padding and border to both html, body and h1 so that we can see that the background-color stays within h1 and so that we can also see the boundaries of the elements. If you save this file (stylesheets/main.css) and you load the HTML page on your browser, you will see the following:



Background Color of h1 stays within it

This proves that the background of the child (h1) stays within it. Does not propagate to the parent body element.

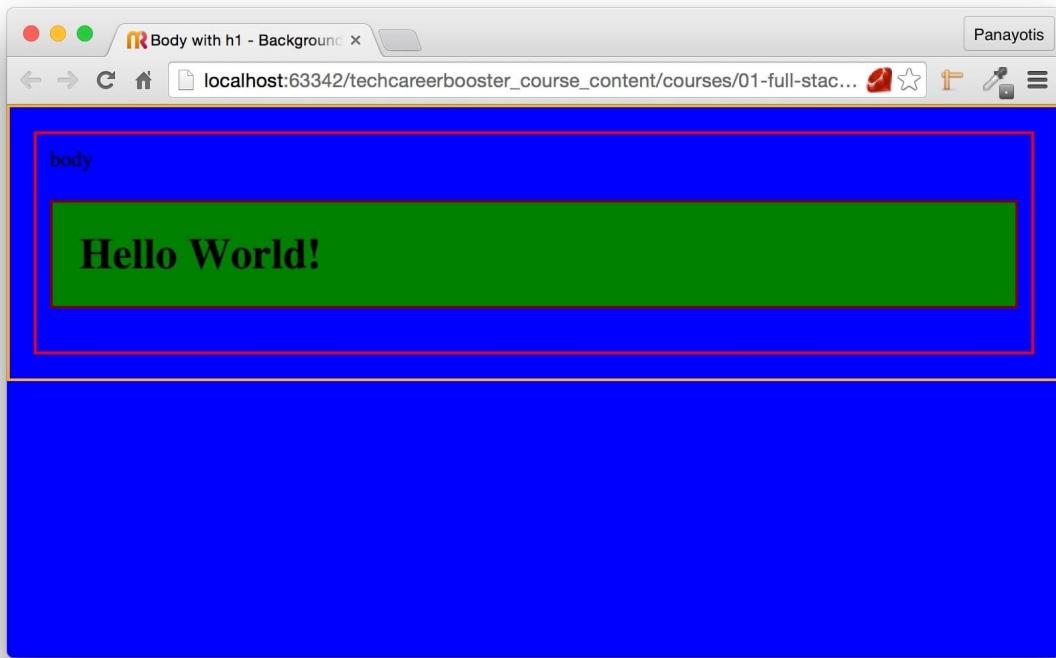
There is, however, an exception to this rule. The background properties (including color) of the body element, propagate to the parent element. The parent element of the body element is the html element. You can see that on the previous example. It is the orange-border surrounded area.

Let's see that. We apply background-color Blue to the body element:

```
1  html {
2      padding: 10px;
3      border: 2px solid #ffb21a;
4  }
5  body {
6      background-color: Blue;
7      padding: 10px;
8      border: 2px solid red;
9  }
10
11 h1 {
12     background-color: Green;
13     padding: 20px;
14     border: 2px solid Maroon;
15 }
```

(the above code snippet online)

Save the CSS file and reload the page. What you will see is the following:



Background Color of body Propagates to html

So, the `background-color` is propagated to the `html` element, and becomes the `background-color` of the `html` element. **But not only that.** It seems that the propagation didn't stop on the `html` element. It has been propagated to the whole canvas, the browser window drawing area.

Important. The `background` properties of the `root` element are propagated to the whole canvas by **design**. This is how it has been specified and this is how it works.

That is why you see the whole canvas being blue.

Finally, if you want the `html` to have different `background-color` from the `body`, then just assign a different `background-color` to it. For example, try `background-color: white;` to the `html` element. If you reload the page, you will see this:



html background color

As you can see, now the `html` color has its own `background-color` and does not use the `background-color` of `body` element. However, the rule for the root element still applies. I.e. the `background-color` of the `html` element (which is the root element) floods the canvas area.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Using Chrome visit the page “www.google.com”.
2. Inspect the following elements. Take a screenshot of their CSS Box Model as it is presented by the developer tools. For each one of the boxes, make sure you write down margin, border, padding and content sizes.
 1. The Google doodle image
 2. The Google search box area.
 3. The Google Search box button.

4. The Google “I’m feeling lucky” button.
3. Implement the following page:

Shopping List

- Cheese
- Rice
- Coffee
- Milk
- Wine

CSS Box Model Shopping List

Make sure that:

1. The page has 2 elements, besides the `<body>`. An `<h1>` and an unordered list ``.
2. The `body` has the following properties when inspected with developer tools:



A screenshot of the Chrome DevTools Elements tab. The left pane shows the DOM tree:

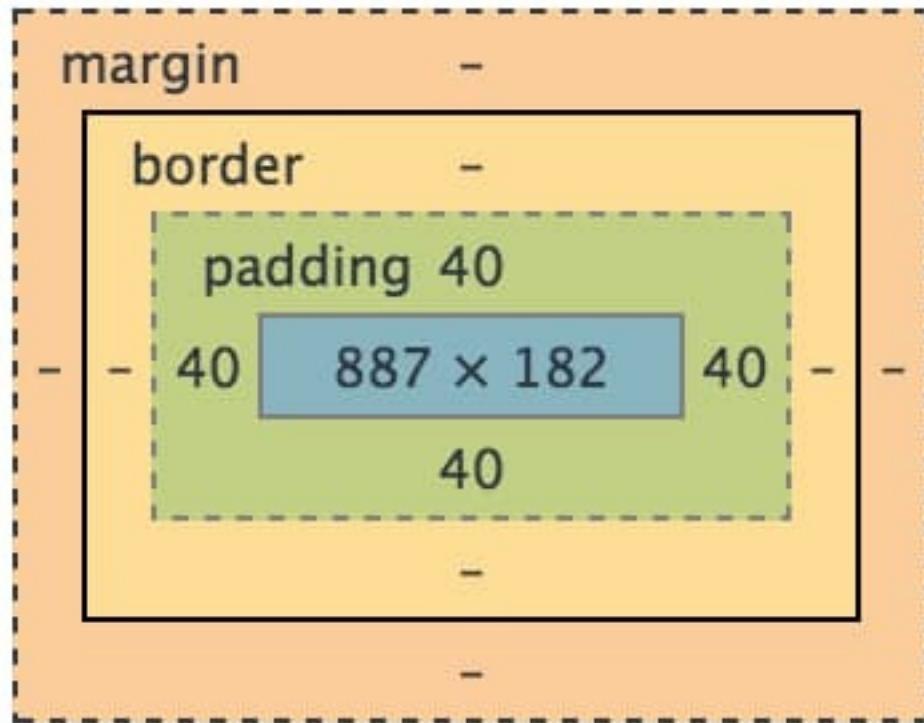
```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body cz-shortcut-listen="true">
    <h1>Shopping List</h1>
    <ul>...</ul>
  </body>
</html>
```

The right pane shows the Styles tab with the following CSS rules:

```
element.style { }
body {
  box-sizing: border-box;
  margin: 0;
  padding: 40px 40px;
}
body {
  display: block;
  margin: 8px;
}
```

The "task.css:1" label indicates the source of the first rule. The "user agent stylesheet" label indicates the source of the second rule. The status bar at the bottom shows the selected elements are "html", "body", and "h1".

Shopping List body element styles



Shopping List body element computed tab

3. The h1 has the following properties when inspected with developer tools:

The screenshot shows a browser window with the title "Task for Box Model". The address bar shows the URL "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-developer/section-1/task-for-box-model". The main content area displays the text "Shopping List" followed by an unordered list:

- Rice
- Coffee
- Milk
- Wine

In the bottom right corner of the content area, there is a small "task.css" icon.

The developer tools sidebar shows the "Elements" tab selected. In the "Styles" panel, the computed styles for the h1 element are listed:

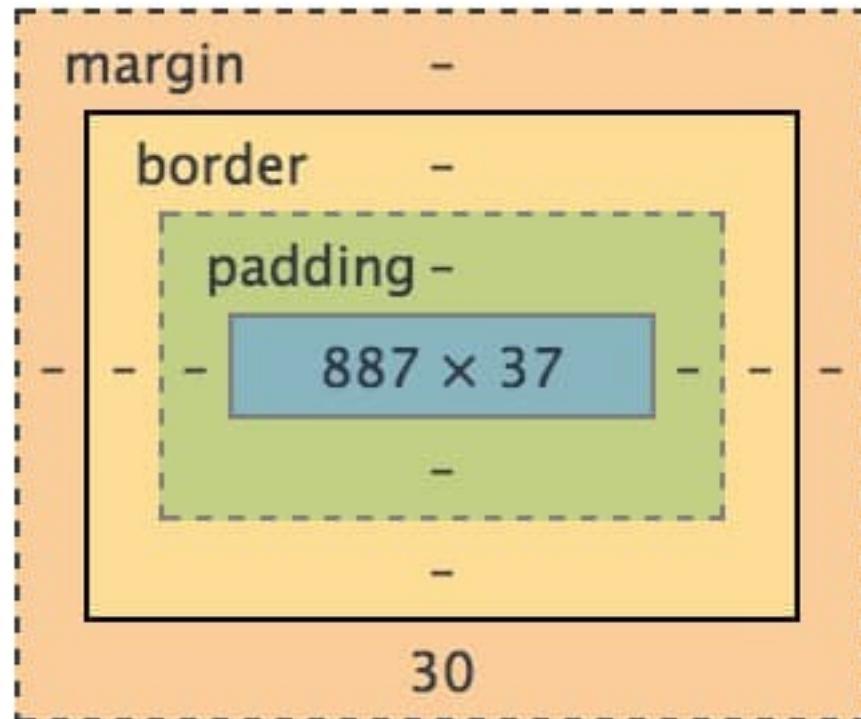
```

h1 {
    margin-bottom: 30px;
}

```

The "task.css:13" and "task.css:8" labels indicate the source files for these rules.

Shopping List h1 element styles



Shopping List `h1` element computed tab

4. The `ul` has default style as applied by the browser.
5. The `li` inside the `ul` has the following properties when inspected with developer tools:

The screenshot shows a browser window titled "Task for Box Model" displaying a "Shopping List" page. The list contains the following items:

- Cheese
- Coffee
- Milk
- Wine

The item "Cheese" is highlighted with a blue background. Below the browser window is the Chrome DevTools interface, specifically the "Elements" tab. The DOM tree shows the HTML structure:

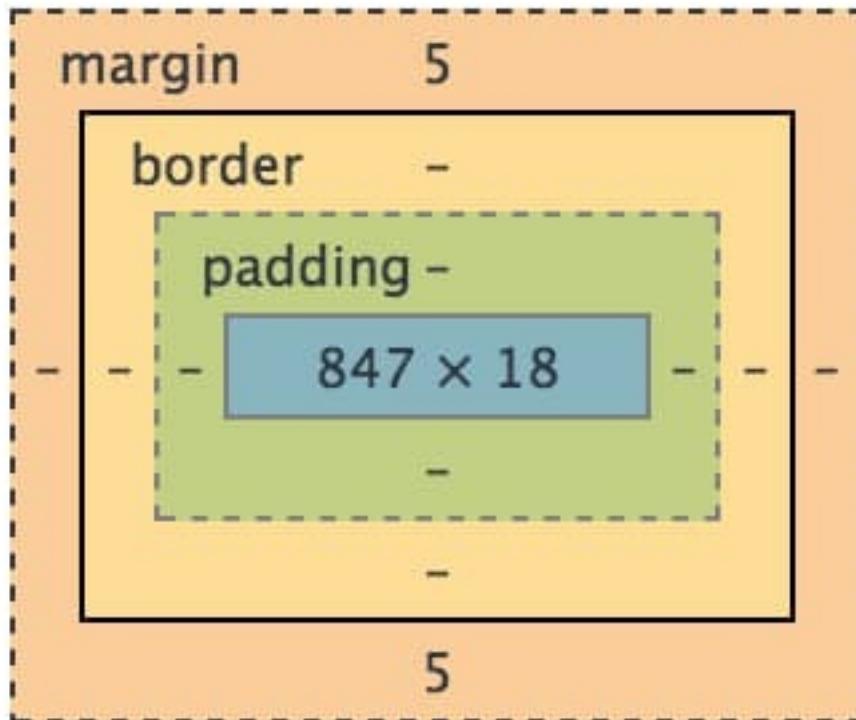
```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body cz-shortcut-listen="true">
    <h1>Shopping List</h1>
    <ul>
      <li>Cheese</li>
      <li>Rice</li>
      <li>Coffee</li>
      <li>Wine</li>
    </ul>
  </body>
</html>
```

The "Styles" panel on the right shows the computed styles for the selected `li` element:

```
element.style {
}
ul li {
  margin: 5px 0;
}
```

The file `task.css:17` is indicated as the source for the `margin` style.

Shopping List li element styles



Shopping List li element computed tab

6. All elements should have `box-sizing` property equal to `border-box`. However, do not set

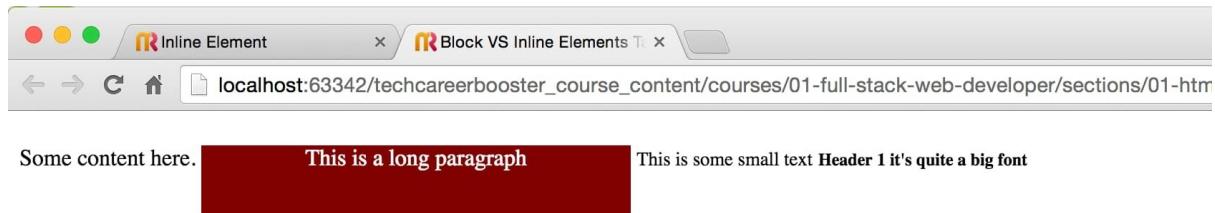
this property to each one of them literally. Use a technique that would allow you to write this rule with as little CSS coding as possible.

Chapter 17 - Block And Inline Elements

Summary

In this chapter, we explain the difference between block and inline elements. This is a fundamental concept, that will allow you to understand how elements are organized on the HTML page.

You are going to build a simple page like this:



Page Demonstrating Block vs Inline elements

Learning Goals

- Learn about the difference between `block` and `inline` elements.
- Learn about the `display` style property.
- Learn about the default HTML `block` elements.
- Learn about the default HTML `inline` elements.
- Learn how you can specify the width and height of an inline element.

Block and Inline HTML Elements

Now that we have learned about the CSS Box Model and how we can visualize the area that element occupies, using the developer tools, it's a good point in time to talk about the `block` vs `inline` elements.

Block HTML Elements

A block HTML element always starts at a new line and takes the full width available, from left to right, as much as it can. The available width of a block HTML element is the width that its parent (containing element) allows it to have.

Let's take for example the following HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Block VS Inline Elements</title>
6      </head>
7
8      <body>
9          Text before &lt;p&gt;<p>Lorem ipsum</p>Text after &lt;p&gt;
10         </body>
11
12     </html>
```

(the above code snippet online)

If you load this page on your browser, you will see this:

Text before <p>
Lorem ipsum
Text after <p>

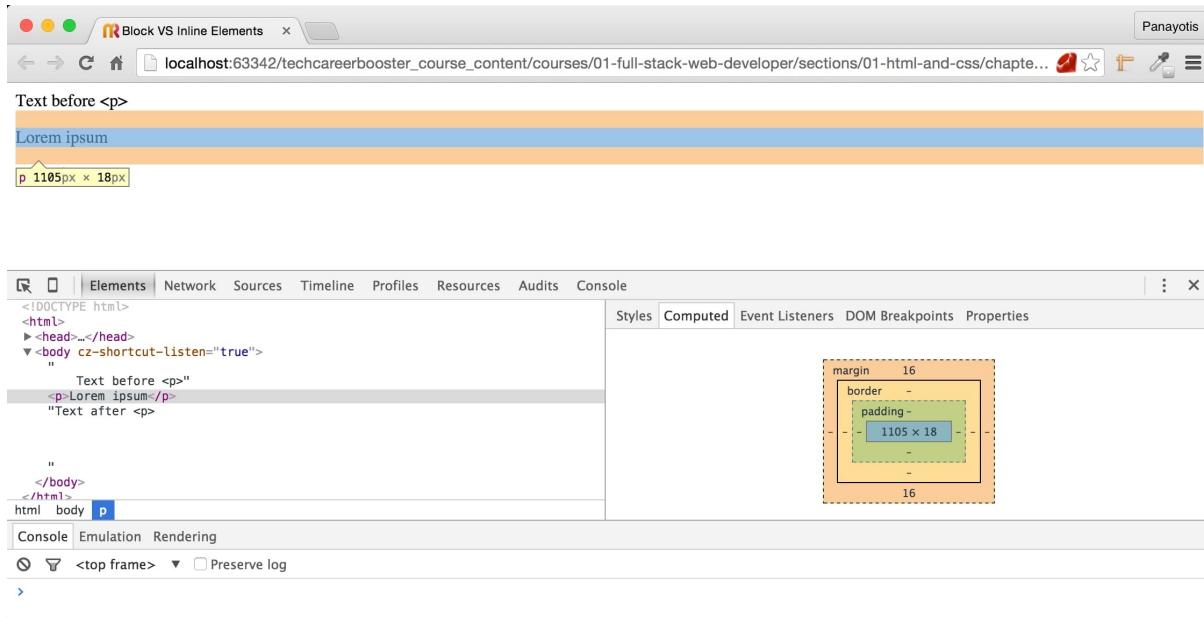
lines before and after <p>



Block VS Inline P and Text Before and After

Do you see that there are empty lines before and after the p element?

If you also inspect the p element using developer tools, you will see the following:



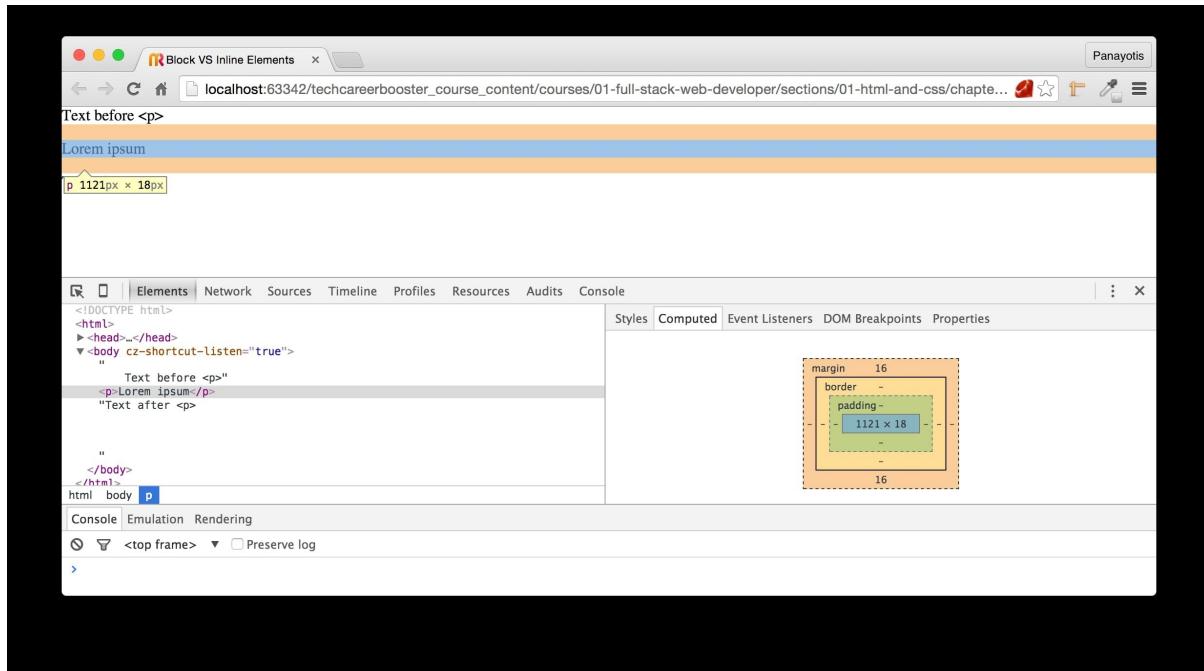
Block vs inline p inspected

As you can see, the `<p>` element occupies the whole width that has available. Note that the parent element of this `<p>` is the `<body>`, which, by default, has left and right margins. That's why you see some free white space left and right to the `<p>`. If we remove the `<body>` margins, then we will have this:

```
1 body {
2     margin: 0 0;
3 }
```

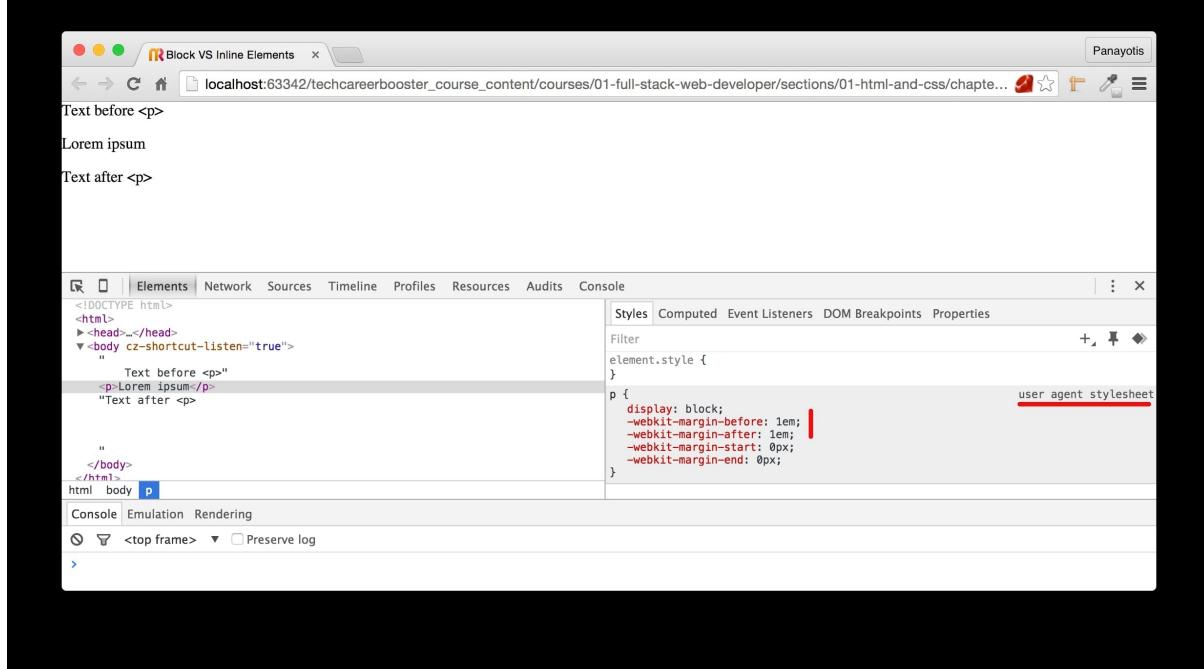
(the above code snippet online)

Save the above CSS rule inside `stylesheets/main.css` and link this file to your HTML page. If you reload, you will see this result here:



Body with zero margins

Do you also see that `<p>` element has by default top and bottom margins? This is the browser default style and its part of the user agent stylesheet rules. You can see that if you click the Styles tab on the developer tools.



Styles tab user agent stylesheet

Note: Don't worry about the fact that margins are being specified using style properties prefixed with `-webkit`. This is how Chrome (a Webkit browser) names some of its default and other properties inside its user agent stylesheet. Same does Safari, which

is another Webkit engine browser. Firefox names those `margin-block-start` (for `margin-top`) and `margin-block-end` (for `margin-bottom`).

As you can see from the user agent stylesheet, the property that makes an HTML element behave like a block element is the style property with name `display`.

Inline HTML Elements

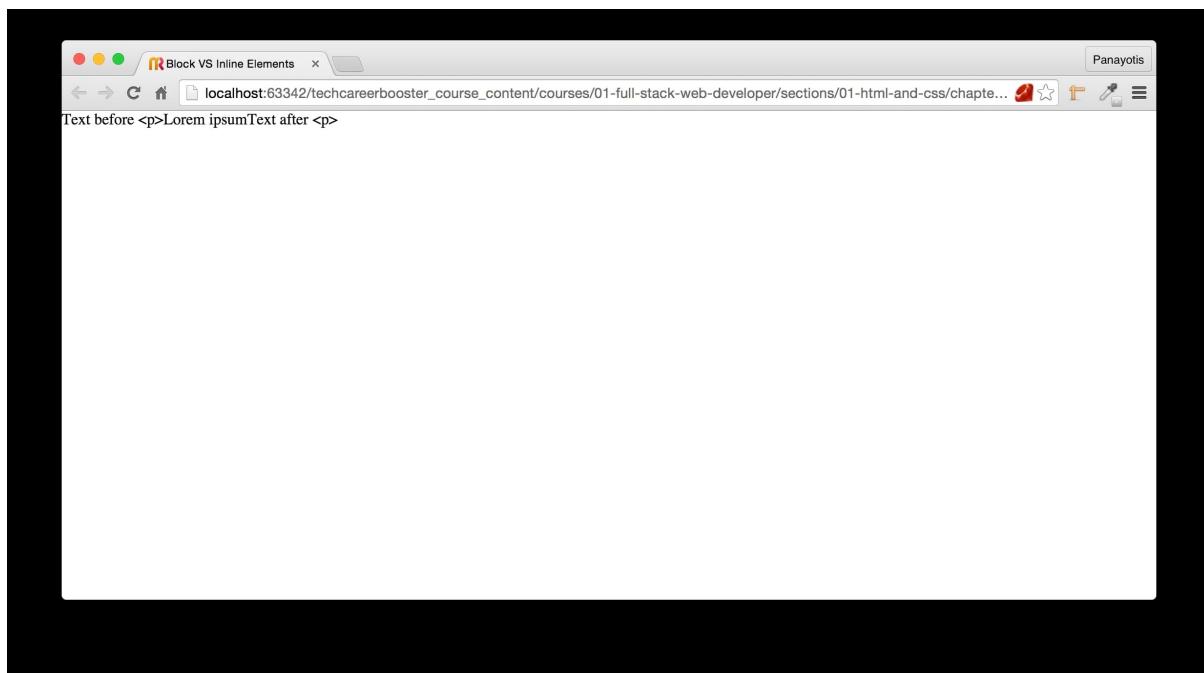
`display` property can take various values, but one other important value, relevant to our discussion, is the value `inline`.

Change the CSS file for the page that we are working on now, to have the following content:

```
1 body {  
2     margin: 0 0;  
3 }  
4  
5 p {  
6     display: inline;  
7 }
```

(the above code snippet online)

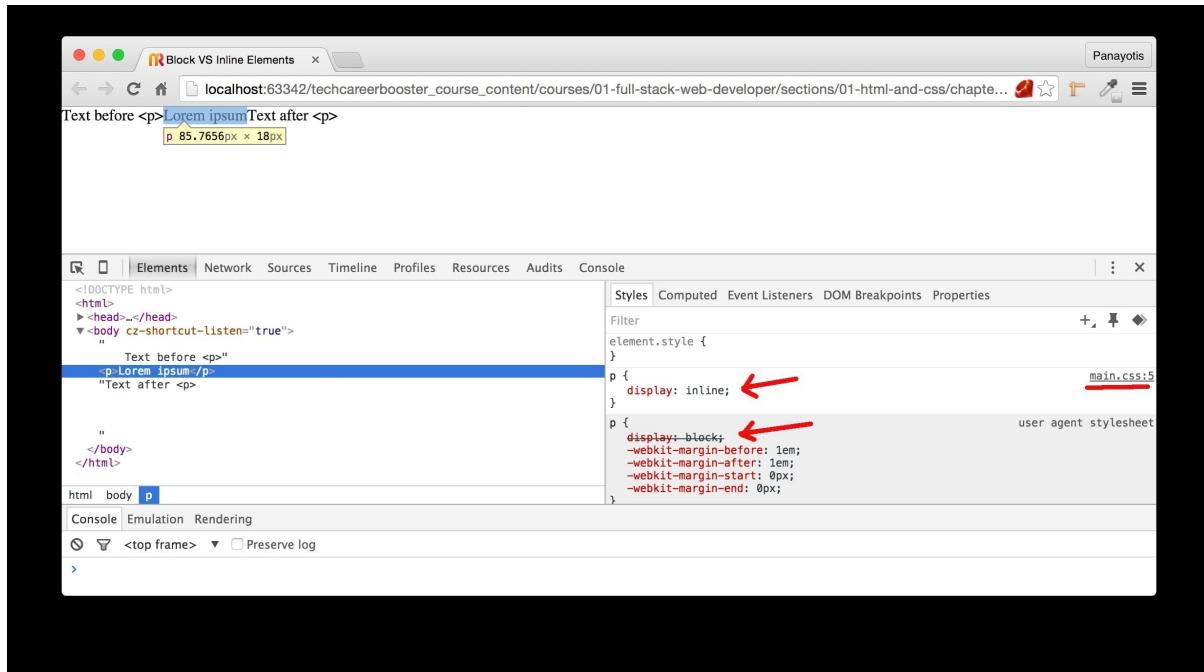
If you reload the page you will see the following result:



P display property set to inline

Aha! The `<p>` element does not have a new line before and a new line after. And this is what an `inline` element is. It has the `display` property set to `inline`, as you can see in the following

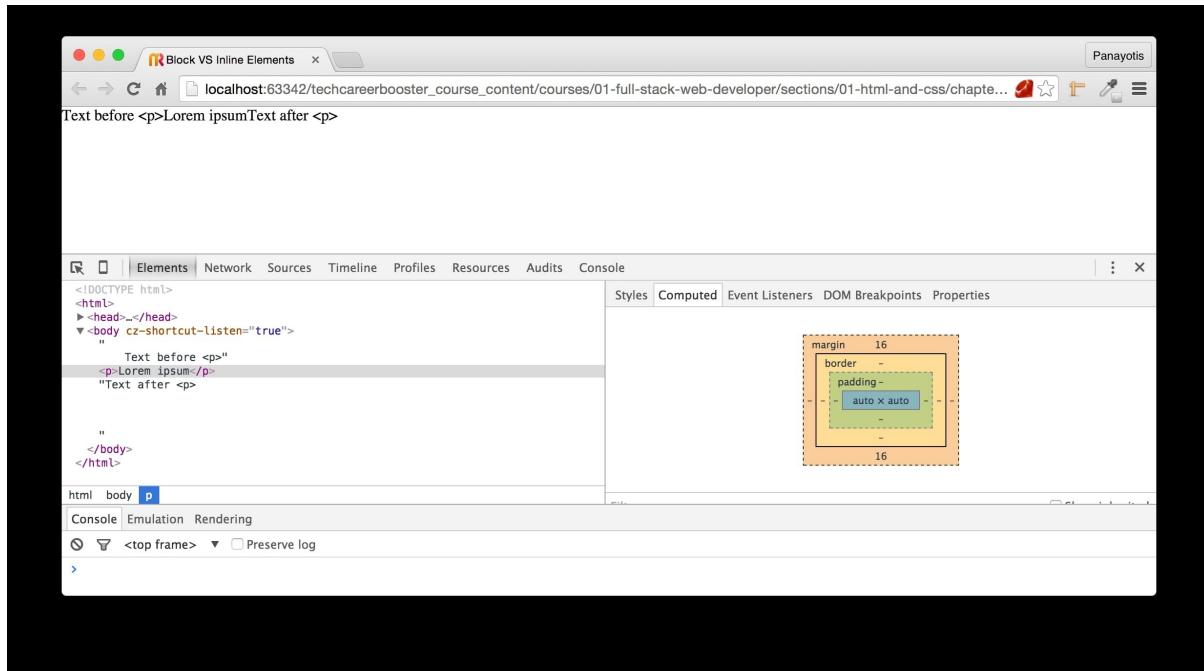
picture, and the element appears next to the previously printed content, to its expected position as the flow of the document is drawn from top to bottom.



P with display property inline styles

See how the user agent style `display: block;` is displayed as deleted by the developer tools. This is because it has been overridden by the value in our `main.css` file. In fact, developer tools, displays the exact line number where the overriding rule exists inside your `main.css` file. If you click on `main.css` it will take you there and you will be able to see the actual rule and CSS file inside the developer tools tab.

See also the new box model for this particular p:



`p` display inline box model

Although you can see that there is top and bottom margin, in fact, these are not used.

Important: When an HTML element is `inline`, then, there are some style properties that are automatically calculated and you cannot set them to specific values: `* width`, `* height`, `* margin-top`, `* margin-bottom`

Default Block Elements

Now that you understand the difference between an `inline` element and a `block` element, and you already know that a `<p>` element is, by default, a `block` element, with `display` property `block`, I can give you the list of the elements that are, by default, `block` elements.

Default HTML Block Elements

| | |
|-------------------------------|--|
| <code><address></code> | <code><article></code> |
| <code><aside></code> | <code><blockquote></code> |
| <code><canvas></code> | <code><dd></code> |
| <code><div></code> | <code><dl></code> |
| <code><fieldset></code> | <code><figcaption></code> |
| <code><figure></code> | <code><footer></code> |
| <code><form></code> | <code><h1>, ..., <h6></code> |
| <code><header></code> | <code><hgroup></code> |
| <code><hr></code> | <code></code> |
| <code><main></code> | <code><nav></code> |
| <code><noscript></code> | <code></code> |
| <code><output></code> | <code><p></code> |
| <code><pre></code> | <code><section></code> |

Default HTML Block Elements

| | |
|-----------------|--------------------|
| <table> | <tfoot> <video> |
|-----------------|--------------------|

During this course, you will learn how to use many of these elements, so, don't worry if most of them do not ring any bell to you.

Default Inline Elements

The list of default inline elements is this:

Default HTML Inline Elements

| | |
|--|---|
| <a> <acronym> <bdo> <cite> <dfn> <i> <input> <label> <object> <samp> <select> <sub> <textarea> <tt> | <abbr> <big> <button> <code> <kbd> <map> <q> <script> <small> <sup> <time> <var> |
|--|---|

Setting width and height for inline elements

We have already seen that if an element is an inline element, then its width and height cannot be specified. Take, for example, the following page:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Inline Element</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>

```

```
10    <span id="my-span">This is the content of a span</span>
11  </body>
12
13
14 </html>
```

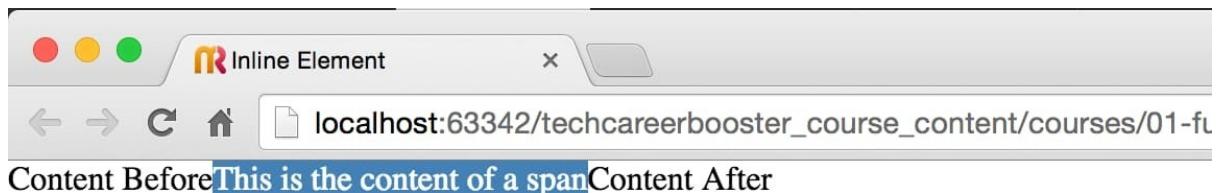
(the above code snippet online)

with the following CSS rules:

```
1  * {
2    box-sizing: border-box;
3    margin: 0 0;
4  }
5
6 #my-span {
7   background-color: #4682B4;
8   color: White;
9   width: 300px;
10  height: 300px;
11 }
```

(the above code snippet online)

If you load this page on your browser, you will see this:



Inline Element with Width And Height

What if you really want to give some width and height to this inline element? Our suggestion here is that you may need to reconsider your tag and instead of using an inline tag, try to use a block tag. Does that solve your problem? If not, and you still want your content to be positioned

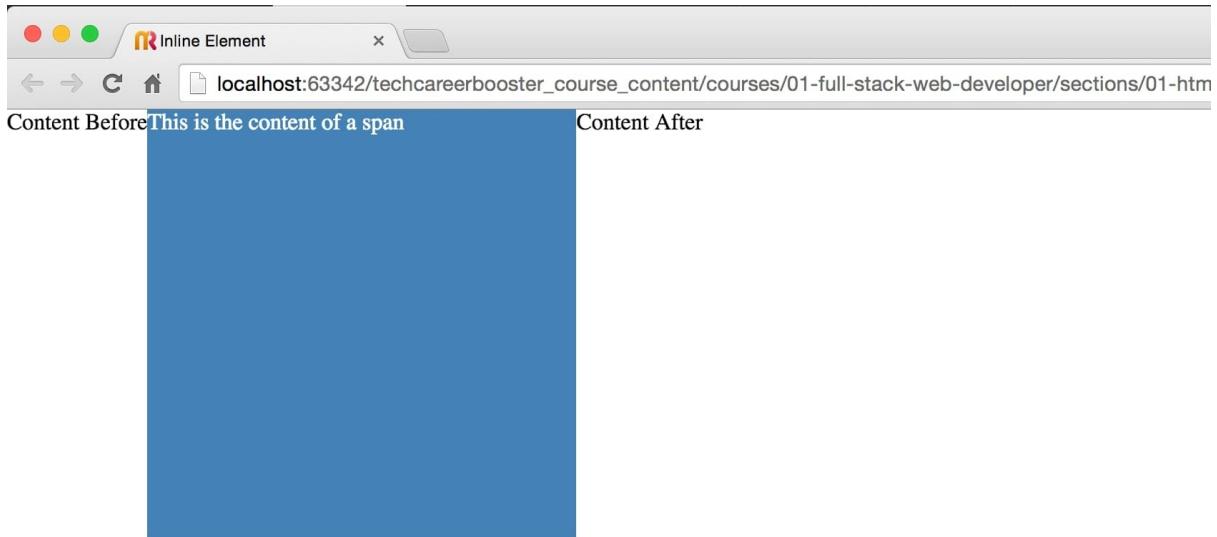
without new lines before and after the element, then you have the option of `inline-block` value for the `display` property.

Let's change the CSS rule file and add a `display` property with value `inline-block` for your `span` element:

```
1 * {  
2     box-sizing: border-box;  
3     margin: 0 0;  
4 }  
5  
6 #my-span {  
7     background-color: #4682B4;  
8     color: White;  
9     width: 300px;  
10    height: 300px;  
11    display: inline-block;  
12 }
```

(the above code snippet online)

If you reload the page, you will see something like this:

A screenshot of the Chrome DevTools Elements tab. The DOM tree shows the following structure:

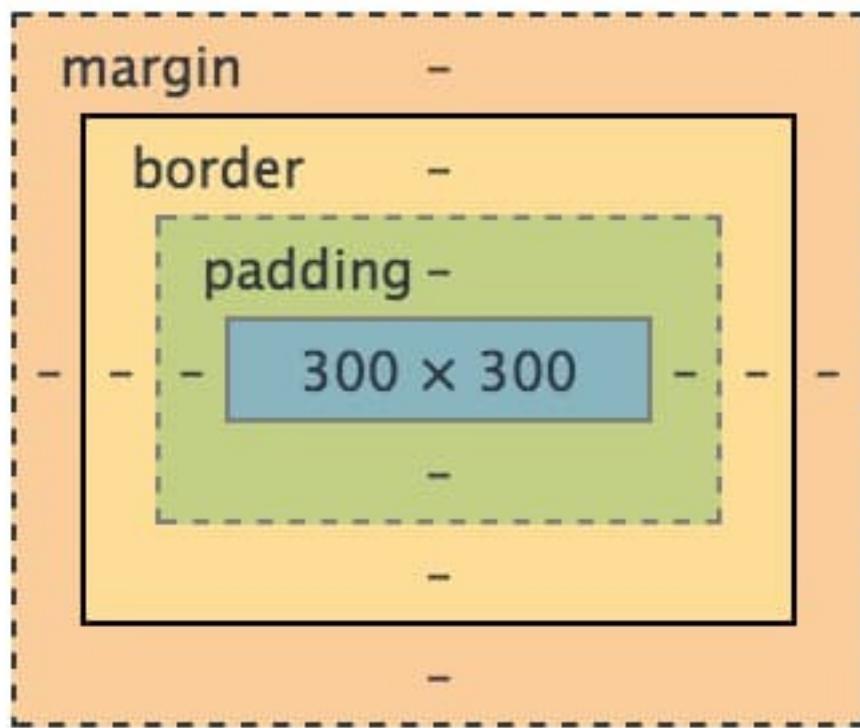
```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body cz-shortcut-listen="true">
    "Content Before"
    <span id="my-span">This is the content of a span</span>
    "Content After"
```

The span element is selected. The right panel shows the "Styles" tab with the following CSS:

```
element.style { }
#my-span {
  background-color: #4682B4;
  color: White;
  width: 300px;
  height: 300px;
  display: inline-block;
```

Display inline block

and if you also see the CSS box model, it will be:



CSS Box Model for inline block element

As you can see here, the initially `inline` element is now displayed with the actual `width` and `height` that we have specified in the CSS file.

Closing Note

That was the first encounter with `block` and `inline` elements. Later on, you will learn more advanced techniques that have to do with how your elements are positioned on your page.

Tasks

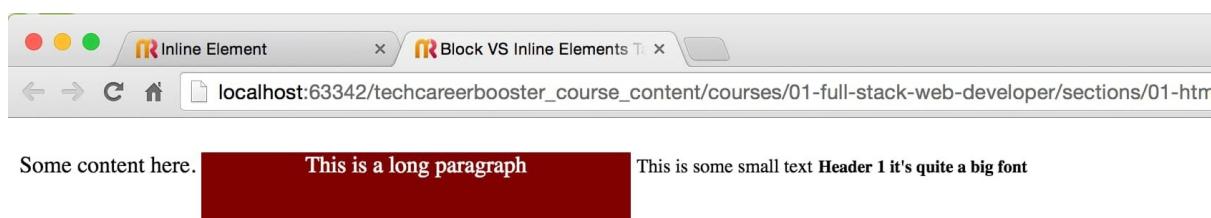
Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Answer the following questions and send them to your mentor:
 1. What is the `display` property of `body`?

2. What is the display property of a `<p>`?
 3. What is the display property of an `<h1>`?
 4. What is the display property of a ``?
 5. What is the display property of a ``?
 6. What is the display property of an ``?
 7. What is the display property of a `<table>`?
 8. Can we specify the `width` property of a `block` element?
 9. Can we specify the `width` property of an `inline` element?
 10. Can we specify the `height` property of a `block` element?
 11. Can we specify the `height` property of an `inline` element?
 12. What display property value should I use in order to be able to specify the `width` and `height` of a ``?
2. Create the following page:



Block vs Inline Elements - Task

In order to do that, use the following HTML markup:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Block VS Inline Elements Task</title>
6          <link rel="stylesheet" href="stylesheets/task.css" type="text/css">
7      </head>
8
9      <body>
10
11          <span>Some content here.</span>
12          <p>
13              This is a long paragraph
14          </p>
15          <small>This is some small text</small>
16
17          <h1>Header 1 it's quite a big font</h1>
```

```
18  
19      </body>  
20  
21  </html>
```

(the above code snippet online)

We are giving you here some CSS rules that you will need (but not all that you will need):

1. All elements should have `box-sizing` equal to `border-box`.
2. Width of paragraph should be `300px`
3. Height of paragraph should be `50px`
4. `text-align` can be used to center the text.
5. `<h1>` `font-size` should be `12px`.

Chapter 18 - Dividing Page To Sections Using Divs

Summary

In this chapter, we will learn how to divide our canvas area to different sections.

You will finally build a web page like this:



Dividing Page Into Sections

Learning Goals

1. Learn how to use the `<div>` element.
2. Learn that `<div>` element is a block element.
3. Learn how you can limit the width of a `<div>` element.
4. Learn how you can use the `display` property with value `inline-block` in order to position columns, one next to each other.

5. Learn about the `float` style property.
6. Learn about the `clear` style property.

Dividing Page to Sections Using `<div>`

We now proceed to one of the most important parts of HTML authoring. It is a way to divide our HTML page into sections.

For example, how do we create a 3 column layout with header and footer? Like this?



3 Columns Layout with Header and Footer

The main tool that will allow us to divide our page into sections is `<div>`.

Important: Do not use HTML tables to divide your page to sections. Use tables only if you want to display tabular data inside a section of your page.

Let's start. Our target is to build the HTML page above. We will take it step by step.

Step 1: Empty Page

Let's start with an empty HTML page that also references the file that will hold our CSS rules. Note that `stylesheets/content.css` is empty as well.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>3 Columns Layout with Header and Footer</title>
6          <link rel="stylesheet" href="stylesheets/content.css">
7      </head>
8
9      <body>
10
11
12  </body>
13
14 </html>
```

(the above code snippet online)

Step 2: Let's set the basic rules

We will start with resetting all the elements to border-box. That will make our sizing calculations easier:

```
1  * {
2      box-sizing: border-box;
3 }
```

(the above code snippet online)

We still have an empty page of course, and if we reload, there is nothing more to see.

Step 3: Dividing our page

Before we insert the actual content of our page, we need to understand how we want the page to be divided into sections. Looking at the final result, we identify 5 sections:

1. Header
2. Left Column
3. Middle Column

4. Right Column
5. Footer

This is pretty much standard layout. And the standard HTML element that is used to divide pages into sections is the `<div>` element. Hence, you probably need to have 5 div elements, one for each section.

Let's put the first section in:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Three Column Layout with Header and Footer</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10         <div id="header"></div>
11
12     </body>
13
14
15 </html>

```

(the above code snippet online)

If you save and reload your page, you will still see nothing, because the div that we inserted does not have any content. Let's add some. The final result can be achieved with an `<h3>` and a `<p>` inside the header div. Here it is:

```

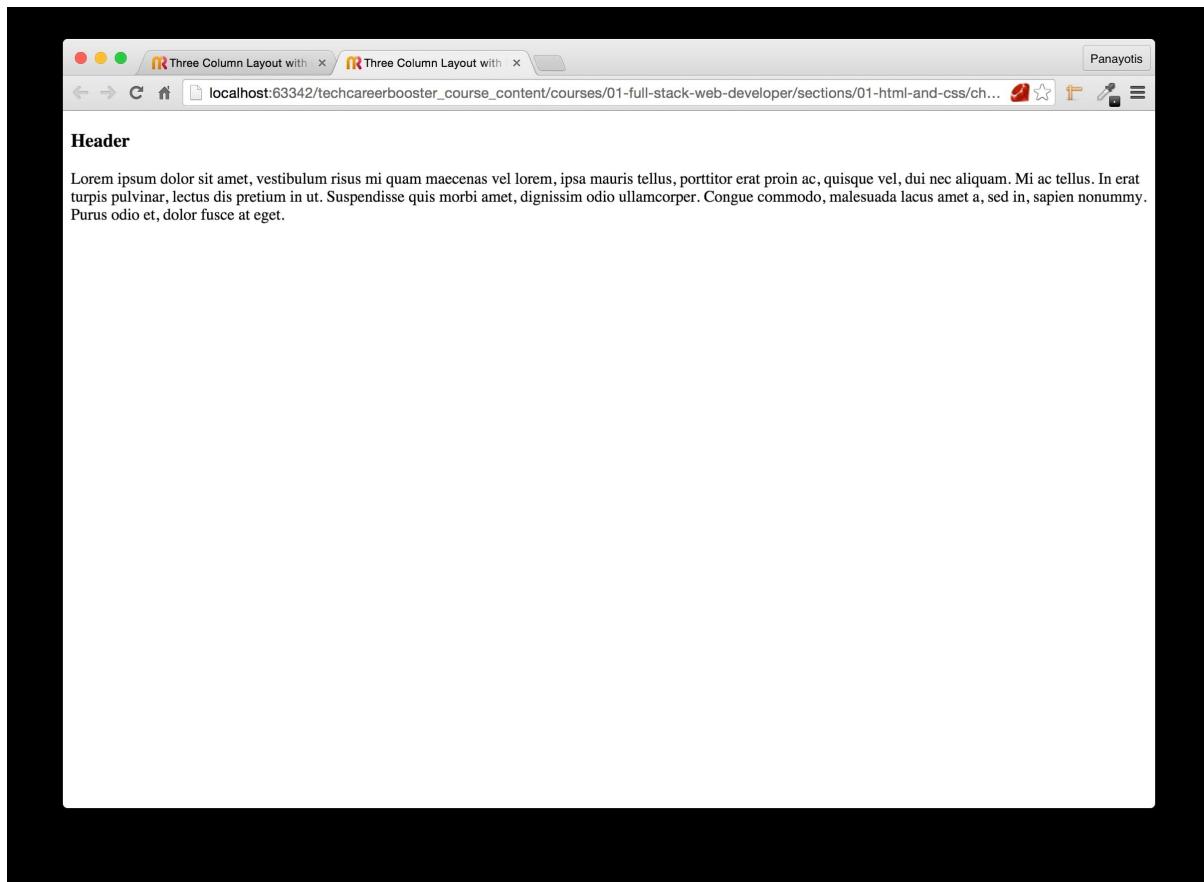
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Three Column Layout with Header and Footer</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10         <div id="header">
11             <h3>Header</h3>
12             <p>
13                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
14 ipsa mauris tellus,
15 porttitor erat proin ac, quisque vel, dui nec aliquam. Mi ac tellus. In e\

```

```
17 rat turpis pulvinar, lectus dis pretium in ut.  
18 Suspendisse quis morbi amet, dignissim odio ullamcorper. Congue commodo, \  
19 malesuada lacus amet a, sed in, sapien nonummy.  
20 Purus odio et, dolor fusce at eget.  
21 </p>  
22 </div>  
23  
24 </body>  
25  
26 </html>
```

(the above code snippet online)

If you reload the page, you will see this:



Dividing Page With Divs Header Without Style

As you already know, the `<div>` element is a block element and occupies, by default, the whole width of the browser. Content before or after that, will be on different lines.

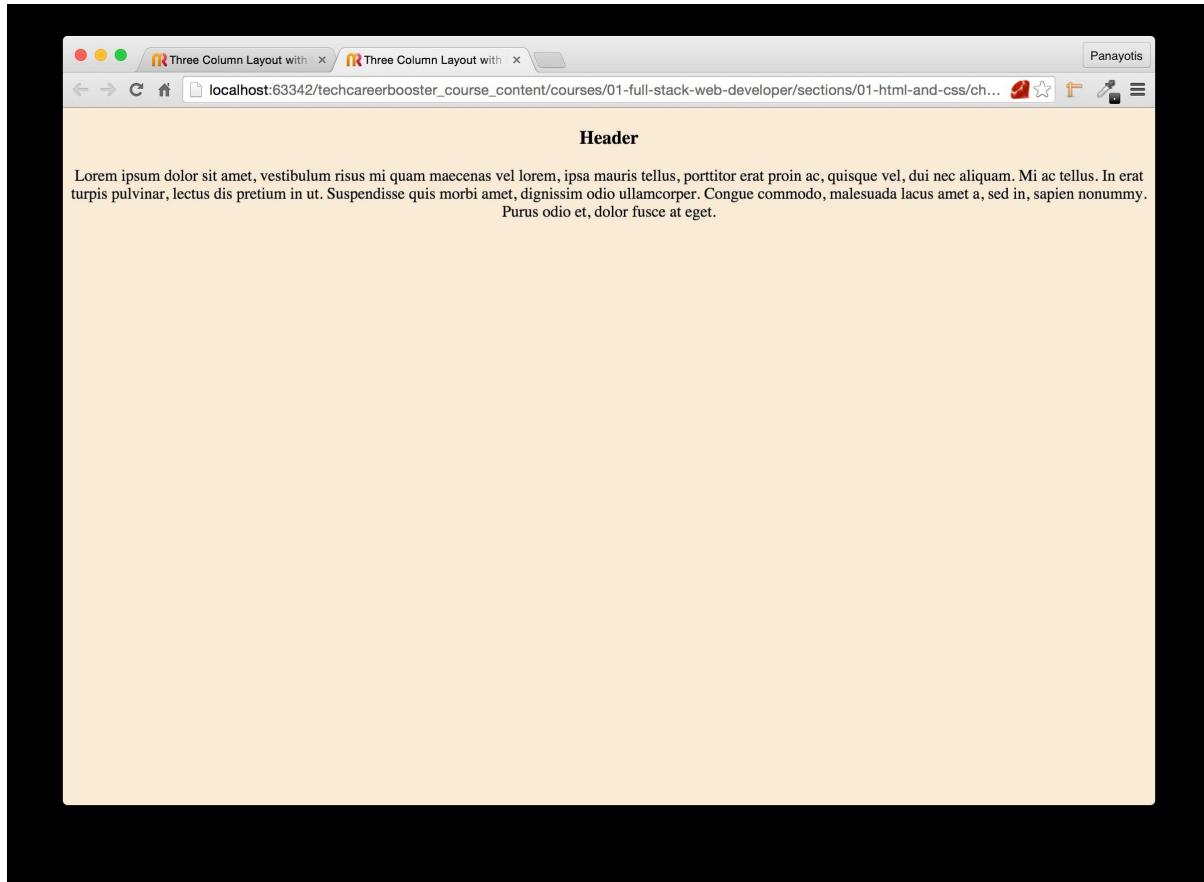
Hence, we are pretty much ready with that. But, let's style it a little bit more in order to bring it where we want. As we inspect the final result, we need to specify the text alignment to be centered and the background color to be of a particular color. Actually, text alignment center needs to be for the whole document. Same goes for the background color. If you look at the final result carefully, you will see that the background color of the header is actually the background

of the whole document. Let's do that:

```
1 * {
2   box-sizing: border-box;
3 }
4
5 body {
6   background-color: antiquewhite;
7   text-align: center;
8 }
```

(the above code snippet online)

If you apply these rules and you reload the page, you will see the following:



Header with Styling

If you look at the page that we want to arrive to, header seems to be ok. There are some details, but we can consider that done, for now and proceed to the rest of the sections of our page.

Step 4 - Left Column

We will now put on our page the left column. We need another `<div>` element with the necessary content inside. We will also apply a background style in order to make it stand out. Let's update our HTML page with the new `<div>` and then our CSS rules file:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Three Column Layout with Header and Footer</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="header">
12             <h3>Header</h3>
13             <p>
14                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
15                 ipsa mauris tellus,
16                 porttitor erat proin ac, quisque vel, dui nec aliquam. Mi ac tellus. In e\
17                 rat turpis pulvinar, lectus dis pretium in ut.
18                 Suspendisse quis morbi amet, dignissim odio ullamcorper. Congue commodo, \
19                 malesuada lacus amet a, sed in, sapien nonummy.
20                 Purus odio et, dolor fusce at eget.
21             </p>
22         </div>
23
24         <div id="left-column">
25             <h3>Left Column</h3>
26
27             <p>
28                 Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\
29                 is nunc, porta eu leo in ac bibendum. Justo egestas
30                 porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
31                 eos vel porttitor amet erat nunc, sed penatibus turpis.
32                 Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
33                 uada, taciti vitae malesuada ac porttitor eget cum,
34                 magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
35                 mollis dis. Blandit sollicitudin luctus fringilla placerat
36                 quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
37                 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
38                 placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
39                 ie erat mattis nullam nostra. Dignissim orci suspendisse in
40                 commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \
41                 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu
42                 risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\
43                 praesent laoreet, lobortis torquent magna habitasse ut
44                 pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\
45                 ris molestie maecenas cum arcu hendrerit, in per nunc,
46                 ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \
```

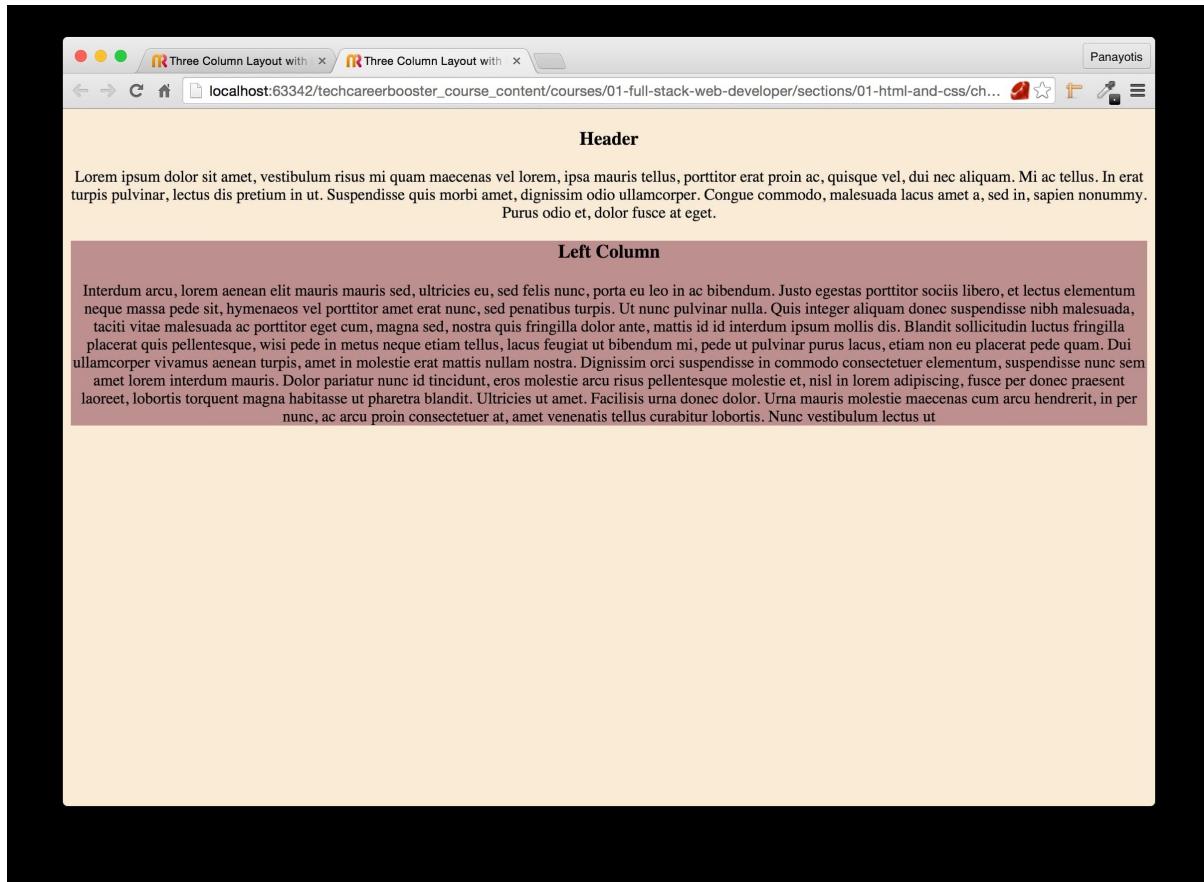
```
47 Nunc vestibulum lectus ut
48     </p>
49     </div>
50     </body>
51
52 </html>
```

(the above code snippet online)

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     background-color: antiquewhite;
7     text-align: center;
8 }
9
10 #left-column {
11     background-color: #BC8F8F;
12 }
```

(the above code snippet online)

As you see from the HTML code above, we have put our new `<div>` exactly after the previous one, as a sibling element. What is the visual result? Let's reload the page and see:



Header and Left Column

One `<div>` is displayed below the other. And this is expected. As we said, the `<div>` element is a **block** style element. Hence has a line change before and a line change after and occupies the whole available width.

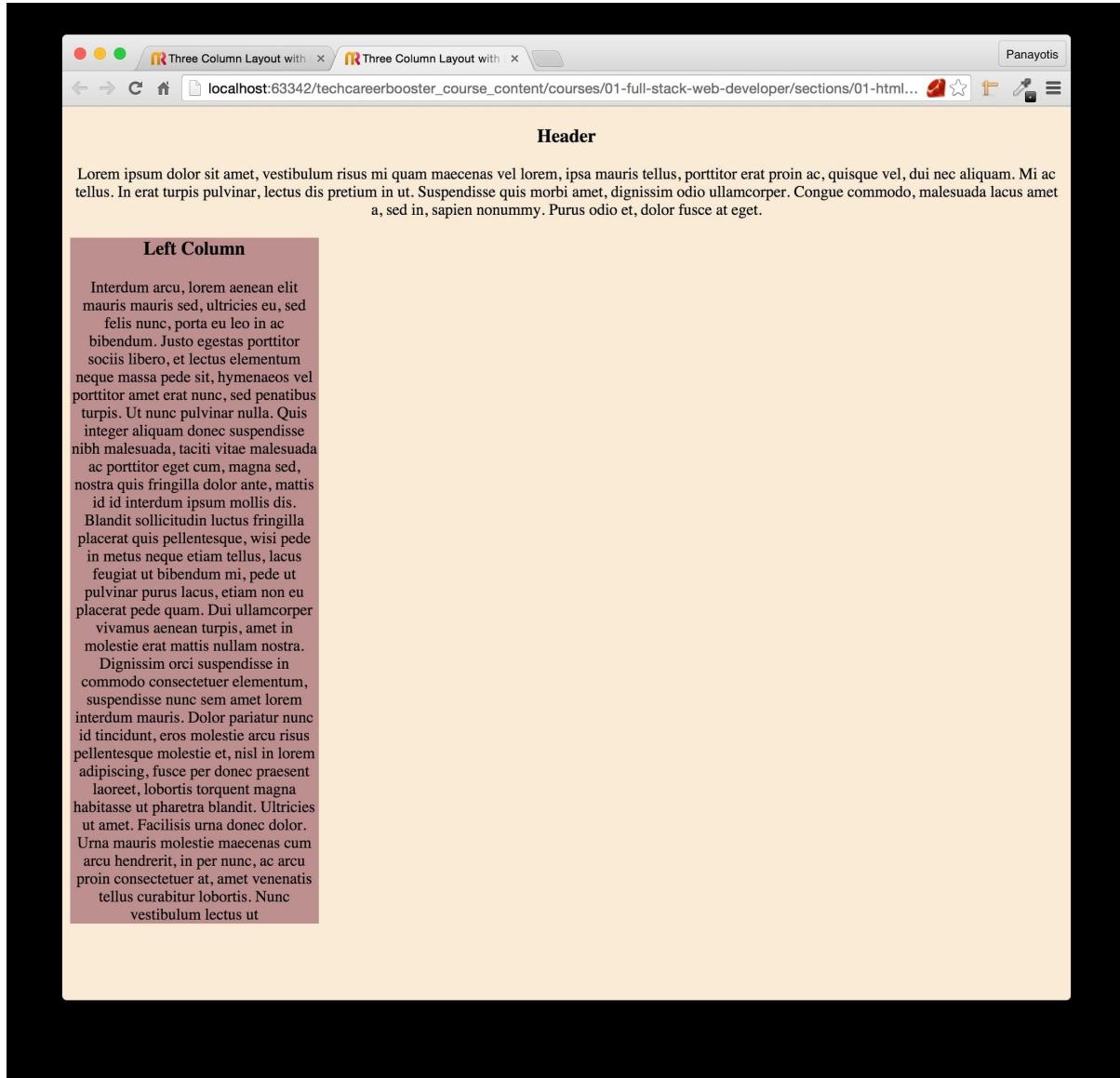
Let's see now how we can limit the width of this new div to be a quarter of the whole page width. Our intention is to have the left and right columns be 25% of the page width, and leave the rest of the page width (50%) for the middle column. Shall we just add the required width to the left column?

```

1 * {
2   box-sizing: border-box;
3 }
4
5 body {
6   background-color: antiquewhite;
7   text-align: center;
8 }
9
10 #left-column {
11   background-color: #BC8F8F;
12   width: 25%;
13 }
```

(the above code snippet online)

Let's reload our page after that. What we will see is this:



Left column with 25% width

That's much better.

Step 5 - Middle Column

Let's proceed to the middle column. This needs to be another `<div>`. Let's put it inside our HTML page and give it another background color in order to make it stand out.

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     background-color: antiquewhite;
7     text-align: center;
8 }
9
10 #left-column {
11     background-color: #BC8F8F;
12     width: 25%;
13 }
14
15 #middle-column {
16     background-color: #4682B4;
17 }
```

(the above code snippet online)

and the HTML now becomes:

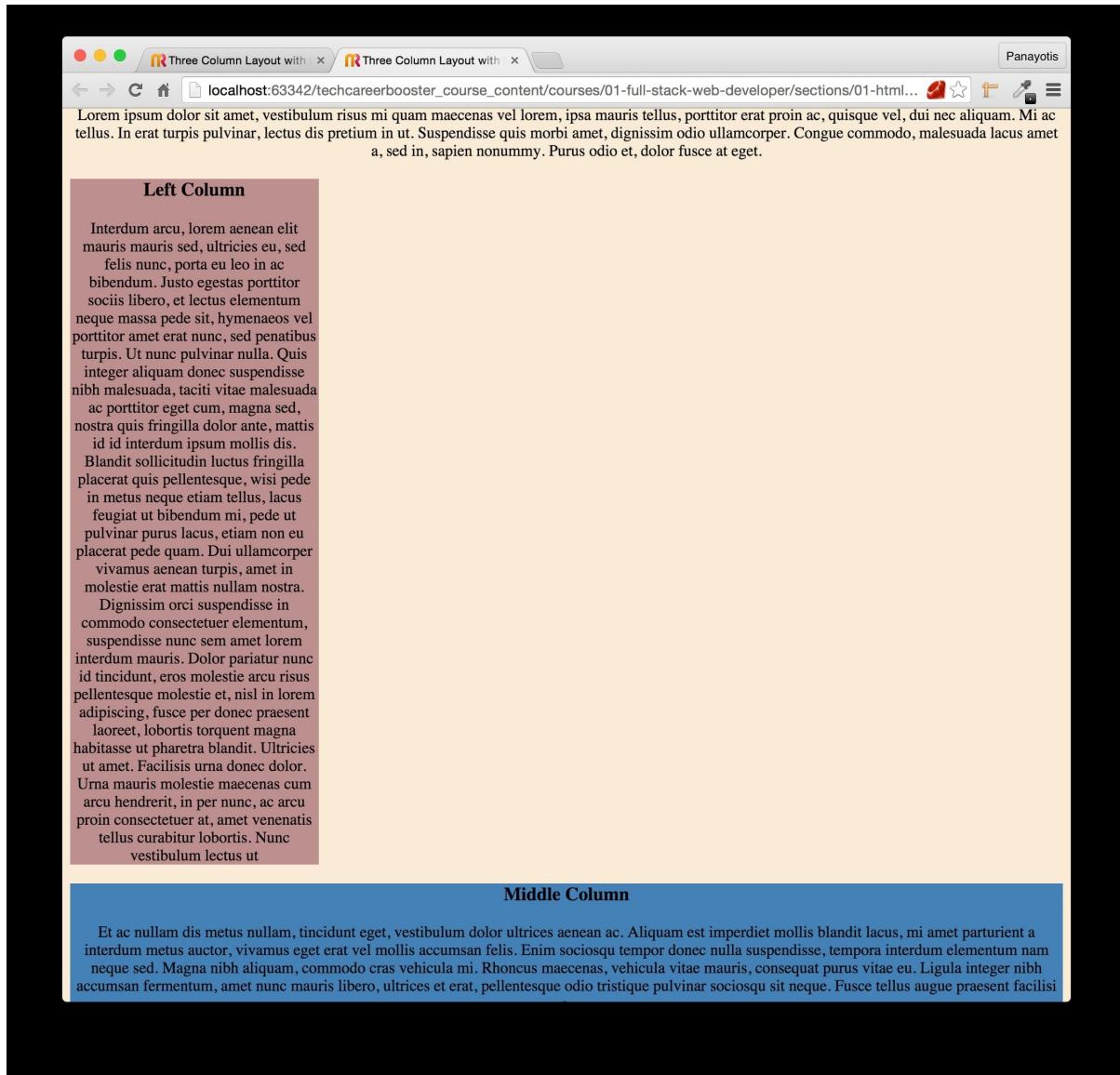
```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8">
5         <title>Three Column Layout with Header and Footer</title>
6         <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7     </head>
8
9     <body>
10
11         <div id="header">
12             <h3>Header</h3>
13             <p>
14                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
15 ipsa mauris tellus,
16                 porttitor erat proin ac, quisque vel, dui nec aliquam. Mi ac tellus. In e\
17 rat turpis pulvinar, lectus dis pretium in ut.
18                 Suspendisse quis morbi amet, dignissim odio ullamcorper. Congue commodo, \
19 malesuada lacus amet a, sed in, sapien nonummy.
20                 Purus odio et, dolor fusce at eget.
21             </p>
22         </div>
23
24         <div id="left-column">
25             <h3>Left Column</h3>
```

```
26
27     <p>
28         Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\
29 is nunc, porta eu leo in ac bibendum. Justo egestas
30         porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
31 eos vel porttitor amet erat nunc, sed penatibus turpis.
32         Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
33 uada, taciti vitae malesuada ac porttitor eget cum,
34         magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
35 mollis dis. Blandit sollicitudin luctus fringilla placerat
36         quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
37 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
38         placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
39 ie erat mattis nullam nostra. Dignissim orci suspendisse in
40         commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \
41 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu
42         risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\
43 praesent laoreet, lobortis torquent magna habitasse ut
44         pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\
45 ris molestie maecenas cum arcu hendrerit, in per nunc,
46         ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \
47 Nunc vestibulum lectus ut
48     </p>
49 </div>
50
51 <div id="middle-column">
52     <h3>Middle Column</h3>
53
54     <p>
55         Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
56 aenean ac. Aliquam est imperdiet mollis blandit lacus,
57         mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
58 accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
59         tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
60 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
61         purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
62 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
63         sit neque. Fusce tellus augue praesent facilisi a.<br/>
64         Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
65 aenean ac. Aliquam est imperdiet mollis blandit lacus,
66         mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
67 accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
68         tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
69 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
70         purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
71 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
```

```
72      sit neque. Fusce tellus augue praesent facilisi a.<br/>
73      Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
74 aenean ac. Aliquam est imperdiet mollis blandit lacinia,
75      mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
76 accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
77      tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
78 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
79      purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
80 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
81      sit neque. Fusce tellus augue praesent facilisi a.<br/>
82      Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
83 aenean ac. Aliquam est imperdiet mollis blandit lacinia,
84      mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
85 accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
86      tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
87 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
88      purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
89 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
90      sit neque. Fusce tellus augue praesent facilisi a.<br/>
91      </p>
92      </div>
93
94  </body>
95
96 </html>
```

(the above code snippet online)

If you load this page on your browser, the result will be the following:



Header, Left and Middle Column Below

The result is the expected one. Isn't it? The fact that we defined the `width` of the left column `div` does not stop it from being a block style element. But even if it were an inline-block element, then, still, the middle column `div`, is a block element, which means it stays on its own line.

So, what is the solution to bring both `div`s (left and middle column) on the same line and still allow them to have limited widths? The `inline-block` value of the `display` property, for both of them. Let's do it.

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     background-color: antiquewhite;
7     text-align: center;
8 }
9
10 #left-column {
11     background-color: #BC8F8F;
12     width: 25%;
13     display: inline-block;
14 }
15
16 #middle-column {
17     background-color: #4682B4;
18     width: 50%;
19     display: inline-block;
20 }
```

(the above code snippet online)

As you can see above, we have set `display` property to the value `inline-block` for both divs. We have also specified the width of the middle column to be 50%. Is that getting better? Let's reload the page and see:



Header, Left and Middle Column On Same Line

It does getting better.

Step 6 - Right Column

Ok. I guess that now, we know how we can add the right column. It is another div. Let's put it inside our HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Three Column Layout with Header and Footer</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="header">
12             <h3>Header</h3>
13             <p>
14                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
15                 ipsa mauris tellus,
16                 porttitor erat proin ac, quisque vel, dui nec aliquam. Mi ac tellus. In e\
17                 rat turpis pulvinar, lectus dis pretium in ut.
18                 Suspendisse quis morbi amet, dignissim odio ullamcorper. Congue commodo, \
19                 malesuada lacus amet a, sed in, sapien nonummy.
20                 Purus odio et, dolor fusce at eget.
21             </p>
22         </div>
23
24         <div id="left-column">
25             <h3>Left Column</h3>
26
27             <p>
28                 Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\
29                 is nunc, porta eu leo in ac bibendum. Justo egestas
30                 porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
31                 eos vel porttitor amet erat nunc, sed penatibus turpis.
32                 Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
33                 uada, taciti vitae malesuada ac porttitor eget cum,
34                 magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
35                 mollis dis. Blandit sollicitudin luctus fringilla placerat
36                 quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
37                 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
38                 placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
39                 ie erat mattis nullam nostra. Dignissim orci suspendisse in
40                 commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \
41                 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu
42                 risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\
43                 praesent laoreet, lobortis torquent magna habitasse ut
44                 pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\
45                 ris molestie maecenas cum arcu hendrerit, in per nunc,
46                 ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \
```

```
47 Nunc vestibulum lectus ut
48     </p>
49     </div>
50
51     <div id="middle-column">
52         <h3>Middle Column</h3>
53
54         <p>
55             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
56             aenean ac. Aliquam est imperdiet mollis blandit lacus,
57                 mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
58             accumsan felis. Nam tempore donec nulla suspendisse,
59                 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
60             s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
61                 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
62             libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
63                 sit neque. Fusce tellus augue praesent facilisi a.<br/>
64             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
65             aenean ac. Aliquam est imperdiet mollis blandit lacus,
66                 mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
67             accumsan felis. Nam tempore donec nulla suspendisse,
68                 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
69             s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
70                 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
71             libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
72                 sit neque. Fusce tellus augue praesent facilisi a.<br/>
73             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
74             aenean ac. Aliquam est imperdiet mollis blandit lacus,
75                 mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
76             accumsan felis. Nam tempore donec nulla suspendisse,
77                 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
78             s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
79                 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
80             libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
81                 sit neque. Fusce tellus augue praesent facilisi a.<br/>
82             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
83             aenean ac. Aliquam est imperdiet mollis blandit lacus,
84                 mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
85             accumsan felis. Nam tempore donec nulla suspendisse,
86                 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
87             s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
88                 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
89             libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
90                 sit neque. Fusce tellus augue praesent facilisi a.<br/>
91             </p>
92         </div>
```

```

93
94     <div id="right-column">
95         <h3>Right Column</h3>
96         <p>
97             Diam elementum eleifend. Morbi nec lorem, mi ipsum bibendum aliquam neque\
98             , turpis donec pellentesque porta tempus mauris, justo a odio,
99                 ut cras. A dapibus, donec ipsum morbi ipsum ultrices lacus risus, ut vive\
100            rra, repellendus lorem etiam quis vitae, semper sit fusce pretium
101                suspendisse. In urna nunc proin egestas placerat mauris, lorem commodo im\
102            perdi et nascetur, nisl massa a ullamcorper laoreet,
103                nullam nunc arcu turpis vitae. Augue dui, aliquam porttitor lorem etiam. \
104            Felis mauris mollis sed justo sed feugiat, vivamus ullamcorper
105                ligula dui id pellentesque, morbi ligula dui consequat et bibendum, solli\
106            citudin habitasse commodo varius eaque urna, facilisis enim amet
107                volutpat. In pretium ut porttitor in vitae, amet pellentesque pede quis p\
108            haretra justo esse, parturient amet eros, donec quis etiam,
109                nulla justo ipsum dui vivamus phasellus. Eget enim cursus lectus nec soll\
110            icitudin, amet eget pede, eget nulla. Nunc amet eros rhoncus
111                lobortis, leo egestas non nullam suspendisse, non hendrerit orci. Justo v\
112            estibulum donec, vivamus elit enim. Vitae iaculis ipsum sapien nec,
113                aliquam odio felis. Donec dictum quisque ultrices risus varius pede.
114            </p>
115        </div>
116
117    </body>
118
119 </html>

```

(the above code snippet online)

Let's also add a background color and make the div to be inline-block, with width 25%. Otherwise, it will appear below the left and middle column. Our CSS becomes:

```

1  * {
2      box-sizing: border-box;
3  }
4
5  body {
6      background-color: antiquewhite;
7      text-align: center;
8  }
9
10 #left-column {
11     background-color: #BC8F8F;
12     width: 25%;
13     display: inline-block;
14 }

```

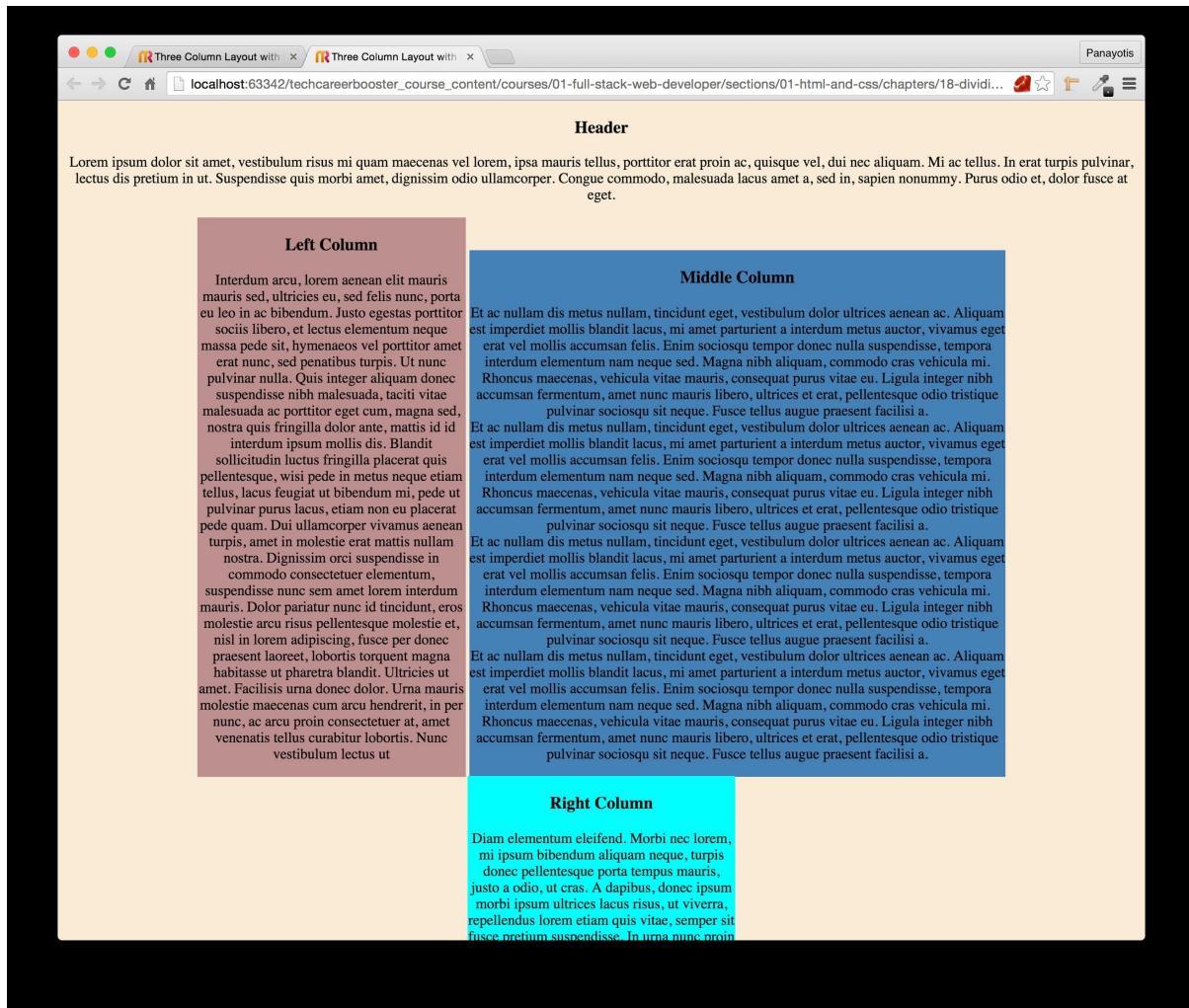
```

15
16 #middle-column {
17   background-color: #4682B4;
18   width: 50%;
19   display: inline-block;
20 }
21
22 #right-column {
23   background-color: aqua;
24   width: 25%;
25   display: inline-block;
26 }

```

(the above code snippet online)

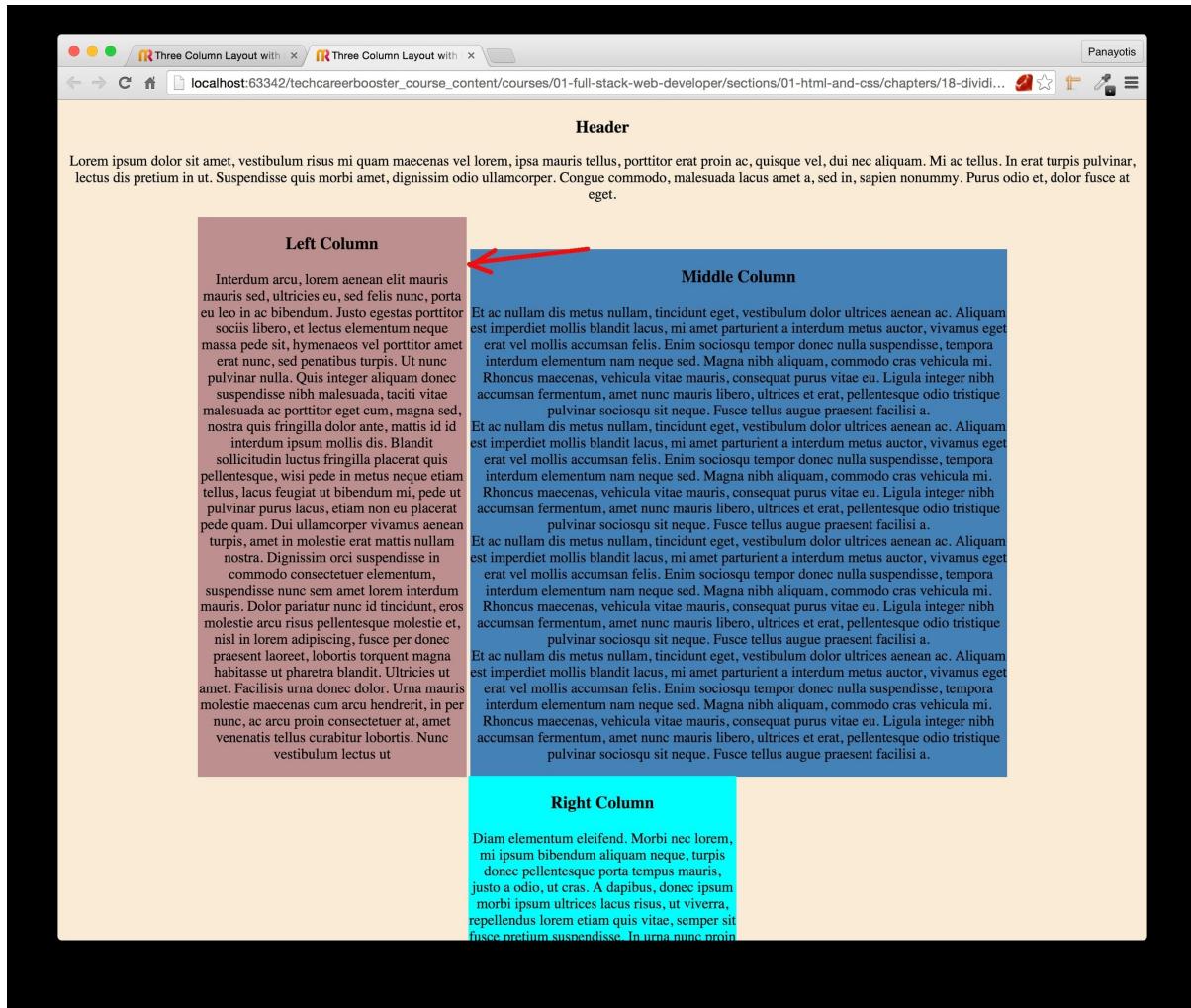
If we reload this page, the result will be the following:



Right Column Below Due to Blanks

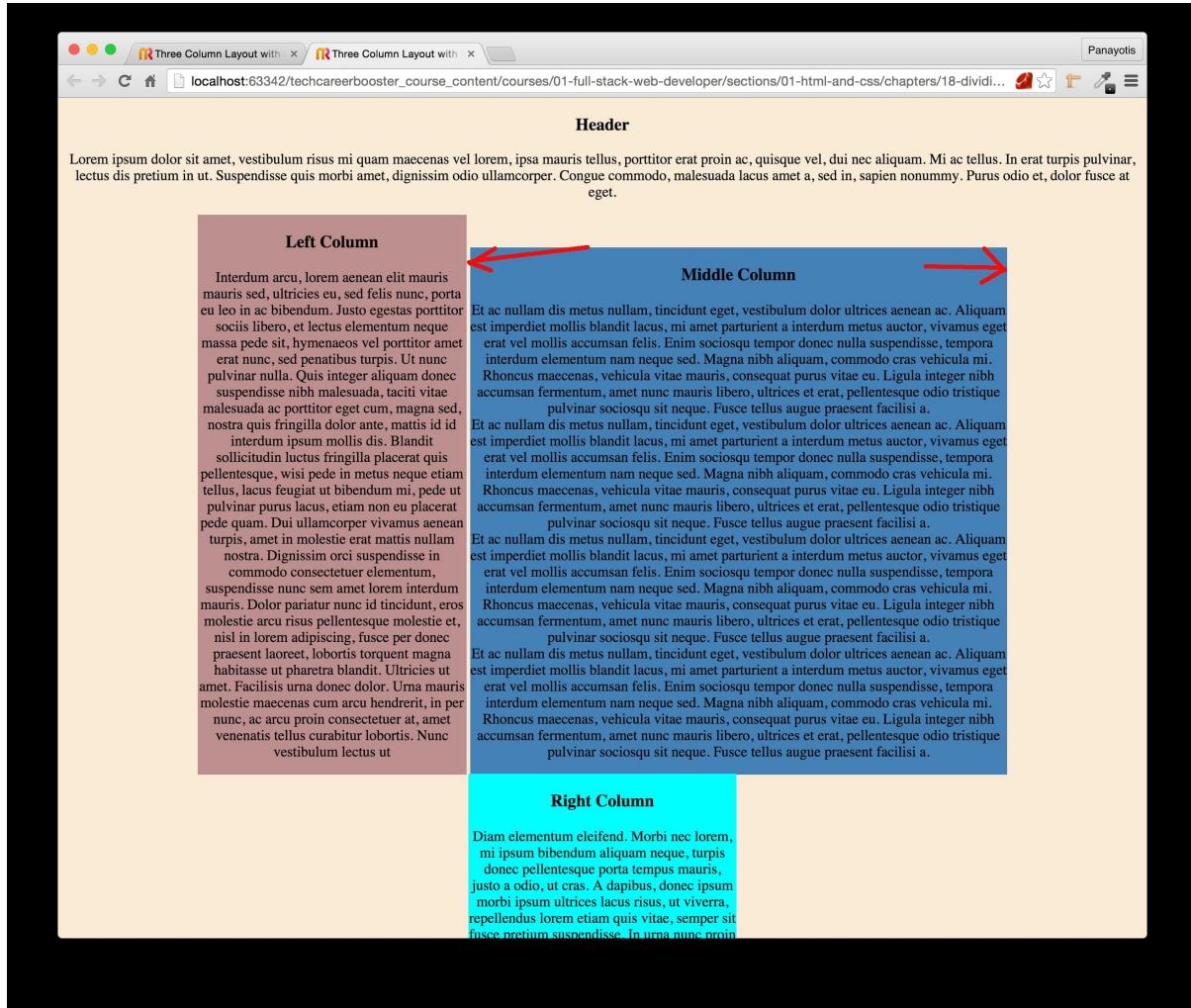
Ooops. This is not what we really expected. Is it? The right column div, although it has the correct style properties, does not have enough room to be positioned to the right of the middle

column and wraps to the next line. Why is that? Did you notice that there is a small blank vertical gap between left column and middle column? Here:



Header, Left and Middle Column Below - Gap Marked

Actually, there is another one too, between middle column and right column. Now positioned right to the middle column and it is not visible (but exists), because the right column has jumped to the next line. Here:



Header, Left and Middle Column Below - Gap Marked next to Middle Column

The small gaps, added to the widths of the 3 divs ($25\% + 50\% + 25\%$) make the whole line not fit in the width of the page and that's why the right column wraps below.

Why does the browser create these gaps? It does it because we have it in our HTML code. The `inline` elements that have spaces around them, are printed with their spaces together. We have turned our divs to `inline-block` elements (which are basically inline elements with the ability to specify width and height) and our HTML code has spaces around them: (you can see the following picture if you right click on your page and you select View Source)

```

 21  <div id="left-column">
 22    <h3>Left Column</h3>
 23
 24  <p>
 25    Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed felis nunc, porta eu leo in ac bibendum. Justo egestas
 26    porttitor sociis libero, et lectus elementum neque massa pede sit, hymenaeos vel porttitor amet erat nunc, sed penatibus turpis.
 27    Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh malesuada, taciti vitae malesuada ac porttitor eget cum,
 28    magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum mollis dis. Blandit sollicitudin luctus fringilla placerat
 29    quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat ut bibendum mi, pede ut pulvinar purus lacus, etiam non eu
 30    placet pede quam. Duis ullamcorper vivamus aenean turpis, amet in molestie erat mattis nunc nostra. Dignissim orci suspendisse in
 31    corporis ultricies et, nisl facilisis urna donec dolor. Urna mauris molestie maecenas cum arcu hendrerit, in per nunc,
 32    ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. Nunc vestibulum lectus ut
 33    pharetra blandit. Ultricies et amet. Facilisis urna donec dolor. Urna mauris molestie maecenas cum arcu hendrerit, in per nunc,
 34    ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. Nunc vestibulum lectus ut
 35  </p>
 36  </div> ←
 37  <div id="middle-column">
 38    <h3>Middle Column</h3>
 39
 40  <p>
 41    Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices aenean ac. Aliquam est imperdiet mollis blandit lacus,
 42    mi amet parturient a interdum metus auctor, vivamus egert erat vel mollis accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
 43    tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cras vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
 44    purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
 45    sit neque. Fusce tellus augue praesent facilisi a.<br/>
 46    Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices aenean ac. Aliquam est imperdiet mollis blandit lacus,
 47    mi amet parturient a interdum metus auctor, vivamus egert erat vel mollis accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
 48    tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cras vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
 49    purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
 50    sit neque. Fusce tellus augue praesent facilisi a.<br/>
 51    Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices aenean ac. Aliquam est imperdiet mollis blandit lacus,
 52    mi amet parturient a interdum metus auctor, vivamus egert erat vel mollis accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
 53    tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cras vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
 54    purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
 55    sit neque. Fusce tellus augue praesent facilisi a.<br/>
 56    Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices aenean ac. Aliquam est imperdiet mollis blandit lacus,
 57    mi amet parturient a interdum metus auctor, vivamus egert erat vel mollis accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
 58    tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cras vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
 59    purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
 60    sit neque. Fusce tellus augue praesent facilisi a.<br/>
 61  </p>
 62  </div> ←
 63  <div id="right-column">
 64    <h3>Right Column</h3>
 65
 66  <p>
 67    Diem elementum eleifend. Morbi nec lorem, mi ipsum bibendum aliquam neque, turpis donec pellentesque porta tempus mauris, justo a odio,
 68    ut cras. A dapibus, donec ipsum morbi ipsum ultrices lacus risus, ut viverra, repellendus lorem etiam quis vite, semper sit fusce pretium
 69    suspendisse. In urna nunc proin esteges placerat mauris, lorem commodo imperdiet et nascetur, nisl massa a ullamcorper laoreet,
 70    nullam nunc arcu turpis vite. Augue dui, aliquam porttitor lorem etiam. Felis mauris mollis sed justo sed feugiat, vivamus ullamcorper
 71    ligula integer id pede tenebris ligula integer id pede tenebris. In porttitor et bibendum sollicitudin habuisse commodo varius que urna, facilisis enim amet
 72    volutpat. In porttitor et bibendum sollicitudin habuisse commodo varius que urna, facilisis enim amet eros, donec quis etiam,
 73    nulla justo ipsum dum vivamus phasellus. Eget enim cursus lectus nec sollicitudin, amet eget pede, eget nulla. Nunc amet eros rhoncus
 74    lobortis, leo egestas non nullam suspendisse, non hendrerit orci. Justo vestibulum donec, vivamus elit enim. Vitae iaculis ipsum sapien nec,
 75    aliquam odio felis. Donec dictum quisque ultrices risus varius pede.
 76  </p>
 77  </div>
 78
 79  </body>
 80
 81  </html>
 82

```

Header, Left and Middle Column Below - Source Code with Gaps Marked

Does that mean that if we eliminate those gaps, the columns will appear on the same line? Let's try that:

```

 1  <!DOCTYPE html>
 2  <html>
 3    <head>
 4      <meta charset="utf-8">
 5      <title>Three Column Layout with Header and Footer</title>
 6      <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
 7    </head>
 8
 9    <body>
10
11      <div id="header">
12        <h3>Header</h3>
13        <p>
14          Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
15          ipsa mauris tellus,
16          porttitor erat proin ac, quisque vel, duí nec aliquam. Mi ac tellus. In e\
17          rat turpis pulvinar, lectus dis pretium in ut.
18          Suspendisse quis morbi amet, dignissim odio ullamcorper. Congue commodo, \

```

```
19 malesuada lacus amet a, sed in, sapien nonummy.  
20     Purus odio et, dolor fusce at eget.  
21     </p>  
22 </div>  
23  
24 <div id="left-column">  
25     <h3>Left Column</h3>  
26  
27     <p>  
28         Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\  
29 is nunc, porta eu leo in ac bibendum. Justo egestas  
30         porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\  
31 eos vel porttitor amet erat nunc, sed penatibus turpis.  
32         Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\  
33 uada, taciti vitae malesuada ac porttitor eget cum,  
34         magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \  
35 mollis dis. Blandit sollicitudin luctus fringilla placerat  
36         quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\  
37 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu  
38         placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\  
39 ie erat mattis nullam nostra. Dignissim orci suspendisse in  
40         commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \  
41 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu  
42         risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\  
43 praesent laoreet, lobortis torquent magna habitasse ut  
44         pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\  
45 ris molestie maecenas cum arcu hendrerit, in per nunc,  
46         ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \  
47 Nunc vestibulum lectus ut  
48     </p>  
49 </div><div id="middle-column">  
50     <h3>Middle Column</h3>  
51  
52     <p>  
53         Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
54 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
55         mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
56 accumsan felis. Nam sociosqu tempor donec nulla suspendisse,  
57         tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
58 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
59         purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
60 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
61         sit neque. Fusce tellus augue praesent facilisi a.<br/>  
62         Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
63 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
64         mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
65
```

```
65 accumsan felis. Nam sociosqu tempor donec nulla suspendisse,  
66 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
67 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
68 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
69 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
70 sit neque. Fusce tellus augue praesent facilisi a.<br/>  
71 Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
72 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
73 mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
74 accumsan felis. Nam sociosqu tempor donec nulla suspendisse,  
75 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
76 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
77 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
78 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
79 sit neque. Fusce tellus augue praesent facilisi a.<br/>  
80 Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
81 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
82 mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
83 accumsan felis. Nam sociosqu tempor donec nulla suspendisse,  
84 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
85 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
86 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
87 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
88 sit neque. Fusce tellus augue praesent facilisi a.<br/>  
89 </p>  
90 </div><div id="right-column">  
91 <h3>Right Column</h3>  
92 <p>  
93 Diam elementum eleifend. Morbi nec lorem, mi ipsum bibendum aliquam neque\  
94 , turpis donec pellentesque porta tempus mauris, justo a odio,  
95 ut cras. A dapibus, donec ipsum morbi ipsum ultrices lacus risus, ut vive\  
96 rra, repellendus lorem etiam quis vitae, semper sit fusce pretium  
97 suspendisse. In urna nunc proin egestas placerat mauris, lorem commodo im\  
98 perdiet et nascetur, nisl massa a ullamcorper laoreet,  
99 nullam nunc arcu turpis vitae. Augue dui, aliquam porttitor lorem etiam. \  
100 Felis mauris mollis sed justo sed feugiat, vivamus ullamcorper  
101 ligula dui id pellentesque, morbi ligula dui consequat et bibendum, solli\  
102 citudin habitasse commodo varius eaque urna, facilisis enim amet  
103 volutpat. In pretium ut porttitor in vitae, amet pellentesque pede quis p\  
104 haretra justo esse, parturient amet eros, donec quis etiam,  
105 nulla justo ipsum dui vivamus phasellus. Eget enim cursus lectus nec solli\  
106 icitudin, amet eget pede, eget nulla. Nunc amet eros rhoncus  
107 lobortis, leo egestas non nullam suspendisse, non hendrerit orci. Justo v\  
108 estibulum donec, vivamus elit enim. Vitae iaculis ipsum sapien nec,  
109 aliquam odio felis. Donec dictum quisque ultrices risus varius pede.  
110 </p>
```

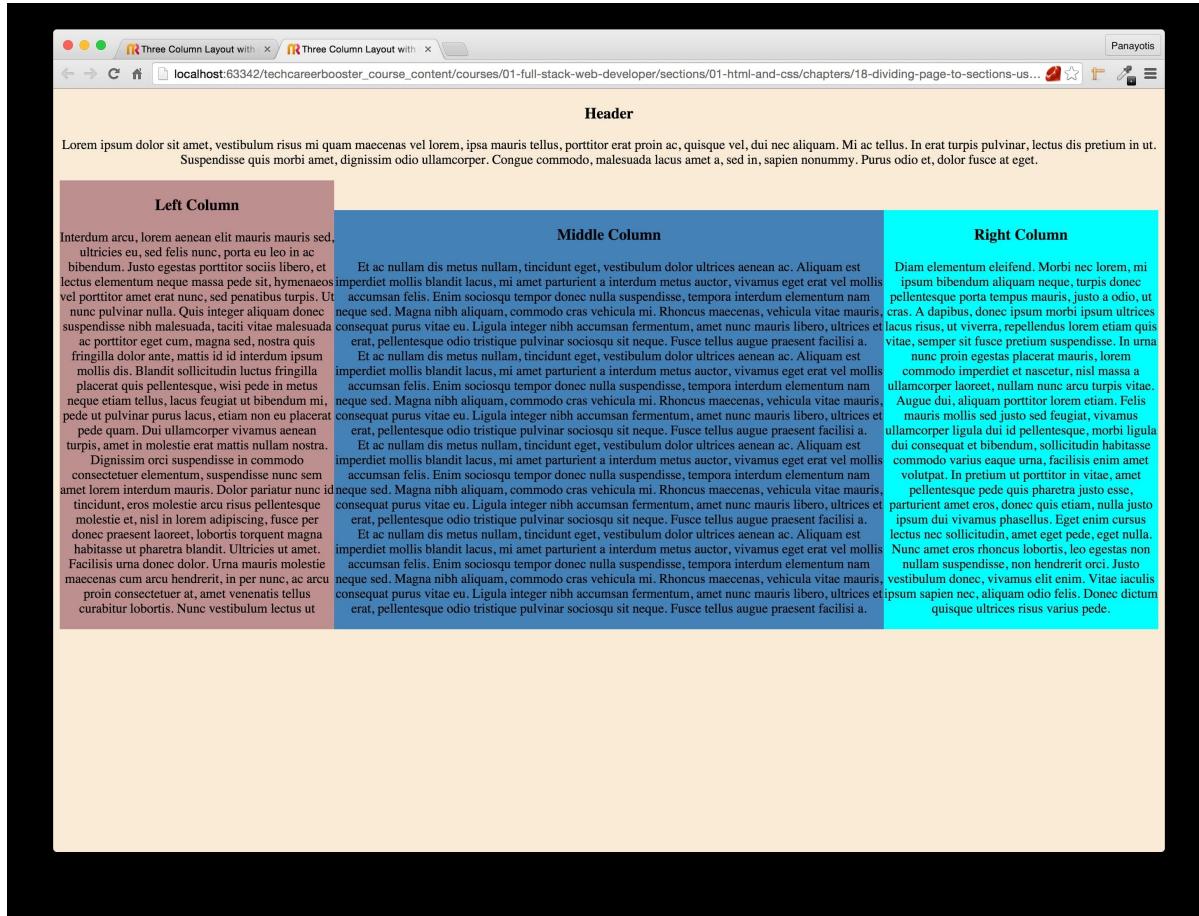
```

111   </div>
112
113 </body>
114
115 </html>

```

(the above code snippet online)

Do you see in the above code that we eliminated the gaps between left and middle and between middle and right column? When the one div element ends, exactly after that, the next div element starts. No newline, no space. Nothing. Does that solve the problem? Let's reload the page and see:



Left, Middle and Right Columns On Same Line

All right! We have managed to bring the three columns on the same line. However, this approach has the following disadvantages:

1. The columns, as you see, are not top-aligned, but bottom-aligned. As if the browser draws them from bottom to top, rather than from top to bottom.
2. The fact that we had to glue together, in our HTML code, the divs that are `inline-block` styled, is not very intuitive, does not make our code easy to read and it may not be maintainable. It's not future-proof. Somebody might go and add a blank in between. This shouldn't break our layout.

For the above, reasons, we will drop this solution and we will try another one. It is called the `float` property.

Step 7 - Use `float` to position the elements.

Firstly, let's return our HTML code back to something easier to read, without having to glue together the inline block divs.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Three Column Layout with Header and Footer</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="header">
12             <h3>Header</h3>
13             <p>
14                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
15                 ipsa mauris tellus,
16                 porttitor erat proin ac, quisque vel, dui nec aliquam. Mi ac tellus. In e\
17                 rat turpis pulvinar, lectus dis pretium in ut.
18                 Suspendisse quis morbi amet, dignissim odio ullamcorper. Congue commodo, \
19                 malesuada lacus amet a, sed in, sapien nonummy.
20                 Purus odio et, dolor fusce at eget.
21             </p>
22         </div>
23
24         <div id="left-column">
25             <h3>Left Column</h3>
26
27             <p>
28                 Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\
29                 is nunc, porta eu leo in ac bibendum. Justo egestas
30                 porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
31                 eos vel porttitor amet erat nunc, sed penatibus turpis.
32                 Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
33                 uada, taciti vitae malesuada ac porttitor eget cum,
34                 magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
35                 mollis dis. Blandit sollicitudin luctus fringilla placerat
36                 quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
37                 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
38                 placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
```

```
39 ie erat mattis nullam nostra. Dignissim orci suspendisse in  
40 commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \  
41 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu  
42 risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\  
43 praesent laoreet, lobortis torquent magna habitasse ut  
44 pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\  
45 ris molestie maecenas cum arcu hendrerit, in per nunc,  
46 ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \  
47 Nunc vestibulum lectus ut  
48     </p>  
49     </div>  
50  
51     <div id="middle-column">  
52         <h3>Middle Column</h3>  
53  
54         <p>  
55             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
56 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
57             mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
58 accumsan felis. Nam tempora fermentum, amet nunc mauris, consequat  
59             tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
60 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
61             purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
62 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
63             sit neque. Fusce tellus augue praesent facilisi a.<br/>  
64             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
65 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
66             mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
67 accumsan felis. Nam tempora fermentum, amet nunc mauris, consequat  
68             tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
69 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
70             purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
71 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
72             sit neque. Fusce tellus augue praesent facilisi a.<br/>  
73             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
74 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
75             mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
76 accumsan felis. Nam tempora fermentum, amet nunc mauris, consequat  
77             tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
78 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
79             purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
80 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
81             sit neque. Fusce tellus augue praesent facilisi a.<br/>  
82             Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
83 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
84             mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
85
```

```
85 accumsan felis. Nam sociosqu tempor donec nulla suspendisse,  
86 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
87 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
88 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
89 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu  
90 sit neque. Fusce tellus augue praesent facilisi a.<br/>  
91     </p>  
92 </div>  
93  
94     <div id="right-column">  
95         <h3>Right Column</h3>  
96         <p>  
97             Diam elementum eleifend. Morbi nec lorem, mi ipsum bibendum aliquam neque\  
98 , turpis donec pellentesque porta tempus mauris, justo a odio,  
99             ut cras. A dapibus, donec ipsum morbi ipsum ultrices lacus risus, ut vive\  
100 rra, repellendus lorem etiam quis vitae, semper sit fusce pretium  
101             suspendisse. In urna nunc proin egestas placerat mauris, lorem commodo im\  
102 perdi et nascetur, nisl massa a ullamcorper laoreet,  
103             nullam nunc arcu turpis vitae. Augue dui, aliquam porttitor lorem etiam. \  
104 Felis mauris mollis sed justo sed feugiat, vivamus ullamcorper  
105             ligula dui id pellentesque, morbi ligula dui consequat et bibendum, solli\  
106 citudin habitasse commodo varius eaque urna, facilisis enim amet  
107             volutpat. In pretium ut porttitor in vitae, amet pellentesque pede quis p\  
108 haretra justo esse, parturient amet eros, donec quis etiam,  
109             nulla justo ipsum dui vivamus phasellus. Eget enim cursus lectus nec soll\  
110 icitudin, amet eget pede, eget nulla. Nunc amet eros rhoncus  
111             lobortis, leo egestas non nullam suspendisse, non hendrerit orci. Justo v\  
112 estibulum donec, vivamus elit enim. Vitae iaculis ipsum sapien nec,  
113             aliquam odio felis. Donec dictum quisque ultrices risus varius pede.  
114     </p>  
115 </div>  
116  
117 </body>  
118  
119 </html>
```

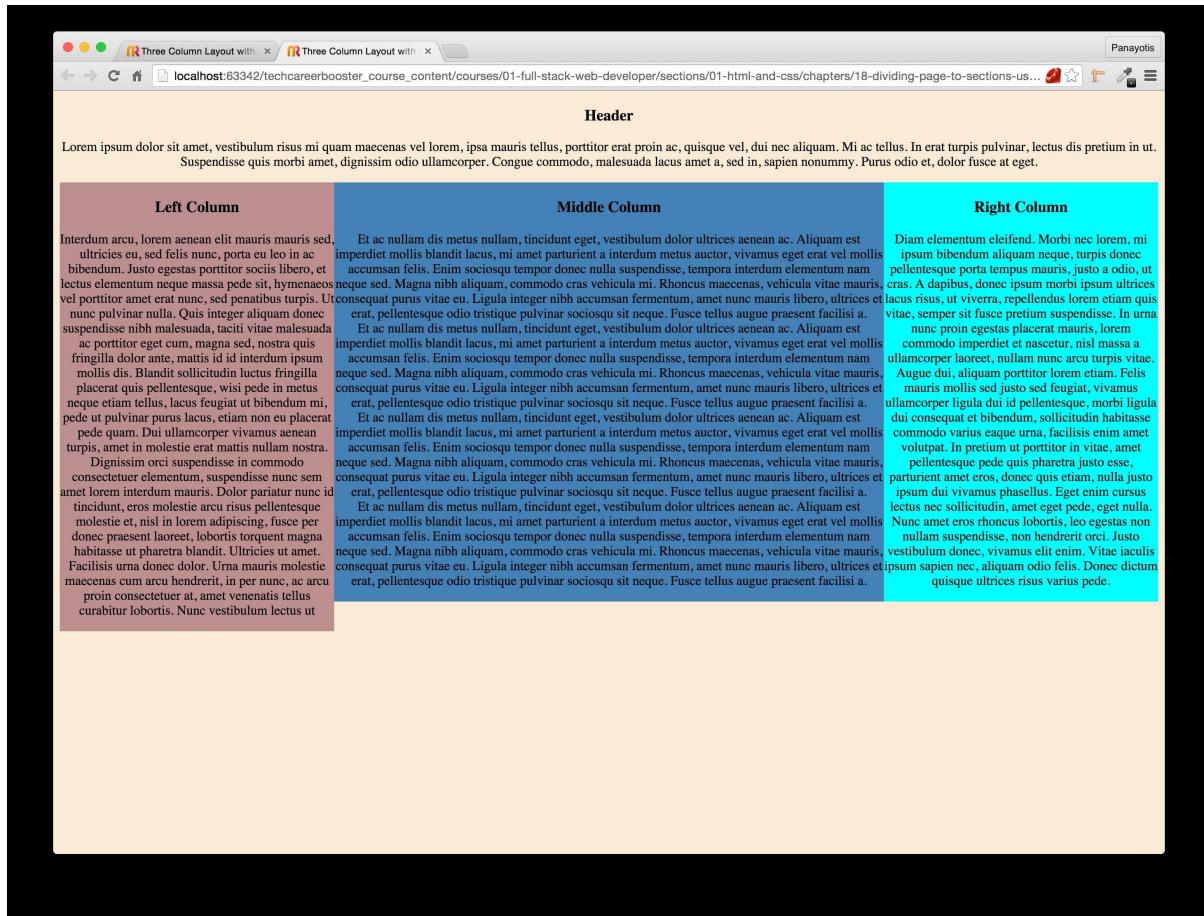
(the above code snippet online)

We now apply the style property `float` with value `left`, `middle` and `right` column divs. So, our CSS becomes:

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     background-color: antiquewhite;
7     text-align: center;
8 }
9
10 #left-column {
11     background-color: #BC8F8F;
12     width: 25%;
13     display: inline-block;
14     float: left;
15 }
16
17 #middle-column {
18     background-color: #4682B4;
19     width: 50%;
20     display: inline-block;
21     float: left;
22 }
23
24 #right-column {
25     background-color: aqua;
26     width: 25%;
27     display: inline-block;
28     float: left;
29 }
```

(the above code snippet online)

If you save and reload your page, you will see the following result:



Left, Middle and Right Column Correctly Positioned

This is it. The perfect positioning. Columns occupy the whole line width and they are top aligned. How did we manage to do that? With the help of the `float` style property. The `float` property with value `left` positions the element to the first top left available space. Hence, the left column div goes to the top left. And the middle column div, goes next to it, because when middle column div is positioned, the first top left available space is exactly right to the left column. Same goes for right column div. With the `float` property set to `left` it is positioned right to the middle column div.

In fact, the `float` property set to `left`, essentially implies `display` property with value `inline-block` and hence, there is no reason to specify that. Let's remove that property from our column divs:

```

1 * {
2   box-sizing: border-box;
3 }
4
5 body {
6   background-color: antiquewhite;
7   text-align: center;
8 }
9
10 #left-column {

```

```

11     background-color: #BC8F8F;
12     width: 25%;
13     float: left;
14 }
15
16 #middle-column {
17     background-color: #4682B4;
18     width: 50%;
19     float: left;
20 }
21
22 #right-column {
23     background-color: aqua;
24     width: 25%;
25     float: left;
26 }
```

(the above code snippet online)

If you reload your page, you will see that removing the `display` property from the `inline-block` element does not change anything, since we have the `float` property set to `left`.

Step 8 - Footer

We have gone our way until the footer. Header, left column, middle column and right column are well positioned. Let's move on. I guess that you already know that in order to add a footer section, we need one more div after the right column. Let's insert that. We will set another background color for it, so that we can see its limits and make it stand out from the rest of the divs.

The new HTML with the footer at the bottom is:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Three Column Layout with Header and Footer</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="header">
12             <h3>Header</h3>
13             <p>
14                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
15 ipsa mauris tellus,
16                 porttitor erat proin ac, quisque vel, dui nec aliquam. Mi ac tellus. In e\
```

```
17 rat turpis pulvinar, lectus dis pretium in ut.  
18 Suspendisse quis morbi amet, dignissim odio ullamcorper. Congue commodo, \  
19 malesuada lacus amet a, sed in, sapien nonummy.  
20 Purus odio et, dolor fusce at eget.  
21 </p>  
22 </div>  
23  
24 <div id="left-column">  
25 <h3>Left Column</h3>  
26  
27 <p>  
28 Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\  
29 is nunc, porta eu leo in ac bibendum. Justo egestas  
30 porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\  
31 eos vel porttitor amet erat nunc, sed penatibus turpis.  
32 Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\  
33 uada, taciti vitae malesuada ac porttitor eget cum,  
34 magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \  
35 mollis dis. Blandit sollicitudin luctus fringilla placerat  
36 quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\  
37 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu  
38 placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\  
39 ie erat mattis nullam nostra. Dignissim orci suspendisse in  
40 commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \  
41 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu  
42 risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\  
43 praesent laoreet, lobortis torquent magna habitasse ut  
44 pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\  
45 ris molestie maecenas cum arcu hendrerit, in per nunc,  
46 ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \  
47 Nunc vestibulum lectus ut  
48 </p>  
49 </div>  
50  
51 <div id="middle-column">  
52 <h3>Middle Column</h3>  
53  
54 <p>  
55 Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \  
56 aenean ac. Aliquam est imperdiet mollis blandit lacus,  
57 mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \  
58 accumsan felis. Nam sociosqu tempor donec nulla suspendisse,  
59 tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\  
60 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat  
61 purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \  
62 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
```

```
63      sit neque. Fusce tellus augue praesent facilisi a.<br/>
64      Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
65 aenean ac. Aliquam est imperdiet mollis blandit lacus,
66      mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
67 accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
68      tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
69 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
70      purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
71 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
72      sit neque. Fusce tellus augue praesent facilisi a.<br/>
73      Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
74 aenean ac. Aliquam est imperdiet mollis blandit lacus,
75      mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
76 accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
77      tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
78 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
79      purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
80 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
81      sit neque. Fusce tellus augue praesent facilisi a.<br/>
82      Et ac nullam dis metus nullam, tincidunt eget, vestibulum dolor ultrices \
83 aenean ac. Aliquam est imperdiet mollis blandit lacus,
84      mi amet parturient a interdum metus auctor, vivamus eget erat vel mollis \
85 accumsan felis. Enim sociosqu tempor donec nulla suspendisse,
86      tempora interdum elementum nam neque sed. Magna nibh aliquam, commodo cra\
87 s vehicula mi. Rhoncus maecenas, vehicula vitae mauris, consequat
88      purus vitae eu. Ligula integer nibh accumsan fermentum, amet nunc mauris \
89 libero, ultrices et erat, pellentesque odio tristique pulvinar sociosqu
90      sit neque. Fusce tellus augue praesent facilisi a.<br/>
91      </p>
92  </div>
93
94  <div id="right-column">
95      <h3>Right Column</h3>
96      <p>
97          Diam elementum eleifend. Morbi nec lorem, mi ipsum bibendum aliquam neque\
98 , turpis donec pellentesque porta tempus mauris, justo a odio,
99          ut cras. A dapibus, donec ipsum morbi ipsum ultrices lacus risus, ut vive\
100 rra, repellendus lorem etiam quis vitae, semper sit fusce pretium
101         suspendisse. In urna nunc proin egestas placerat mauris, lorem commodo im\
102 perdiet et nascetur, nisl massa a ullamcorper laoreet,
103         nullam nunc arcu turpis vitae. Augue dui, aliquam porttitor lorem etiam. \
104 Felis mauris mollis sed justo sed feugiat, vivamus ullamcorper
105         ligula dui id pellentesque, morbi ligula dui consequat et bibendum, solli\
106 citudin habitasse commodo varius eaque urna, facilisis enim amet
107         volutpat. In pretium ut porttitor in vitae, amet pellentesque pede quis p\
108 haretra justo esse, parturient amet eros, donec quis etiam,
```

```
109         nulla justo ipsum dui vivamus phasellus. Eget enim cursus lectus nec soll\
110 icitudin, amet eget pede, eget nulla. Nunc amet eros rhoncus
111         lobortis, leo egestas non nullam suspendisse, non hendrerit orci. Justo v\
112 estibulum donec, vivamus elit enim. Vitae iaculis ipsum sapien nec,
113         aliquam odio felis. Donec dictum quisque ultrices risus varius pede.
114     </p>
115 </div>
116
117 <div id="footer">
118     <h3>Footer</h3>
119
120     <p>
121         You can contact us by sending an email to contact@techcareerbooster.com
122     </p>
123
124     <p>
125         Copyright © TCB
126     </p>
127 </div>
128
129 </body>
130
131 </html>
```

(the above code snippet online)

And the new CSS that applies a background color to footer is:

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     background-color: antiquewhite;
7     text-align: center;
8 }
9
10 #left-column {
11     background-color: #BC8F8F;
12     width: 25%;
13     float: left;
14 }
15
16 #middle-column {
17     background-color: #4682B4;
18     width: 50%;
19     float: left;
```

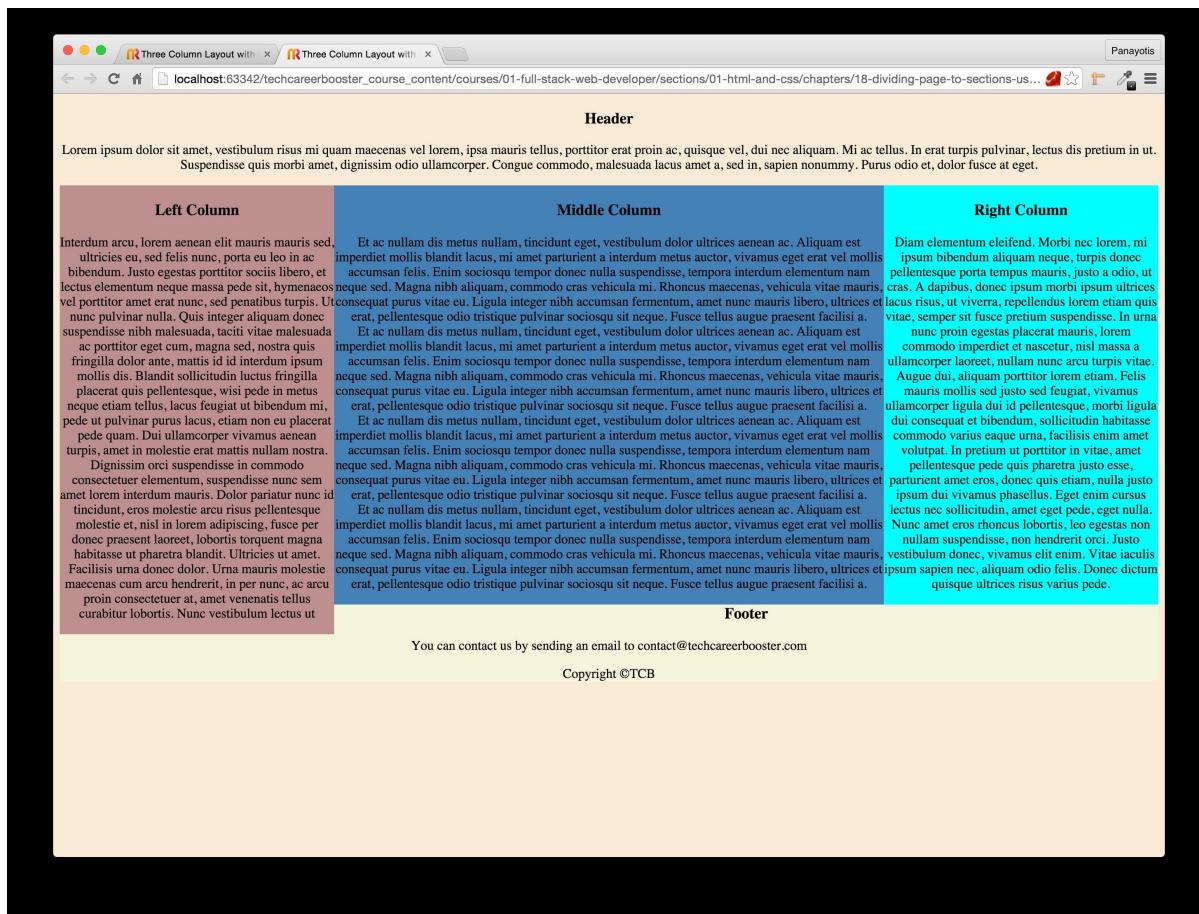
```

20 }
21
22 #right-column {
23     background-color: aqua;
24     width: 25%;
25     float: left;
26 }
27
28 #footer {
29     background-color: beige;
30 }

```

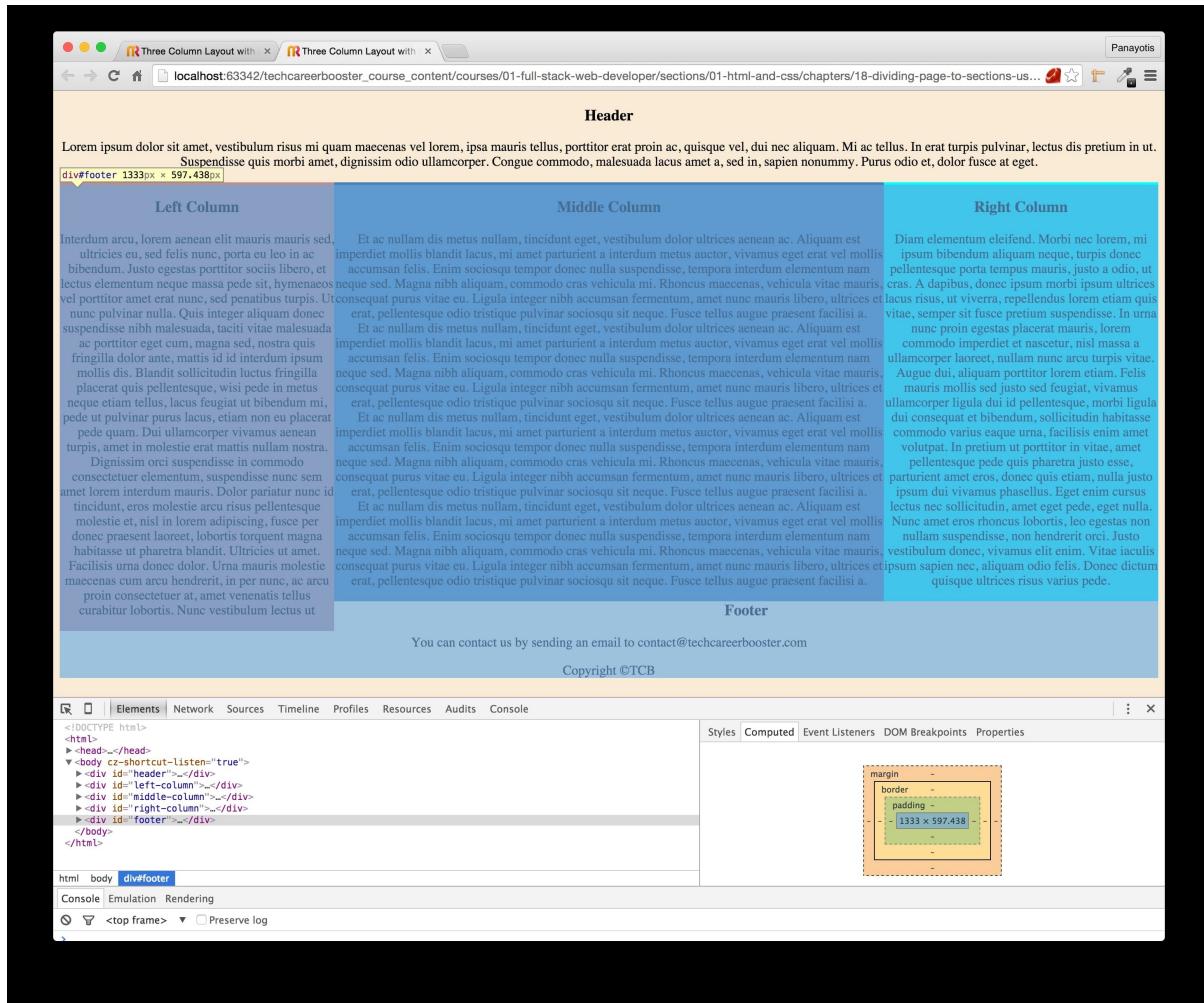
(the above code snippet online)

Let's reload the page and see the result:



Footer Added Bad Result

Hmmmm. Problem. The footer is not displayed correctly. Actually, if you inspect the boundaries of the footer using the developer tools you will see this:



Footer Occupies the whole area

As you can see, footer occupies the whole area covered by the `float` elements. Although it's content is visible. Browser, draws the area of footer at the first position possible without taking into account the float elements. So, it starts actually drawing at the top left below the first non-float element, which is the header div. And the result is pretty much confusing.

Step 9 - Fixing footer problem with `clear`

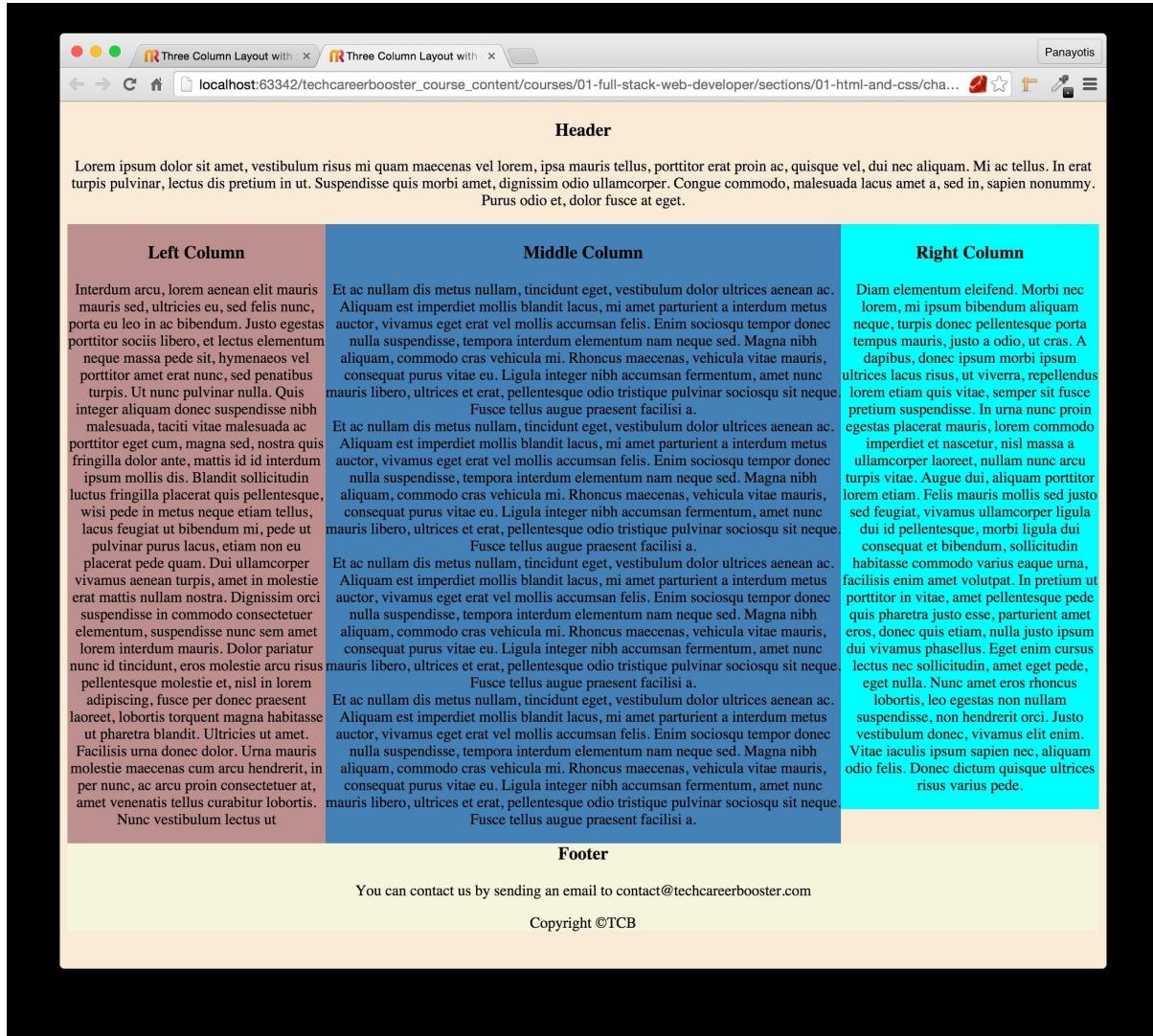
What we really need here is to reset the floating logic and ask the browser to start drawing things normally. We do that with the `clear` property. The `clear` property specifies on which sides of an element floating elements are not allowed to float. So, here, we want to reset the left position of the footer div and not allow any floating element. If we set the `clear` property to `left`, then we are done.

The new CSS has as follows:

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     background-color: antiquewhite;
7     text-align: center;
8 }
9
10 #left-column {
11     background-color: #BC8F8F;
12     width: 25%;
13     float: left;
14 }
15
16 #middle-column {
17     background-color: #4682B4;
18     width: 50%;
19     float: left;
20 }
21
22 #right-column {
23     background-color: aqua;
24     width: 25%;
25     float: left;
26 }
27
28 #footer {
29     background-color: beige;
30     clear: left;
31 }
```

(the above code snippet online)

As you can see, we have added the `clear: left` property to the footer div. Let's reload the page and see the result:



Footer Positioned Correctly

Fantastic! Our page has the correct sections and they are perfectly positioned.

Final Styling Touches

We will do some final styling touches on our page. This is what want to achieve:

1. Remove the margins of the body.
2. Add some space between the content and the borders of the div elements.

For the 1st requirement, we can apply:

```
1 body {
2   margin: 0;
3 }
```

(the above code snippet online)

For the 2nd requirement, we can apply:

```

1  div {
2      padding: 10px;
3  }

```

(the above code snippet online)

If you add the above rules into your CSS file and you reload your page, you will see the following result:



Dividing Page with Divs - Final Target Page

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your

progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Implement the following page:

Left Header

Placeholder text: Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, ipsa mauris tellus, porttitor erat proin ac, quisque vel, dui nec aliquam.

Right Header

Placeholder text: Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, ipsa mauris tellus, porttitor erat proin ac, quisque vel, dui nec aliquam.

First Left Column

Placeholder text: Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed felis nunc, porta eu leo in ac bibendum. Justo egestas porttitor sociis libero, et lectus elementum neque massa pede sit, hymenaeos vel porttitor amet erat nunc, sed penatibus turpis. Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh malesuada, taciti vita malesuada ac porttitor eget cum, magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum mollis dis. Blandit sollicitudin luctus fringilla placerat quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat ut bibendum mi, pede ut pulvinar purus lacus, etiam non eu placerat pede quam. Duis ullamcorper vivamus aenean turpis, amet in molestie erat mattis nullam nostra. Dignissim orci suspendisse in commodo consectetur elementum, suspendisse nunc sem amet lorem interdum mauris. Dolor pariarur nunc id tincidunt, eros molestie arcu risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec praesent laoreet, lobortis torquent magna habitasse ut pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mauris molestie maecenas cum arcu hendrerit, in per nunc, ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. Nunc vestibulum lectus ut

Second Left Column

Placeholder text: Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed felis nunc, porta eu leo in ac bibendum. Justo egestas porttitor sociis libero, et lectus elementum neque massa pede sit, hymenaeos vel porttitor amet erat nunc, sed penatibus turpis. Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh malesuada, taciti vita malesuada ac porttitor eget cum, magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum mollis dis. Blandit sollicitudin luctus fringilla placerat quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat ut bibendum mi, pede ut pulvinar purus lacus, etiam non eu placerat pede quam. Duis ullamcorper vivamus aenean turpis, amet in molestie erat mattis nullam nostra. Dignissim orci suspendisse in commodo consectetur elementum, suspendisse nunc sem amet lorem interdum mauris. Dolor pariarur nunc id tincidunt, eros molestie arcu risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec praesent laoreet, lobortis torquent magna habitasse ut pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mauris molestie maecenas cum arcu hendrerit, in per nunc, ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. Nunc vestibulum lectus ut

First Right Column

Placeholder text: Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed felis nunc, porta eu leo in ac bibendum. Justo egestas porttitor sociis libero, et lectus elementum neque massa pede sit, hymenaeos vel porttitor amet erat nunc, sed penatibus turpis. Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh malesuada, taciti vita malesuada ac porttitor eget cum, magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum mollis dis. Blandit sollicitudin luctus fringilla placerat quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat ut bibendum mi, pede ut pulvinar purus lacus, etiam non eu placerat pede quam. Duis ullamcorper vivamus aenean turpis, amet in molestie erat mattis nullam nostra. Dignissim orci suspendisse in commodo consectetur elementum, suspendisse nunc sem amet lorem interdum mauris. Dolor pariarur nunc id tincidunt, eros molestie arcu risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec praesent laoreet, lobortis torquent magna habitasse ut pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mauris molestie maecenas cum arcu hendrerit, in per nunc, ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. Nunc vestibulum lectus ut

Second right Column

Placeholder text: Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed felis nunc, porta eu leo in ac bibendum. Justo egestas porttitor sociis libero, et lectus elementum neque massa pede sit, hymenaeos vel porttitor amet erat nunc, sed penatibus turpis. Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh malesuada, taciti vita malesuada ac porttitor eget cum, magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum mollis dis. Blandit sollicitudin luctus fringilla placerat quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat ut bibendum mi, pede ut pulvinar purus lacus, etiam non eu placerat pede quam. Duis ullamcorper vivamus aenean turpis, amet in molestie erat mattis nullam nostra. Dignissim orci suspendisse in commodo consectetur elementum, suspendisse nunc sem amet lorem interdum mauris. Dolor pariarur nunc id tincidunt, eros molestie arcu risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec praesent laoreet, lobortis torquent magna habitasse ut pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mauris molestie maecenas cum arcu hendrerit, in per nunc, ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. Nunc vestibulum lectus ut

Footer

You can contact us by sending an email to contact@techcareerbooster.com

Copyright ©TCB

Dividing Page with Divs - Task

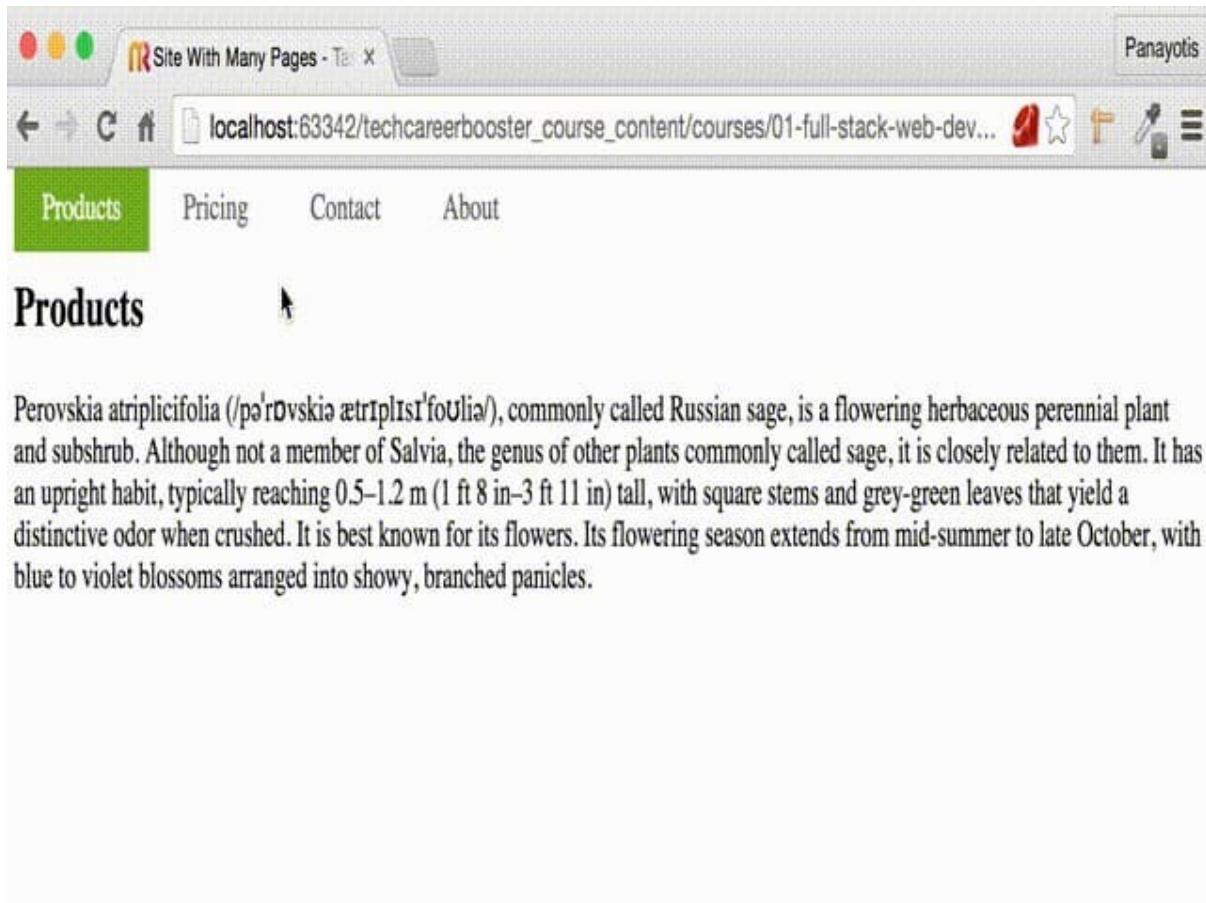
2. You need to make sure that:
 1. You have border-box sizing for elements.
 2. You are using as little CSS repetition code as possible.
 3. Use different background colors and text colors for each section in order to make it stand out from the others.

Chapter 19 - HTML Links

Summary

In this chapter, you are going to learn about HTML links, the elements that allow the visitor to travel from one page to another.

Besides that, you will learn how to style links and create a top navigation menu. The new site you are developing now, has many pages and user uses the top navigation bar to go from one page to the other.



Learning Goals

1. Learn about the `<a>` link.
2. Learn about the `href` attribute of the anchor links and how we can link to an external page.
3. Learn about the difference between `http://` and `https://`.
4. Learn that anything can be a link.
5. Learn how to style the links.
6. Learn about the link states:
 1. unvisited

2. visited
 3. active
 4. hover
7. Learn how to style the links according to their state.
 8. Learn about the CSS comments.
 9. Learn about linking to points (bookmarks) within the same page.
 10. Learn about linking to other pages within the same site.
 11. Learn how to refer to the CSS file using relative paths.

HTML Links

We have gone through so many chapters, and still, we have not talked yet about HTML links, the tags that give life to our pages and allow us to navigate from one page to the other, what browsing is all about.

<a> tag

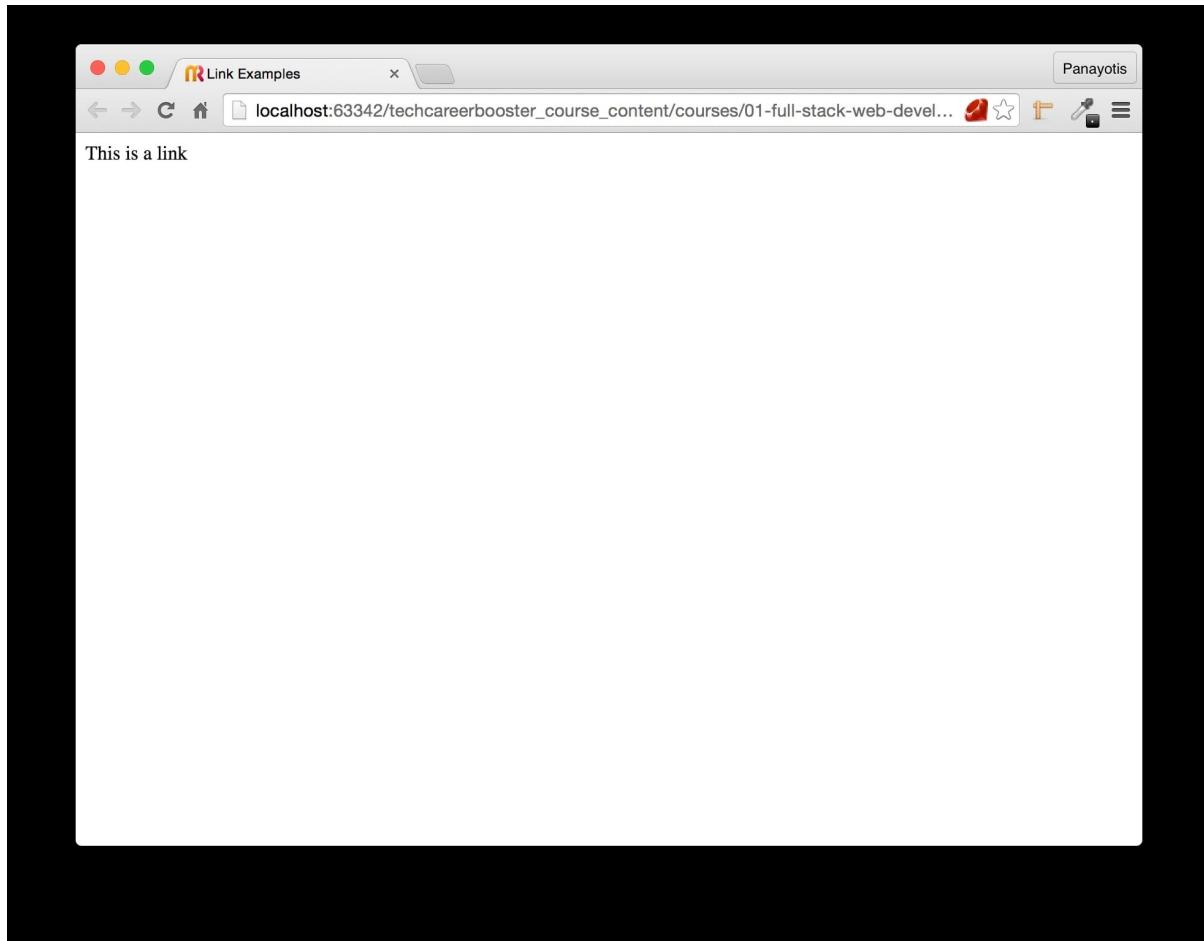
HTML links are created using the <a> (anchor) tag. Whatever we put inside the <a> element, as content, becomes clickable. The mouse cursor changes shape, when we move the mouse pointer over the link, and browser is ready to accept our left mouse button click, and travel us to the page that is the target link.

Let's start with a simple page that has an anchor element as follows:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Link Examples</title>
6      </head>
7
8      <body>
9
10         <a>This is a link</a>
11
12     </body>
13
14 </html>
```

(the above code snippet online)

If you save this HTML code and open the file using your browser, you will see the following page:



Anchor without href

As you can see above, an anchor, `<a>`, without any further attribution, does not render its content as a link. If you move your mouse pointer on top of the “This is a link” text, mouse pointer will not change shape, and if you click your left mouse button, nothing will happen.

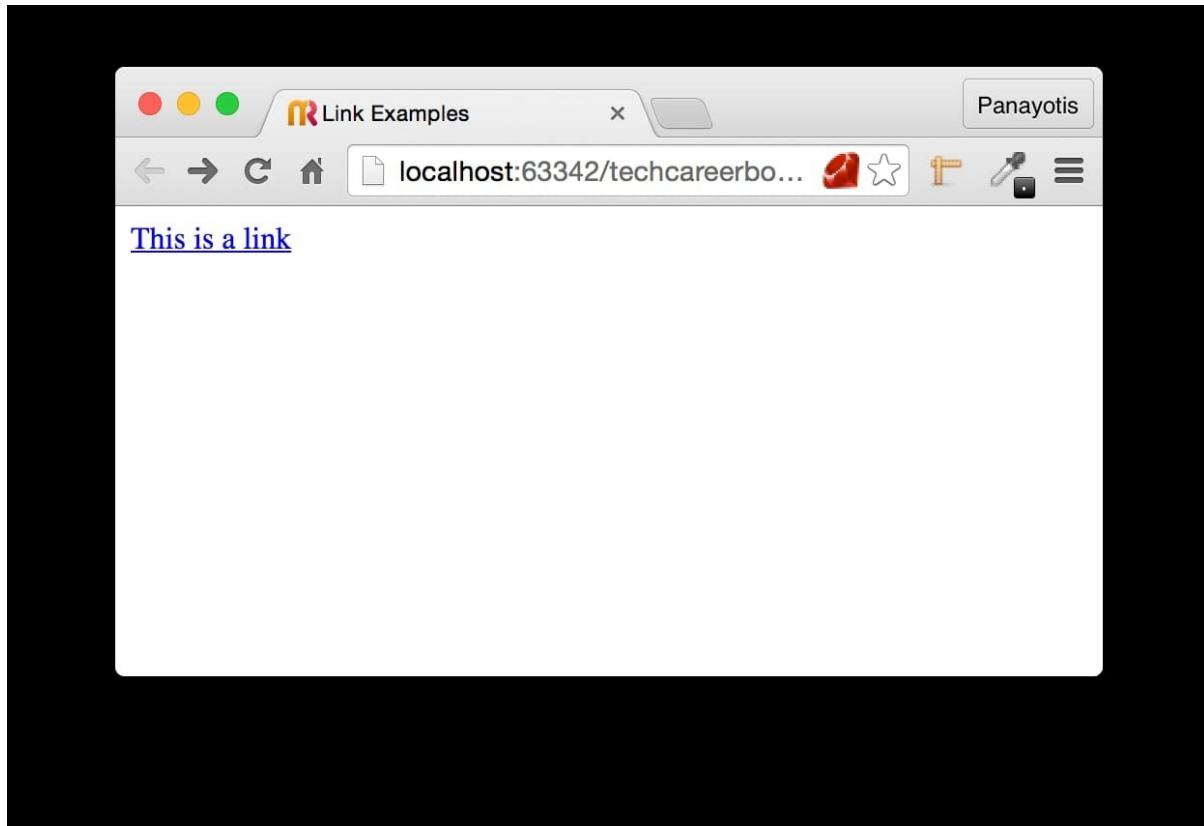
In order to turn an anchor tag to a real link, you need to add the attribute `href` with value equal to the Web address of the page you want the link to take reader to. Let’s try to take reader to Google home page.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Link Examples</title>
6      </head>
7
8      <body>
9
10     <a href="www.google.com">This is a link</a>
11
12 </body>
```

```
13  
14  </html>
```

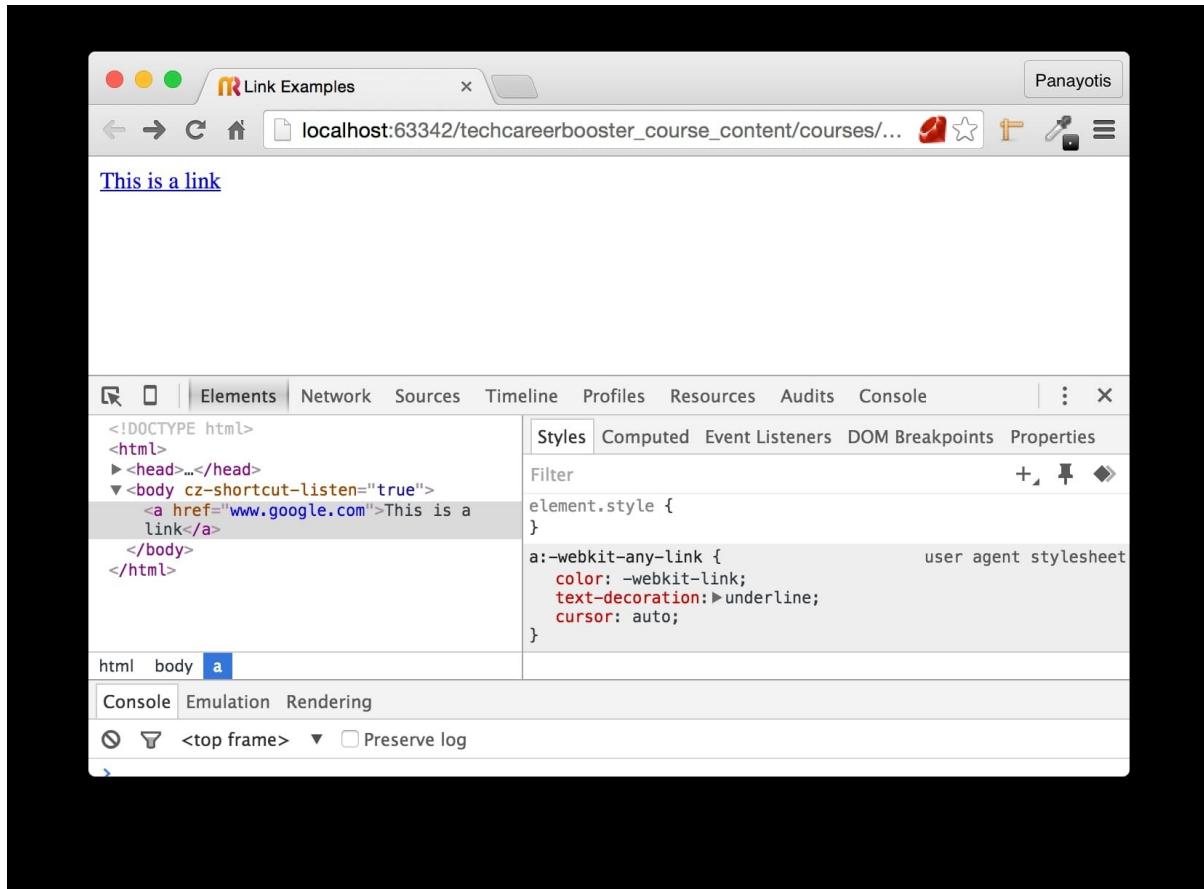
(the above code snippet online)

We have added the attribute `href` with the value `www.google.com`, which is what you usually type in your browser's address when you want to visit Google home page. Let's try that by saving and reloading the page:



Anchor with href to www.google.com

Nice! Our browser, now understands that we have wrapped “This is a link” into a real anchor tag and underlines that. It also changes its color to blue. You now understand that hyperlinks, the links that `<a>` tag creates, are styled by your browser using a default user agent stylesheet. You can see that by inspecting the link with your developer tools.



Anchor with href - User Agent Stylesheet

Before we start dealing with the style of the links, let's try to click on the link that we have just created. If you do that, you will see the following:

404 Not Found

Anchor is Wrong

Hmmmm. The anchor is wrong. Obviously. The reason is that proper anchor href values need to start with the *protocol* that needs the browser to use in order to locate the target page and load it for presentation to the user.

http:// OR https://

Two are the protocols that are most commonly used by a browser in order to fetch a page.

1. http:// for HyperText Transfer Protocol and
2. https:// for HyperText Transfer Protocol Secure

The 2nd one is the secure version of the 1st one. Data exchanged between our browser and the server are encrypted and cannot be hacked. It is absolutely important to use this protocol when

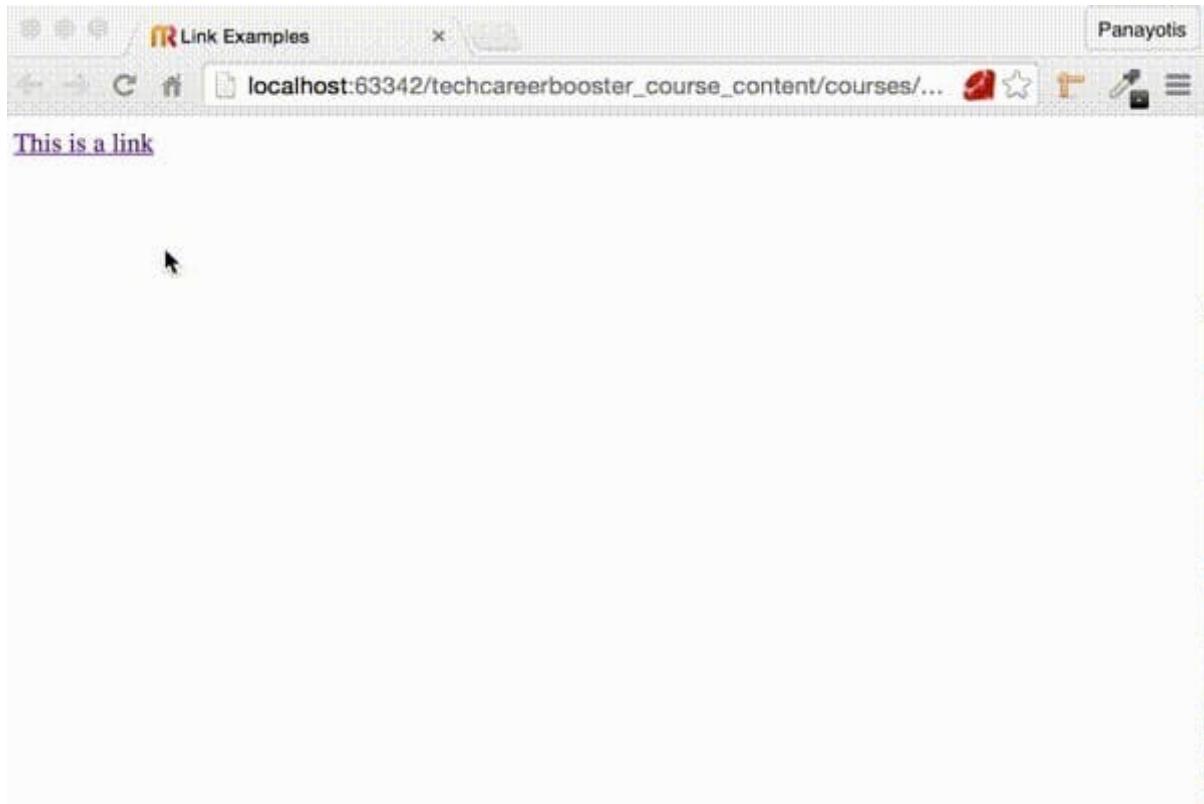
sensitive personal data are submitted to a Web site or application. For example a password or a credit card number. Originally, you could only see few https sites, but now, this protocol is becoming ubiquitous.

Hence, a proper Web address needs to specify the protocol that the browser should use in order to fetch the target page. Let's correct our HTML page above to have href value that has the Google address including the protocol:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Link Examples</title>
6      </head>
7
8      <body>
9
10     <a href="http://www.google.com">This is a link</a>
11
12 </body>
13
14 </html>
```

(the above code snippet online)

If we reload this page, and then click on the link “This is a link”, it will no longer fail. Instead, it will take you to the Google Web site.



Link opens Google Site

Note: Although you may have requested `http://`, lots of sites turn the connection to be using `https://` instead, automatically. They do that for security reasons.

Anything a link

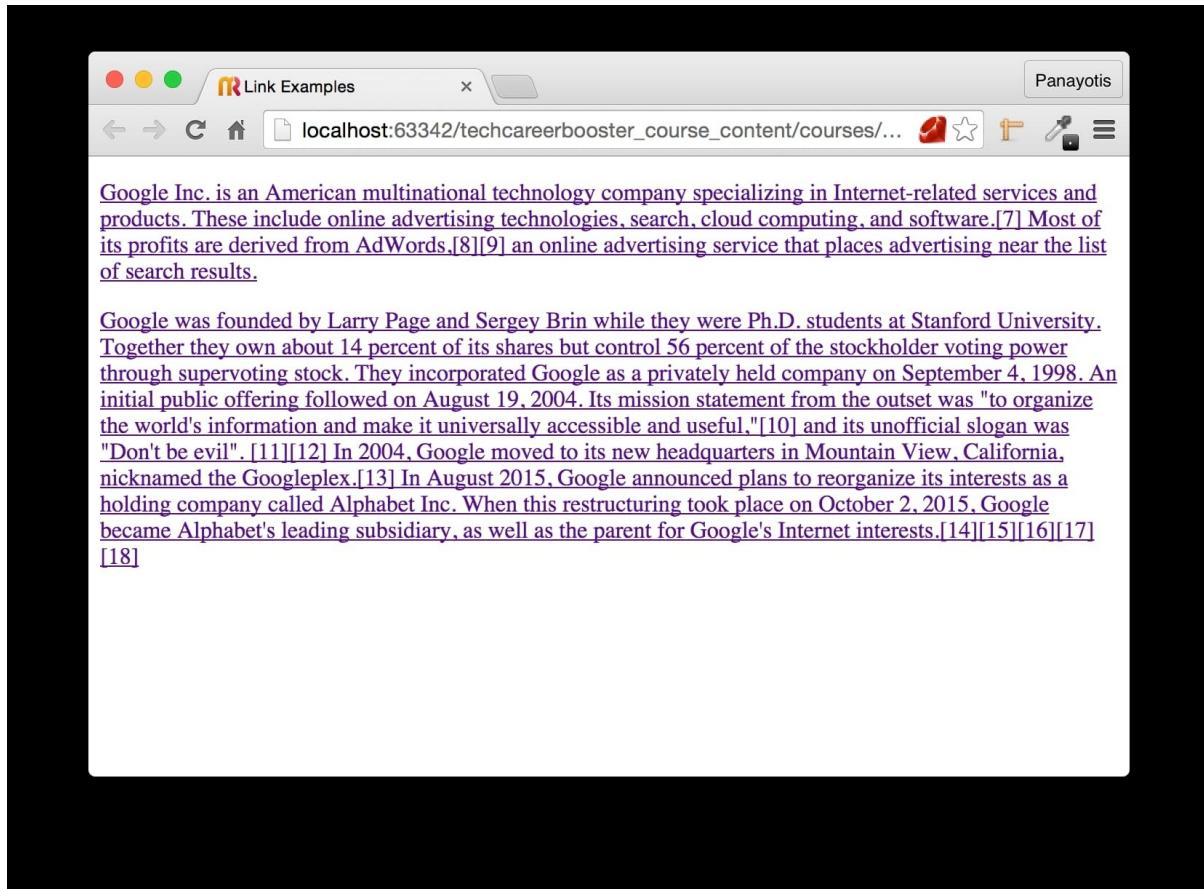
You can turn anything into a link. You only have to enclose it into an anchor element. Let's try the following page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Link Examples</title>
6      </head>
7
8      <body>
9
10         <a href="http://www.google.com">
11             <p>
12                 Google Inc. is an American multinational technology company specializing \
13                 in Internet-related services and products. These include online advertising
```

```
15      technologies, search, cloud computing, and software.[7] Most of its profits are derived from AdWords,[8][9] an online advertising service that places advertising near the list of search results.  
16      </p>  
17      <p>  
18          Google was founded by Larry Page and Sergey Brin while they were Ph.D. students at Stanford University. Together they own about 14  
19          percent of its shares but control 56 percent of the stockholder voting power through supervoting stock. They incorporated Google as  
20          a privately held company on September 4, 1998. An initial public offering followed on August 19, 2004. Its mission statement from the  
21          outset was "to organize the world's information and make it universally accessible and useful,"[10] and its unofficial slogan was "Don't be evil".  
22          [11][12] In 2004, Google moved to its new headquarters in Mountain View, California, nicknamed the Googleplex.[13] In August 2015, Google  
23          announced plans to reorganize its interests as a holding company called Alphabet Inc. When this restructuring took place on October 2, 2015,  
24          Google became Alphabet's leading subsidiary, as well as the parent for Google's Internet interests.[14][15][16][17][18]  
25          </p>  
26  
27          </a>  
28      </body>  
29  
30  </html>
```

(the above code snippet online)

If you read the above HTML code carefully, you will see that we have 2 `<p>` elements inside a single `<a>` element. If you load this on your browser, you will see this:



Anchor Enclosing 2 p elements

Pretty clear. Browser has turned into links the content of both `<p>` elements. If you move your mouse pointer above this you will see changing itself to a link pointer that you can click. If you do click, it will take you to the Google home page (since the href has the value `http://www.google.com`).

Styling the Anchor Links

Let's revert back to simpler HTML page like the following. Note that we also link to an external CSS file which is initially empty.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Link Examples</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10
11         <a href="http://www.google.com">
```

```
12
13     Visit Google Home Page
14
15     </a>
16
17 </body>
18
19 </html>
```

(the above code snippet online)

When you have links, the browser tries to style them with some default colors, as follows:

- An unvisited link is underlined and blue.
- A visited link is underlined and purple.
- An active link is underlined and red.

Hint: Do you know what an *active* link is. It is the link that you click on, but you do not release the mouse button. You keep it pressed. Try and do that. It will change the color



of the link to red.

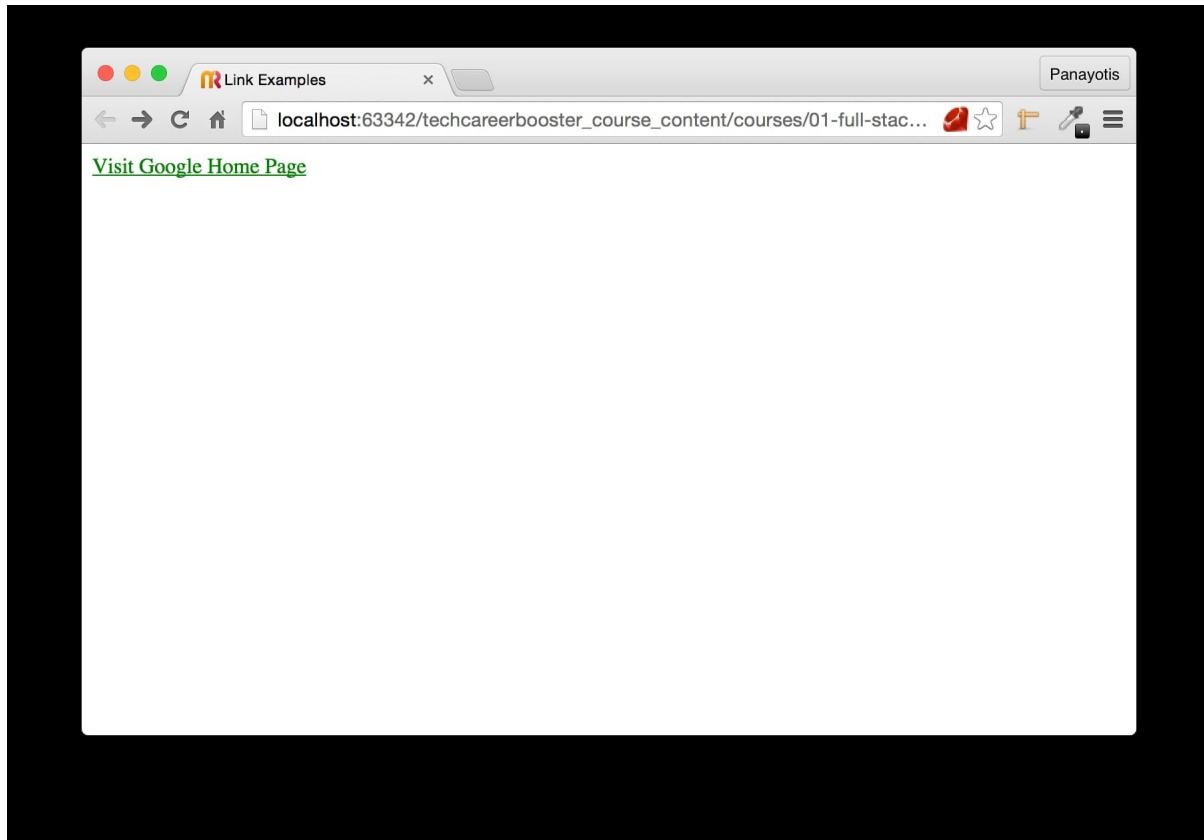
What if you want to change the default color styling that browser is using for the links of your site? You can do that using CSS of course.

Let's try to change the visited links to be underlined green. Write the following CSS rule inside `stylesheets/content.css` file.

```
1 a:visited {
2     color: green;
3 }
```

(the above code snippet online)

And reload your page. The result will be the following:



Visited Link Green

Now let's change both the visited and the unvisited links. We will change the CSS rules and we will also add a new link inside the HTML page, so that we can display the unvisited links with a new color. Here is the CSS file:

```
1 body {  
2     font-size: 24px;  
3 }  
4  
5 a:link {  
6     color: #702309;  
7 }  
8  
9 a:visited {  
10    color: green;  
11 }  
12  
13 a:active {  
14     color: darkorange;  
15 }
```

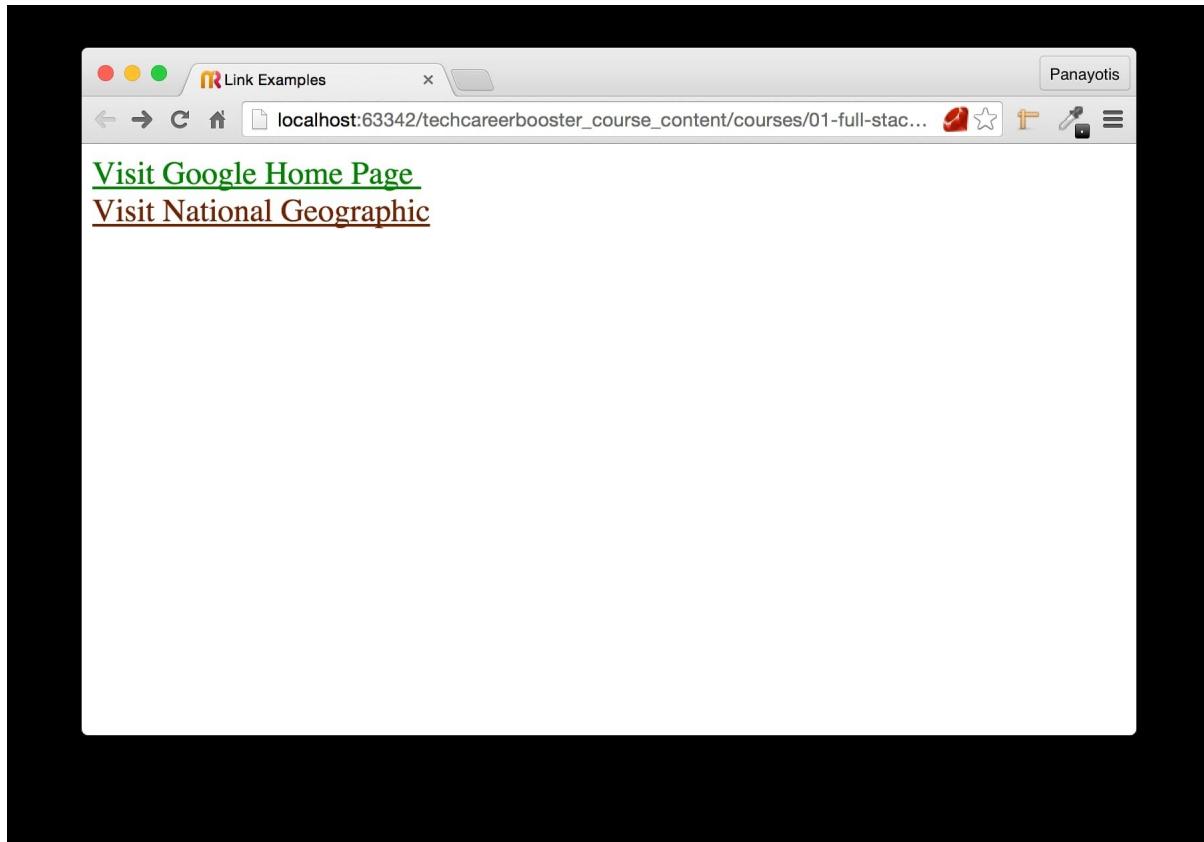
(the above code snippet online)

and here is the new HTML source code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Link Examples</title>
6          <link rel="stylesheet" href="stylesheets/content.css" type="text/css">
7      </head>
8
9      <body>
10
11         <a href="http://www.google.com">
12             Visit Google Home Page
13         </a>
14
15         <br/>
16
17         <a href="http://www.nationalgeographic.com">
18             Visit National Geographic
19         </a>
20
21     </body>
22
23 </html>
```

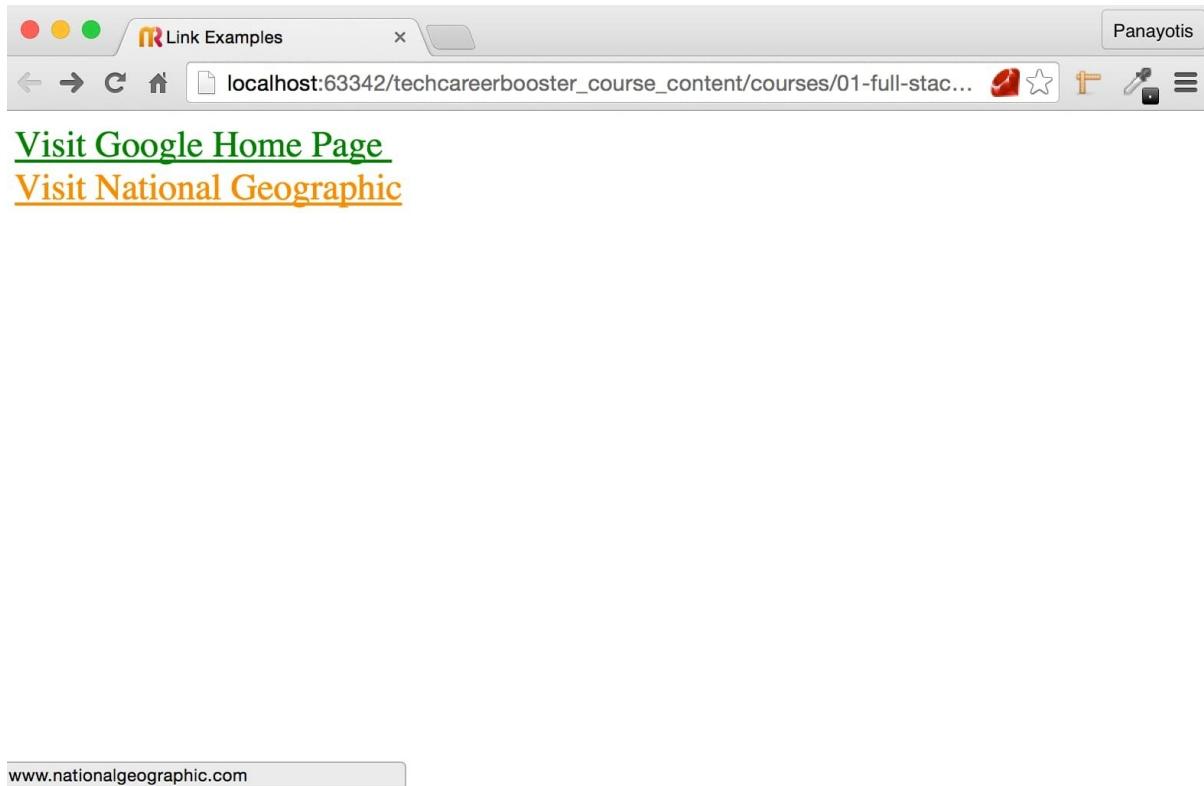
(the above code snippet online)

Save both files and reload the page. The result will be the following:



Visited and Unvisited Links Color

And if you try to visit one link and hold the mouse on top of the link without releasing, you will turn the link to active, with color darkorange. See this here:



Sometimes, we want to change the color of the link when the mouse pointer is on top of the link but without actually clicking on the link. This is the *hover* state. We can change the style of the link in this particular case, with the selector `a:hover`.

Let's change the style of the link while we hover with the mouse over it, as follows:

1. color to be white.
2. background color to be maroon.

In order to do that, the CSS file should be:

```
1 body {  
2     font-size: 24px;  
3 }  
4  
5 a:link {  
6     color: #702309;  
7 }  
8  
9 a:visited {  
10    color: green;  
11 }  
12  
13 a:active {
```

```

14     color: darkorange;
15 }
16
17 a:hover {
18     color: white;
19     background-color: maroon;
20 }

```

(the above code snippet online)

Do you see the last rule with `a:hover`? This is it. The CSS selector that is used to specify the hover state of a link.

If you reload your page with the CSS styling rules, and then move the mouse over one of the links, you will see the following:



Visit Google Home Page
Visit National Geographic

Link with style rule on hover state

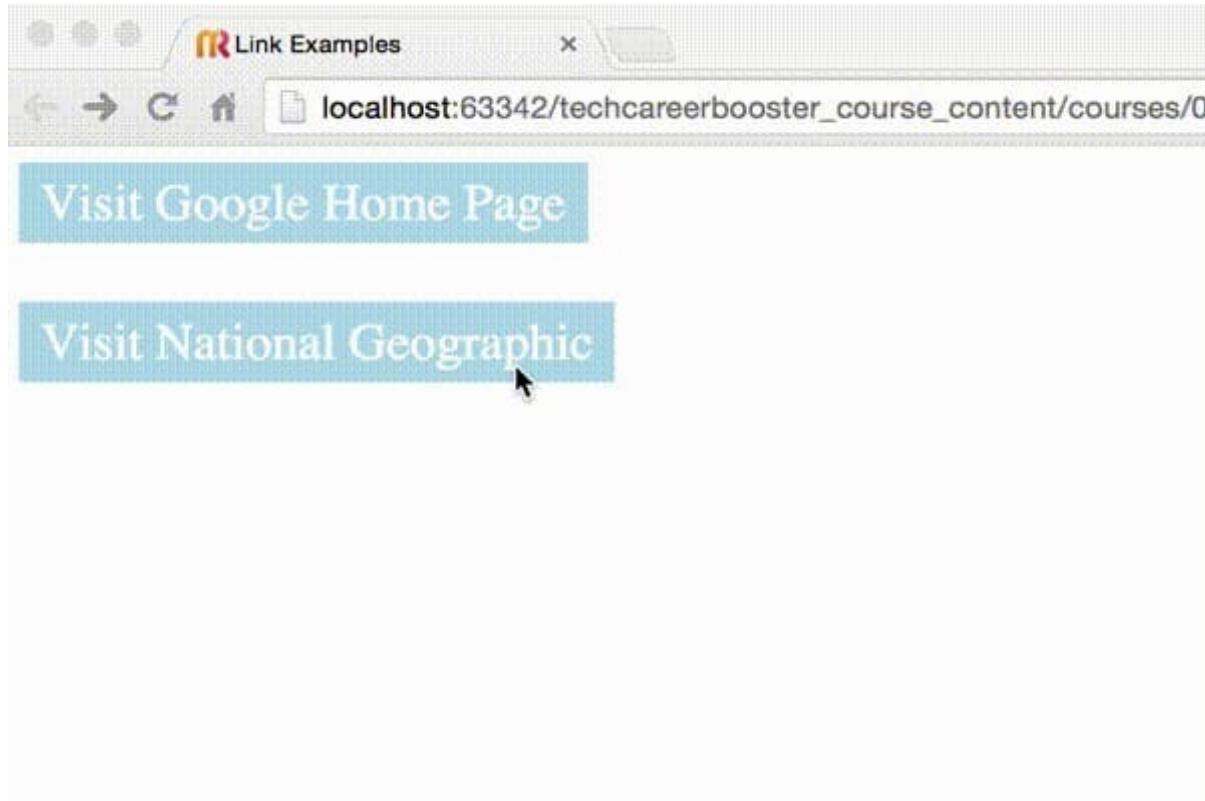
In order to summarize the anchor styling, this table here gives an overview of the CSS selectors that one can use in order to change default colors of the links:

| State | Selector |
|------------------|------------------------|
| Unvisited | <code>a:link</code> |
| Visited | <code>a:visited</code> |
| Active | <code>a:active</code> |
| Mouse Over/Hover | <code>a:hover</code> |

Important: If you want to style links irrespective of their state you select them with the tag name selector `a`. And this is the difference between selecting links with `a` vs `a:link`. The `a:link` is used to style unvisited links. Whereas `a` selects all links irrespective of their state.

Style links as buttons

How can we style links like buttons? For example, how can we achieve the following effect?



Links like buttons

If we carefully see the above picture we can identify the following:

1. The links to have a background color. Let's say that it is: `lightblue: `
2. The links to have text color `white`
3. There is some free space around the content/text of the links. This is the padding. And as we see, the top and bottom padding is a little bit smaller than the left and right padding. Let's assume `5px` for the top and bottom and `10px` for the left and right padding.
4. The unvisited and visited links still have text color being `white`.
5. The active links have a dark background color: `darkblue: `
6. When the mouse goes over a link, the background color is changing to another shade of blue, not very dark. This can be `#4682B4: `
7. The text inside the links is not underlined.

In order to satisfy all these rules we are using the following CSS file:

```

1 body {
2     font-size: 24px;
3 }
4
5 /* Style for all the links irrespective of their state */
6 a {
7     background-color: Lightblue;
8     color: white;
9     padding: 5px 10px; /* this will create the space around the link text and edges of the link */
10}
11}
12
13 a:link,
14 a:visited {
15     color: white;
16     text-decoration: none; /* this will remove any decoration from the link text, like the underline style */
17 }
18}
19
20 a:active {
21     background-color: darkblue;
22 }
23
24 a:hover {
25     background-color: #4682B4;
26 }

```

(the above code snippet online)

Save the above CSS file and reload your page. Do you see the links styled as required? Try to move your mouse over a link. Does it change background color? Try to click on a link and hold your button down and move your mouse a little bit outside the area of the link without releasing. Does it change background color to darkblue?

Side Note - CSS comments

We would like to draw your attention on CSS comments. The previous piece of CSS code, above, uses comments to explain the content of the CSS rules. Example:

```
1 /* Style for all the links irrespective of their state */
```

(the above code snippet online)

The comments are not parsed by the browser as CSS rules. They are completely ignored. But they are useful to the reader and to the author of the rules. They function like documentation of the code.

CSS comments start with /* and finish with */. Whatever you put in between will be considered non-CSS rule content and will be ignored. (like the <!-- and --> delimiters for HTML comments).

Links within the same page - Bookmarks

We have learned how to create a link to another Web page. But, what if we want to link to a position of the same document.

Let's suppose that we have the following HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Links Within the Same Page</title>
6      </head>
7
8      <body>
9
10         <div id="introduction">
11             <h2>Introduction</h2>
12
13             <p>
14                 Perovskia atriplicifolia (/pərvskiə ətriplɪsɪfɒliə/), commonly called \
15                 Russian sage, is a flowering herbaceous perennial plant
16                 and subshrub. Although not a member of Salvia, the genus of other plants \
17                 commonly called sage, it is closely related to them.
18
19                 It has an upright habit, typically reaching 0.5-1.2 m (1 ft 8 in–3 ft 11 \
20                 in) tall, with square stems and grey-green leaves that
21                 yield a distinctive odor when crushed. It is best known for its flowers. \
22                 Its flowering season extends from mid-summer to late October,
23                 with blue to violet blossoms arranged into showy, branched panicles.
24             </p>
25
26             <p>
27                 Native to the steppes and hills of southwestern and central Asia, it was \
28                 introduced to cultivation by Vasily
29                     Perovsky in the 19th century. Successful over a wide range of climate and \
30                     soil conditions, it has since become popular
31                     and widely planted. Several cultivars have been developed, differing prim \
32                     arily in leaf shape and overall height;
33
34                     'Blue Spire' is the most common. This variation has been widely used in g \
35                     ardens and landscaping. P. atriplicifolia was
36                     the Perennial Plant Association's 1995 Plant of the Year, and the 'Blue S \
37                     pire' cultivar received the Award of Garden Merit
38                     from the Royal Horticultural Society.
39             </p>
40             <p>
```

41 The species has a long history of use in traditional medicine in its native range, where it is employed as a treatment
42 for a variety of ailments. This has led to the investigation of its phytochemistry. Its flowers can be eaten in salads or
43 crushed for dyemaking, and the plant has been considered for potential use in the phytoremediation of contaminated soil.
44
45 </p>
46
47 </div>
48
49 <div id="taxonomy">
50 <h2>Taxonomy</h2>
51
52 <p>
53 Perovskia atriplicifolia was described by George Bentham in 1848, based on a specimen collected by William Griffith in Afghanistan,[1]
54 now preserved at the Kew Gardens herbarium as the species's holotype.[2] The specific epithet *atriplicifolia* means "with leaves like *Atriplex*",
55 [3] referring to its similarity to saltbush.[4] Commonly known as Russian sage, *P. atriplicifolia* is neither native to Russia nor a member of
56 *Salvia*,[5] the genus generally referred to as sage.[6]
57 </p>
58
59 <p>
60 A Chinese population was described as a separate species in 1987 and given the name *Perovskia pamirica*,[7] but has since been considered
61 synonymous with *P. atriplicifolia*.
62 </p>
63
64 <p>
65 Within the family Lamiaceae, the large genus *Salvia* had long been believed monophyletic, based on the structure of its stamina.
66 Several smaller genera, including *Dorystaechas*, *Perovskia*, and *Merianandra* were also included in tribe Mentheae, but were thought
67 more distantly related. In 2004, a molecular phylogenetics study based on two cpDNA genes (*rbcL* and *trnL-F*) demonstrated that
68 *Salvia* is not monophyletic, but comprises three identifiable clades. Clade I is more closely related to *Perovskia* than to other
69 members of *Salvia*.[11]
70 </p>
71
72 <p>
73 *P. atriplicifolia* has been the subject of subsequent studies seeking to clarify the relationships within Mentheae. Further research
74 combined palynological analysis of pollen grains with *rbcL* sequencing to provide additional support for the relationship between

87 Perovskia and Salvia clade I. It also distinguished between P. atriplicifolia and P. abrotanoides, while confirming their close
88 relationship.[12] A subsequent multigene study (four cpDNA markers and two nrDNA markers) redrew parts of the Mentheae cladogram,
89 making Rosmarinus a sister group to Perovskia.
90
91 </p>
92
93 <p>
94 Several cultivars of P. atriplicifolia have been developed. They are primarily distinguished by the height of mature plants and the depth
95 of the leaf-margin incisions.[13] Many of these cultivars, especially those with deeply incised leaves, may actually be hybrids of P.
96 atriplicifolia and P. abrotanoides.[13][14] In that context, some may be referred to by the hybrid name P. ×hybrida.[14][15]
97
98 </p>
99
100 <p>
101 The most common cultivar,[16] 'Blue Spire', is among those suspected of being a hybrid.[17][18] It was selected from German plantings
102 by the British Notcutts Nurseries, and first exhibited in 1961.[19][20] 'Blue Spire' grows to approximately 1.2 m (3 ft 11 in), and
103 has large, darker blue flowers.[5][17] In 1993, it received the Royal Horticultural Society's Award of Garden Merit.[21]
104
105 </p>
106
107 <p>
108 'Filigran' reaches a height of 1.2 to 1.3 m (3 ft 11 in to 4 ft 3 in); this tall, sturdy cultivar's name is German for filigree,
109 in reference to its lacy, fern-like foliage.[5][19] 'Little Spire' is shorter, with a mature height of only 0.6 m (2 ft 0 in).[16][22]
110 'Longin' is similar in height to 'Blue Spire' but more upright.[5] Allan Armitage established the late-flowering cultivar 'Mystery of
111 Knightshayes' from a plant at Knightshayes Court.[19] Other cultivars include 'Blue Haze', 'Blue Mist', 'Hybrida' (also called 'Superba'),
112 'Lace', 'Lisslit', 'Rocketman', and 'WALPPB'
113
114 </p>
115 </div>
116
117 <div id="description">
118 <h2>Description</h2>
119
120 <p>
121 Perovskia atriplicifolia is a deciduous perennial subshrub with an erect to spreading habit.[13][27] Superficially, it resembles a much
122 larger version of lavender.[28] Multiple branches arise from a shared rootstalk,[8] growing to a height of 0.5–1.2 m (1 ft 8 in–3 ft 11 in),
123

133 [8][22] with occasional specimens reaching 1.5 m (4 ft 11 in).[4] The mat\
134 ure plant may be 0.6–1.2 m (2 ft 0 in–3 ft 11 in) across.[4]
135 The rigid stems are square in cross-section,[4] and are covered by a indu\
136 mentum formed by stellate, or star-shaped, trichomes and oil droplets.[27]
137 Especially during autumn, these hairs give the stems a silvery appearance\
138 .[29]
139 </p>
140
141 <p>
142 The grayish-green leaves are arranged in opposite pairs,[13][30] and atta\
143 ched to the stems by a short petiole.[27] They are generally 3–5 cm
144 (1.2–2.0 in) long and 0.8–2 cm (0.3–0.8 in) wide,[27] although narrower i\
145 n some populations.[8] The overall leaf shape is oblate, a rounded
146 shape longer than it is wide, to lanceolate, shaped like the head of a la\
147 nce.[27] They are pinnatipartite,[8] with a deeply incised
148 leaf margin that may be either wavy or sharp-toothed; even within a singl\
149 e community of *P. atriplicifolia*, there can be considerable
150 variation in the details of leaf shape.[27] Leaves near the top of branch\
151 es may merge into bracts.[27] The foliage is aromatic,
152 especially when crushed,[4] with a fragrance described as sage-like,[5] a\
153 blend of sage and lavender,[16] or like turpentine.[31]
154 </p>
155
156 <p>
157 The flowering season of *P. atriplicifolia* can be as long as June through \
158 October,[27] although populations in some parts of its range,
159 such as China, may bloom in a much more restricted period.[8] The inflore\
160 scence is a showy panicle, 30–38 cm (12–15 in) long,[4] with
161 many branches.[32] Each of these branches is a raceme, with the individua\
162 l flowers arranged in pairs called verticillasters.[8] Each
163 flower's calyx is purple, densely covered in white or purple hairs, and a\
164 bout 4 mm long. The corolla is tube-shaped, formed from a
165 four-lobed upper lip and a slightly shorter lower lip; the blue or violet\
166 blue petals are about 1 cm long.[8][32] The style has been
167 reported in both an exserted—extending beyond the flower's tube—form and \
168 one contained within the flower;[32] all known examples of *P.*
169 *atriplicifolia* in cultivation have exserted styles.[13] Gardening author \
170 Neil Soderstrom describes the appearance of the flowers from a
171 distance as "like a fine haze or fog".[33]
172 </p>
173 <p>
174 Fruits develop about a month after flowering,[8] and consist of dark brow\
175 n oval nutlets, about 2 mm × 1 mm.
176 </p>
177 </div>
178

```
179     <div id="distribution">
180         <h2>Distribution</h2>
181
182         <p>
183             Widely distributed across Asia in its native range, Perovskia atriplicifolia grows in western China,[4] Pakistan, Afghanistan, Iran,[37]
184             Turkey, and parts of eastern Europe.[38] It is found in steppes and on hillsides,[38] and grows at higher elevations in mountainous regions,
185             including the Himalayas.[39] It has been recorded at 10,000 ft (3,000 m) \ of altitude in the Karakoram.[40] In Pakistan's Quetta district,
186             it is often found in association with the grass Chrysopogon aucheri,[41] \ and may serve as an indicator species for soils with low calcium
187             carbonate and chloride availability.[42] The harsh habitats preferred by \
188 P. atriplicifolia are comparable to the sagebrush steppe of North America.[43]
189         </p>
190
191         <p>
192             In parts of its range, such as the Harboi, these steppe ecosystems are employed as rangeland for grazing animals such as sheep and goats, although
193             this forage is generally of poor nutritional quality. P. atriplicifolia can serve as an important source of phosphorus and zinc,[44] despite being
194             high in poorly-digested material such as neutral detergent fiber and lignin
195         </p>
196     </div>
197
198
199     </body>
200
201     </html>
```

(the above code snippet online)

If we load this page on our browser, we will see that it is a long page:

Long Page Scrolling

What if we wanted to build a table of contents at the beginning, that would help the user navigate within the page? Also, we would like to allow the user to move back to the top of the page, while being at the end of each section.

Table Of Contents And Move Back to The Top

This is the top of the page:



Introduction
Taxonomy
Description
Distribution

Introduction

Perovskia atriplicifolia (/pə'rɒvskiə ə'triplɪsɪ'fɔʊliə/), commonly called Russian sage, and subshrub. Although not a member of *Salvia*, the genus of other plants commonly called sage, it has an upright habit, typically reaching 0.5–1.2 m (1 ft 8 in–3 ft 11 in) tall, with square stems and opposite leaves. It has a distinctive odor when crushed. It is best known for its flowers. Its flowering season extends from July to September, with blue to violet blossoms arranged into showy, branched panicles.

Native to the steppes and hills of southwestern and central Asia, it was introduced to Europe in the 18th century. Successful over a wide range of climate and soil conditions, it has since become widely cultivated. Numerous cultivars have been developed, differing primarily in leaf shape and overall height; 'Blue Spire' and 'Blue Spires' are two well-known cultivars. 'Blue Spire' has been widely used in gardens and landscaping. P. atriplicifolia was the Perennial Plant of the Year in 2009. The 'Blue Spire' cultivar received the Award of Garden Merit from the Royal Horticultural Society in 2004.

[Page With Table Of Contents - Top Of Page](#)

In order to do that, i.e. in order to create links that target points within the same page:

1. the target points need to be HTML elements of your page with an id attribute. This is called a *bookmark*.
2. each link inside the table of contents need to have an href attribute that will target the corresponding HTML element by id. The value of an href attribute that points to another element on the same page starts with the # symbol.

Hint: Do you remember that the CSS selector to select an element by id uses the same # symbol, in order to refer to the id of the element that needs to be selected?

So, the HTML fragment that corresponds to the table of contents on our example page should be:

```
1 <div id="table-of-contents">
2   <a href="#introduction">Introduction</a><br/>
3   <a href="#taxonomy">Taxonomy</a><br/>
4   <a href="#description">Description</a><br/>
5   <a href="#distribution">Distribution</a>
6 </div>
```

(the above code snippet online)

Do you see the href values of the links above? They have a reference that starts with the # symbol. The reference is the id of the corresponding HTML section.

Also, if we want to return back to the top, we can introduce, at the bottom of each section, links like the following:

```
1 <a href="#table-of-contents">Back to Top</a>
```

(the above code snippet online)

This link, when clicked, will jump the reading position back at the top, where the table of contents HTML element is defined.

The final HTML page code is this one:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Links Within the Same Page</title>
6   </head>
7
8   <body>
9     <div id="table-of-contents">
10       <a href="#introduction">Introduction</a><br/>
11       <a href="#taxonomy">Taxonomy</a><br/>
12       <a href="#description">Description</a><br/>
13       <a href="#distribution">Distribution</a>
14     </div>
15
16     <div id="introduction">
17
18       <h2>Introduction</h2>
19
20       <p>
21         Perovskia atriplicifolia (/pərəvskiə ətriplisəfōliə/), commonly called\
22         Russian sage, is a flowering herbaceous perennial plant
23         and subshrub. Although not a member of Salvia, the genus of other plants \
24         commonly called sage, it is closely related to them.
25         It has an upright habit, typically reaching 0.5-1.2 m (1 ft 8 in–3 ft 11 \
26         in) tall, with square stems and grey-green leaves that
27         yield a distinctive odor when crushed. It is best known for its flowers. \
28         Its flowering season extends from mid-summer to late October,
29         with blue to violet blossoms arranged into showy, branched panicles.
30       </p>
31
32       <p>
33         Native to the steppes and hills of southwestern and central Asia, it was \
34         introduced to cultivation by Vasily
35         Perovsky in the 19th century. Successful over a wide range of climate and\
```

```
36   soil conditions, it has since become popular
37       and widely planted. Several cultivars have been developed, differing prim\ 
38 arily in leaf shape and overall height;
39       'Blue Spire' is the most common. This variation has been widely used in g\ 
40 ardens and landscaping. P. atriplicifolia was
41       the Perennial Plant Association's 1995 Plant of the Year, and the 'Blue S\ 
42 pire' cultivar received the Award of Garden Merit
43       from the Royal Horticultural Society.
44   </p>
45
46   <p>
47       The species has a long history of use in traditional medicine in its nati\ 
48 ve range, where it is employed as a treatment
49       for a variety of ailments. This has led to the investigation of its phyto\ 
50 chemistry. Its flowers can be eaten in salads or
51       crushed for dyemaking, and the plant has been considered for potential us\ 
52 e in the phytoremediation of contaminated soil.
53   </p>
54
55   <a href="#table-of-contents">Back To Top</a>
56 </div>
57
58 <div id="taxonomy">
59   <h2>Taxonomy</h2>
60
61   <p>
62       Perovskia atriplicifolia was described by George Bentham in 1848, based on\ 
63 a specimen collected by William Griffith in Afghanistan,[1]
64       now preserved at the Kew Gardens herbarium as the species's holotype.[2] \
65 The specific epithet atriplicifolia means "with leaves like Atriplex",[3] \
66 referring to its similarity to saltbush.[4] Commonly known as Russian\ 
67 sage, P. atriplicifolia is neither native to Russia nor a member of
68       Salvia,[5] the genus generally referred to as sage.[6]
69   </p>
70
71   <p>
72       A Chinese population was described as a separate species in 1987 and given\ 
73 the name Perovskia pamirica,[7] but has since been considered
74       synonymous with P. atriplicifolia.
75   </p>
76
77   <p>
78       Within the family Lamiaceae, the large genus Salvia had long been believe\ 
79 d monophyletic, based on the structure of its stamens.
80       Several smaller genera, including Dorystaechas, Perovskia, and Merianandra \
81 were also included in tribe Mentheae, but were thought
```

82 more distantly related. In 2004, a molecular phylogenetics study based on\\
83 two cpDNA genes (*rbcL* and *trnL-F*) demonstrated that
84 *Salvia* is not monophyletic, but comprises three identifiable clades. Clad\\
85 e I is more closely related to *Perovskia* than to other
86 members of *Salvia*.[11]
87 </p>
88
89 <p>
90 *P. atriplicifolia* has been the subject of subsequent studies seeking to c\\
91 larify the relationships within *Mentheae*. Further research
92 combined palynological analysis of pollen grains with *rbcL* sequencing to \\
93 provide additional support for the relationship between
94 *Perovskia* and *Salvia* clade I. It also distinguished between *P. atriplicif*
95 *olia* and *P. abrotanoides*, while confirming their close
96 relationship.[12] A subsequent multigene study (four cpDNA markers and tw\\
97 o nrDNA markers) redrew parts of the *Mentheae* cladogram,
98 making *Rosmarinus* a sister group to *Perovskia*.
99 </p>
100
101 <p>
102 Several cultivars of *P. atriplicifolia* have been developed. They are prim\\
103 arily distinguished by the height of mature plants and the depth
104 of the leaf-margin incisions.[13] Many of these cultivars, especially tho\\
105 se with deeply incised leaves, may actually be hybrids of *P.*
106 *atriplicifolia* and *P. abrotanoides*.[13][14] In that context, some may be \\
107 referred to by the hybrid name *P. ×hybrida*.[14][15]
108 </p>
109
110 <p>
111 The most common cultivar,[16] 'Blue Spire', is among those suspected of b\\
112 eing a hybrid.[17][18] It was selected from German plantings
113 by the British Notcutts Nurseries, and first exhibited in 1961.[19][20] '\
114 Blue Spire' grows to approximately 1.2 m (3 ft 11 in), and
115 has large, darker blue flowers.[5][17] In 1993, it received the Royal Hor\\
116 ticultural Society's Award of Garden Merit.[21]
117 </p>
118
119 <p>
120 'Filigran' reaches a height of 1.2 to 1.3 m (3 ft 11 in to 4 ft 3 in); th\\
121 is tall, sturdy cultivar's name is German for filigree,
122 in reference to its lacy, fern-like foliage.[5][19] 'Little Spire' is sho\\
123 rter, with a mature height of only 0.6 m (2 ft 0 in).[16][22]
124 'Longin' is similar in height to 'Blue Spire' but more upright.[5] Allan \\
125 Armitage established the late-flowering cultivar 'Mystery of
126 Knightshayes' from a plant at Knightshayes Court.[19] Other cultivars inc\\
127 lude 'Blue Haze', 'Blue Mist', 'Hybrida' (also called 'Superba'),

```
128     'Lace', 'Lisslit', 'Rocketman', and 'WALPPB'  
129     </p>  
130  
131     <a href="#table-of-contents">Back To Top</a>  
132     </div>  
133  
134     <div id="description">  
135         <h2>Description</h2>  
136  
137         <p>  
138             Perovskia atriplicifolia is a deciduous perennial subshrub with an erect \  
139             to spreading habit.[13][27] Superficially, it resembles a much  
140                 larger version of lavender.[28] Multiple branches arise from a shared roo\  
141             tstalk,[8] growing to a height of 0.5–1.2 m (1 ft 8 in–3 ft 11 in),  
142                 [8][22] with occasional specimens reaching 1.5 m (4 ft 11 in).[4] The mat\  
143             ure plant may be 0.6–1.2 m (2 ft 0 in–3 ft 11 in) across.[4]  
144                 The rigid stems are square in cross-section,[4] and are covered by a indu\  
145             mentum formed by stellate, or star-shaped, trichomes and oil droplets.[27]  
146                 Especially during autumn, these hairs give the stems a silvery appearance\  
147             .[29]  
148         </p>  
149  
150         <p>  
151             The grayish-green leaves are arranged in opposite pairs,[13][30] and atta\  
152             ched to the stems by a short petiole.[27] They are generally 3–5 cm  
153                 (1.2–2.0 in) long and 0.8–2 cm (0.3–0.8 in) wide,[27] although narrower i\  
154             n some populations.[8] The overall leaf shape is oblate, a rounded  
155                 shape longer than it is wide, to lanceolate, shaped like the head of a la\  
156             nce.[27] They are pinnatipartite,[8] with a deeply incised  
157                 leaf margin that may be either wavy or sharp-toothed; even within a singl\  
158             e community of P. atriplicifolia, there can be considerable  
159                 variation in the details of leaf shape.[27] Leaves near the top of branch\  
160             es may merge into bracts.[27] The foliage is aromatic,  
161                 especially when crushed,[4] with a fragrance described as sage-like,[5] a\  
162             blend of sage and lavender,[16] or like turpentine.[31]  
163         </p>  
164  
165         <p>  
166             The flowering season of P. atriplicifolia can be as long as June through \  
167             October,[27] although populations in some parts of its range,  
168                 such as China, may bloom in a much more restricted period.[8] The inflore\  
169             scence is a showy panicle, 30–38 cm (12–15 in) long,[4] with  
170                 many branches.[32] Each of these branches is a raceme, with the individua\  
171             l flowers arranged in pairs called verticillasters.[8] Each  
172                 flower's calyx is purple, densely covered in white or purple hairs, and a\  
173             bout 4 mm long. The corolla is tube-shaped, formed from a
```

```
174     four-lobed upper lip and a slightly shorter lower lip; the blue or violet\  
175     blue petals are about 1 cm long.[8][32] The style has been  
176     reported in both an exserted-extending beyond the flower's tube-form and \  
177     one contained within the flower;[32] all known examples of P.  
178     atriplicifolia in cultivation have exserted styles.[13] Gardening author \  
179     Neil Soderstrom describes the appearance of the flowers from a  
180     distance as "like a fine haze or fog".[33]  
181     </p>  
182     <p>  
183         Fruits develop about a month after flowering,[8] and consist of dark brow\  
184     n oval nutlets, about 2 mm × 1 mm.  
185     </p>  
186  
187     <a href="#table-of-contents">Back To Top</a>  
188     </div>  
189  
190     <div id="distribution">  
191         <h2>Distribution</h2>  
192  
193         <p>  
194             Widely distributed across Asia in its native range, Perovskia atriplicifo\  
195     lia grows in western China,[4] Pakistan, Afghanistan, Iran,[37]  
196             Turkey, and parts of eastern Europe.[38] It is found in steppes and on hi\  
197     llsides,[38] and grows at higher elevations in mountainous regions,  
198             including the Himalayas.[39] It has been recorded at 10,000 ft (3,000 m) \  
199     of altitude in the Karakoram.[40] In Pakistan's Quetta district,  
200             it is often found in association with the grass Chrysopogon aucheri,[41] \  
201     and may serve as an indicator species for soils with low calcium  
202             carbonate and chloride availability.[42] The harsh habitats preferred by \  
203     P. atriplicifolia are comparable to the sagebrush steppe of North America.[43]  
204     </p>  
205  
206     <p>  
207         In parts of its range, such as the Harboi, these steppe ecosystems are em\  
208     ployed as rangeland for grazing animals such as sheep and goats, although  
209             this forage is generally of poor nutritional quality. P. atriplicifolia c\  
210     an serve as an important source of phosphorus and zinc,[44] despite being  
211             high in poorly-digested material such as neutral detergent fiber and lign\  
212     in  
213     </p>  
214  
215     <a href="#table-of-contents">Back To Top</a>  
216     </div>  
217  
218  
219     </body>
```

```
220  
221  </html>
```

(the above code snippet online)

Links to Other Pages within Same Site

So far, we have seen

1. Links to external pages, i.e. pages that exist on other sites. Like the `Visit Google Site`.
2. Links within the same page, like `Introduction`.

We can have links within the same site but different page. Let's see an example in order to explain that. Note that this example is going to have 4 pages, a.k.a. our site is going to have 4 pages, 4 HTML files:

1. main.html
2. taxonomy.html
3. description.html
4. distribution.html

The `main.html` file will have the *Introduction* section.

The `taxonomy.html` file will have the *Taxonomy* section.

The `description.html` file will have the *Description* section.

The `distribution.html` file will have the *Distribution* section.

Not only that, we will organize the last 3 pages (taxonomy, description and distribution) into their own folder, named "chapters".

Let's start with the main page, the content of which can be the following:

```
1  <!DOCTYPE html>  
2  <html>  
3      <head>  
4          <meta charset="utf-8">  
5          <title>Site With Many Pages</title>  
6      </head>  
7  
8      <body>  
9  
10         <div id="introduction">  
11             <h2>Introduction</h2>  
12         </div>
```

```

14      <p>
15          Perovskia atriplicifolia (/pərəvskiə ətrɪplɪsɪfɔliə/), commonly called \
16  Russian sage, is a flowering herbaceous perennial plant
17      and subshrub. Although not a member of Salvia, the genus of other plants \
18  commonly called sage, it is closely related to them.
19      It has an upright habit, typically reaching 0.5–1.2 m (1 ft 8 in–3 ft 11 \
20  in) tall, with square stems and grey-green leaves that
21      yield a distinctive odor when crushed. It is best known for its flowers. \
22  Its flowering season extends from mid-summer to late October,
23      with blue to violet blossoms arranged into showy, branched panicles.
24  </p>
25
26  </div>
27
28  </body>
29 </html>

```

[\(the above code snippet online\)](#)

Note that we have not yet created the part that will allow us to go to other pages. Save this file with the name “main.html”.

Now. Go ahead and create the first chapter page, the “taxonomy.html”. Save the following content inside the folder “chapters/taxonomy.html”.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Taxonomy Chapter</title>
6      </head>
7
8      <body>
9
10     <div id="taxonomy">
11         <h2>Taxonomy</h2>
12
13         <p>
14             Perovskia atriplicifolia was described by George Bentham in 1848, based o\
15 n a specimen collected by William Griffith in Afghanistan,[1]
16             now preserved at the Kew Gardens herbarium as the species's holotype.[2] \
17 The specific epithet atriplicifolia means "with leaves like Atriplex",
18             [3] referring to its similarity to saltbush.[4] Commonly known as Russian\
19 sage, P. atriplicifolia is neither native to Russia nor a member of
20             Salvia,[5] the genus generally referred to as sage.[6]
21         </p>
22

```

```
23      <p>
24          A Chinese population was described as a separate species in 1987 and give\
25 n the name Perovskia pamirica,[7] but has since been considered
26              synonymous with P. atriplicifolia.
27      </p>
28
29      <p>
30          Within the family Lamiaceae, the large genus Salvia had long been believe\
31 d monophyletic, based on the structure of its stamina.
32          Several smaller genera, including Dorystaechas, Perovskia, and Meriandra \
33 were also included in tribe Mentheae, but were thought
34              more distantly related. In 2004, a molecular phylogenetics study based on\
35 two cpDNA genes (rbcL and trnL-F) demonstrated that
36          Salvia is not monophyletic, but comprises three identifiable clades. Clad\
37 e I is more closely related to Perovskia than to other
38              members of Salvia.[11]
39      </p>
40
41      <p>
42          P. atriplicifolia has been the subject of subsequent studies seeking to c\
43 larify the relationships within Mentheae. Further research
44              combined palynological analysis of pollen grains with rbcL sequencing to \
45 provide additional support for the relationship between
46          Perovskia and Salvia clade I. It also distinguished between P. atriplicif\
47 olia and P. abrotanoides, while confirming their close
48              relationship.[12] A subsequent multigene study (four cpDNA markers and tw\
49 o nrDNA markers) redrew parts of the Mentheae cladogram,
50              making Rosmarinus a sister group to Perovskia.
51      </p>
52
53      <p>
54          Several cultivars of P. atriplicifolia have been developed. They are prim\
55 arily distinguished by the height of mature plants and the depth
56              of the leaf-margin incisions.[13] Many of these cultivars, especially tho\
57 se with deeply incised leaves, may actually be hybrids of P.
58          atriplicifolia and P. abrotanoides.[13][14] In that context, some may be \
59 referred to by the hybrid name P. ×hybrida.[14][15]
60      </p>
61
62      <p>
63          The most common cultivar,[16] 'Blue Spire', is among those suspected of b\
64 eing a hybrid.[17][18] It was selected from German plantings
65              by the British Notcutts Nurseries, and first exhibited in 1961.[19][20] '\
66 Blue Spire' grows to approximately 1.2 m (3 ft 11 in), and
67              has large, darker blue flowers.[5][17] In 1993, it received the Royal Hor\
68 ticultural Society's Award of Garden Merit.[21]
```

```

69      </p>
70
71      <p>
72          'Filigran' reaches a height of 1.2 to 1.3 m (3 ft 11 in to 4 ft 3 in); th\
73  is tall, sturdy cultivar's name is German for filigree,
74      in reference to its lacy, fern-like foliage.[5][19] 'Little Spire' is sho\
75  rter, with a mature height of only 0.6 m (2 ft 0 in).[16][22]
76      'Longin' is similar in height to 'Blue Spire' but more upright.[5] Allan \
77  Armitage established the late-flowering cultivar 'Mystery of
78      Knightshayes' from a plant at Knightshayes Court.[19] Other cultivars inc\
79  lude 'Blue Haze', 'Blue Mist', 'Hybrida' (also called 'Superba'),
80      'Lace', 'Lisslit', 'Rocketman', and 'WALPPB'
81      </p>
82
83      </div>
84
85  </body>
86
87 </html>

```

(the above code snippet online)

Note that we have not put, yet, any links to navigate to other pages.

Similar to the taxonomy chapter page, create the following page and save it inside the file chapters/description.html

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Description Chapter</title>
6      </head>
7
8      <body>
9
10     <div id="description">
11         <h2>Description</h2>
12
13         <p>
14             Perovskia atriplicifolia is a deciduous perennial subshrub with an erect \
15  to spreading habit.[13][27] Superficially, it resembles a much
16             larger version of lavender.[28] Multiple branches arise from a shared roo\
17  tstalk,[8] growing to a height of 0.5–1.2 m (1 ft 8 in–3 ft 11 in),
18             [8][22] with occasional specimens reaching 1.5 m (4 ft 11 in).[4] The mat\
19  ure plant may be 0.6–1.2 m (2 ft 0 in–3 ft 11 in) across.[4]
20             The rigid stems are square in cross-section,[4] and are covered by a indu\

```

```
21 mentum formed by stellate, or star-shaped, trichomes and oil droplets.[27]
22 Especially during autumn, these hairs give the stems a silvery appearance\
23 .[29]
24     </p>
25
26     <p>
27         The grayish-green leaves are arranged in opposite pairs,[13][30] and atta\
28 ched to the stems by a short petiole.[27] They are generally 3–5 cm
29         (1.2–2.0 in) long and 0.8–2 cm (0.3–0.8 in) wide,[27] although narrower i\
30 n some populations.[8] The overall leaf shape is oblate, a rounded
31         shape longer than it is wide, to lanceolate, shaped like the head of a la\
32 nce.[27] They are pinnatipartite,[8] with a deeply incised
33         leaf margin that may be either wavy or sharp-toothed; even within a singl\
34 e community of P. atriplicifolia, there can be considerable
35         variation in the details of leaf shape.[27] Leaves near the top of branch\
36 es may merge into bracts.[27] The foliage is aromatic,
37         especially when crushed,[4] with a fragrance described as sage-like,[5] a\
38 blend of sage and lavender,[16] or like turpentine.[31]
39     </p>
40
41     <p>
42         The flowering season of P. atriplicifolia can be as long as June through \
43 October,[27] although populations in some parts of its range,
44         such as China, may bloom in a much more restricted period.[8] The inflore\
45 scence is a showy panicle, 30–38 cm (12–15 in) long,[4] with
46         many branches.[32] Each of these branches is a raceme, with the individua\
47 l flowers arranged in pairs called verticillasters.[8] Each
48         flower's calyx is purple, densely covered in white or purple hairs, and a\
49 bout 4 mm long. The corolla is tube-shaped, formed from a
50         four-lobed upper lip and a slightly shorter lower lip; the blue or violet\
51 blue petals are about 1 cm long.[8][32] The style has been
52         reported in both an exserted-extending beyond the flower's tube-form and \
53 one contained within the flower;[32] all known examples of P.
54         atriplicifolia in cultivation have exserted styles.[13] Gardening author \
55 Neil Soderstrom describes the appearance of the flowers from a
56         distance as "like a fine haze or fog".[33]
57     </p>
58     <p>
59         Fruits develop about a month after flowering,[8] and consist of dark brow\
60 n oval nutlets, about 2 mm × 1 mm.
61     </p>
62
63     </div>
64
65     </body>
66
```

67 </html>

(the above code snippet online)

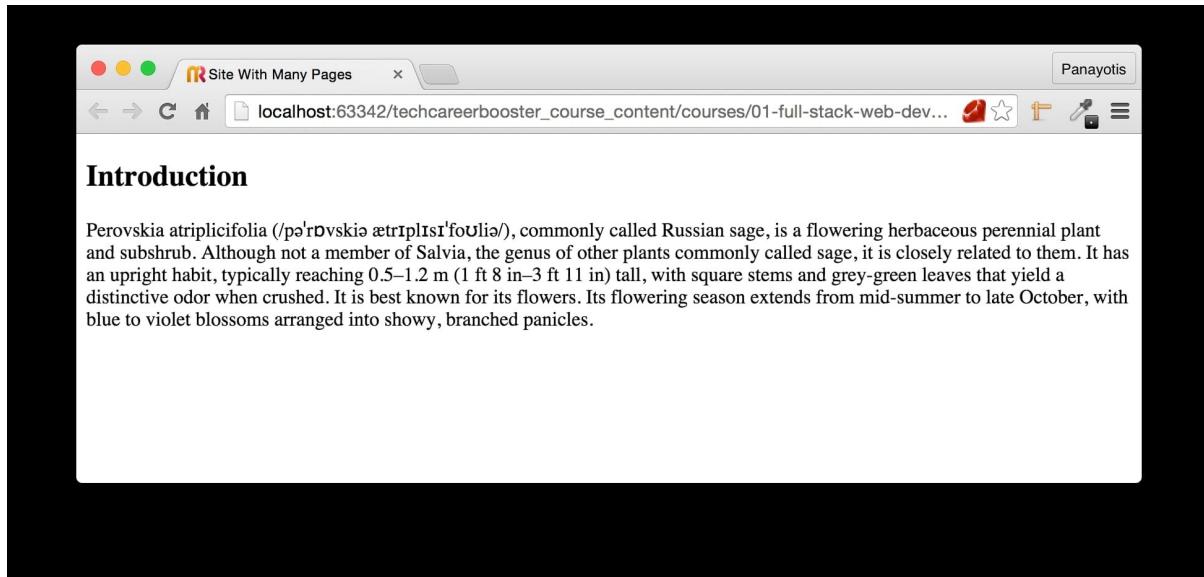
Like before, we have not put yet any links to navigate to other pages.

Finally, we create the file chapters/distribution.html with the following content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Distribution Chapter</title>
6      </head>
7
8      <body>
9
10         <div id="distribution">
11             <h2>Distribution</h2>
12
13             <p>
14                 Widely distributed across Asia in its native range, Perovskia atriplicifolia grows in western China,[4] Pakistan, Afghanistan, Iran,[37]
15                 Turkey, and parts of eastern Europe.[38] It is found in steppes and on hillsides,[38] and grows at higher elevations in mountainous regions,
16                 including the Himalayas.[39] It has been recorded at 10,000 ft (3,000 m) of altitude in the Karakoram.[40] In Pakistan's Quetta district,
17                 it is often found in association with the grass Chrysopogon aucheri,[41] and may serve as an indicator species for soils with low calcium
18                 carbonate and chloride availability.[42] The harsh habitats preferred by P. atriplicifolia are comparable to the sagebrush steppe of North America.[43]
19             </p>
20
21             <p>
22                 In parts of its range, such as the Harboi, these steppe ecosystems are employed as rangeland for grazing animals such as sheep and goats, although
23                 this forage is generally of poor nutritional quality. P. atriplicifolia can serve as an important source of phosphorus and zinc,[44] despite being
24                 high in poorly-digested material such as neutral detergent fiber and lignin
25             </p>
26         </div>
27
28     </body>
29
30 </html>
```

(the above code snippet online)

You can open each one of the above files using your browser. You will see their content but there is not going to be any connection between the pages:



Site With Many Pages - Main, Introduction Page

The screenshot shows a web browser window with the title "Taxonomy Chapter". The URL in the address bar is "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-dev...". The page content is as follows:

Taxonomy

Perovskia atriplicifolia was described by George Bentham in 1848, based on a specimen collected by William Griffith in Afghanistan.^[1] now preserved at the Kew Gardens herbarium as the species's holotype.^[2] The specific epithet *atriplicifolia* means "with leaves like *Atriplex*",^[3] referring to its similarity to saltbush.^[4] Commonly known as Russian sage, *P. atriplicifolia* is neither native to Russia nor a member of *Salvia*,^[5] the genus generally referred to as sage.^[6]

A Chinese population was described as a separate species in 1987 and given the name *Perovskia pamirica*,^[7] but has since been considered synonymous with *P. atriplicifolia*.

Within the family Lamiaceae, the large genus *Salvia* had long been believed monophyletic, based on the structure of its stamens. Several smaller genera, including *Dorystaechas*, *Perovskia*, and *Merienda* were also included in tribe *Mentheae*, but were thought more distantly related. In 2004, a molecular phylogenetics study based on two cpDNA genes (*rbcL* and *trnL-F*) demonstrated that *Salvia* is not monophyletic, but comprises three identifiable clades. Clade I is more closely related to *Perovskia* than to other members of *Salvia*.^[11]

P. atriplicifolia has been the subject of subsequent studies seeking to clarify the relationships within *Mentheae*. Further research combined palynological analysis of pollen grains with *rbcL* sequencing to provide additional support for the relationship between *Perovskia* and *Salvia* clade I. It also distinguished between *P. atriplicifolia* and *P. abrotanoides*, while confirming their close relationship.^[12] A subsequent multigene study (four cpDNA markers and two nrDNA markers) redrew parts of the *Mentheae* cladogram, making *Rosmarinus* a sister group to *Perovskia*.

Several cultivars of *P. atriplicifolia* have been developed. They are primarily distinguished by the height of mature plants and the depth of the leaf-margin incisions.^[13] Many of these cultivars, especially those with deeply incised leaves, may actually be hybrids of *P. atriplicifolia* and *P. abrotanoides*.^{[13][14]} In that context, some may be referred to by the hybrid name *P. ×hybrida*.^{[14][15]}

The most common cultivar,^[16] 'Blue Spire', is among those suspected of being a hybrid.^{[17][18]} It was selected from German plantings by the British Notcutts Nurseries, and first exhibited in 1961.^{[19][20]} 'Blue Spire' grows to approximately 1.2 m (3 ft 11 in), and has large, darker blue flowers.^{[5][17]} In 1993, it received the Royal Horticultural Society's Award of Garden Merit.^[21]

'Filigran' reaches a height of 1.2 to 1.3 m (3 ft 11 in to 4 ft 3 in); this tall, sturdy cultivar's name is German for filigree, in reference to its lacy, fern-like foliage.^{[5][19]} 'Little Spire' is shorter, with a mature height of only 0.6 m (2 ft 0 in).^{[16][22]} 'Longin' is similar in height to 'Blue Spire' but more upright.^[5] Allan Armitage established the late-flowering cultivar 'Mystery of Knightshayes' from a plant at Knightshayes Court.^[19] Other cultivars include 'Blue Haze', 'Blue Mist', 'Hybrida' (also called 'Superba'), 'Lace', 'Lisslit', 'Rocketman', and 'WALPPB'.

Site With Many Pages - Taxonomy Page

Description

Perovskia atriplicifolia is a deciduous perennial subshrub with an erect to spreading habit.[13][27] Superficially, it resembles a much larger version of lavender.[28] Multiple branches arise from a shared rootstalk,[8] growing to a height of 0.5–1.2 m (1 ft 8 in–3 ft 11 in), [8][22] with occasional specimens reaching 1.5 m (4 ft 11 in).[4] The mature plant may be 0.6–1.2 m (2 ft 0 in–3 ft 11 in) across.[4] The rigid stems are square in cross-section,[4] and are covered by a indumentum formed by stellate, or star-shaped, trichomes and oil droplets.[27] Especially during autumn, these hairs give the stems a silvery appearance.[29]

The grayish-green leaves are arranged in opposite pairs,[13][30] and attached to the stems by a short petiole.[27] They are generally 3–5 cm (1.2–2.0 in) long and 0.8–2 cm (0.3–0.8 in) wide,[27] although narrower in some populations.[8] The overall leaf shape is oblanceolate, a rounded shape longer than it is wide, to lanceolate, shaped like the head of a lance.[27] They are pinnatifid, [8] with a deeply incised leaf margin that may be either wavy or sharp-toothed; even within a single community of *P. atriplicifolia*, there can be considerable variation in the details of leaf shape.[27] Leaves near the top of branches may merge into bracts.[27] The foliage is aromatic, especially when crushed,[4] with a fragrance described as sage-like,[5] a blend of sage and lavender,[16] or like turpentine.[31]

The flowering season of *P. atriplicifolia* can be as long as June through October,[27] although populations in some parts of its range, such as China, may bloom in a much more restricted period.[8] The inflorescence is a showy panicle, 30–38 cm (12–15 in) long,[4] with many branches.[32] Each of these branches is a raceme, with the individual flowers arranged in pairs called verticillasters.[8] Each flower's calyx is purple, densely covered in white or purple hairs, and about 4 mm long. The corolla is tube-shaped, formed from a four-lobed upper lip and a slightly shorter lower lip; the blue or violet blue petals are about 1 cm long.[8] [32] The style has been reported in both an exserted—extending beyond the flower's tube—form and one contained within the flower;[32] all known examples of *P. atriplicifolia* in cultivation have exserted styles.[13] Gardening author Neil Soderstrom describes the appearance of the flowers from a distance as "like a fine haze or fog".[33]

Fruits develop about a month after flowering,[8] and consist of dark brown oval nutlets, about 2 mm × 1 mm.

Site With Many Pages - Description Page

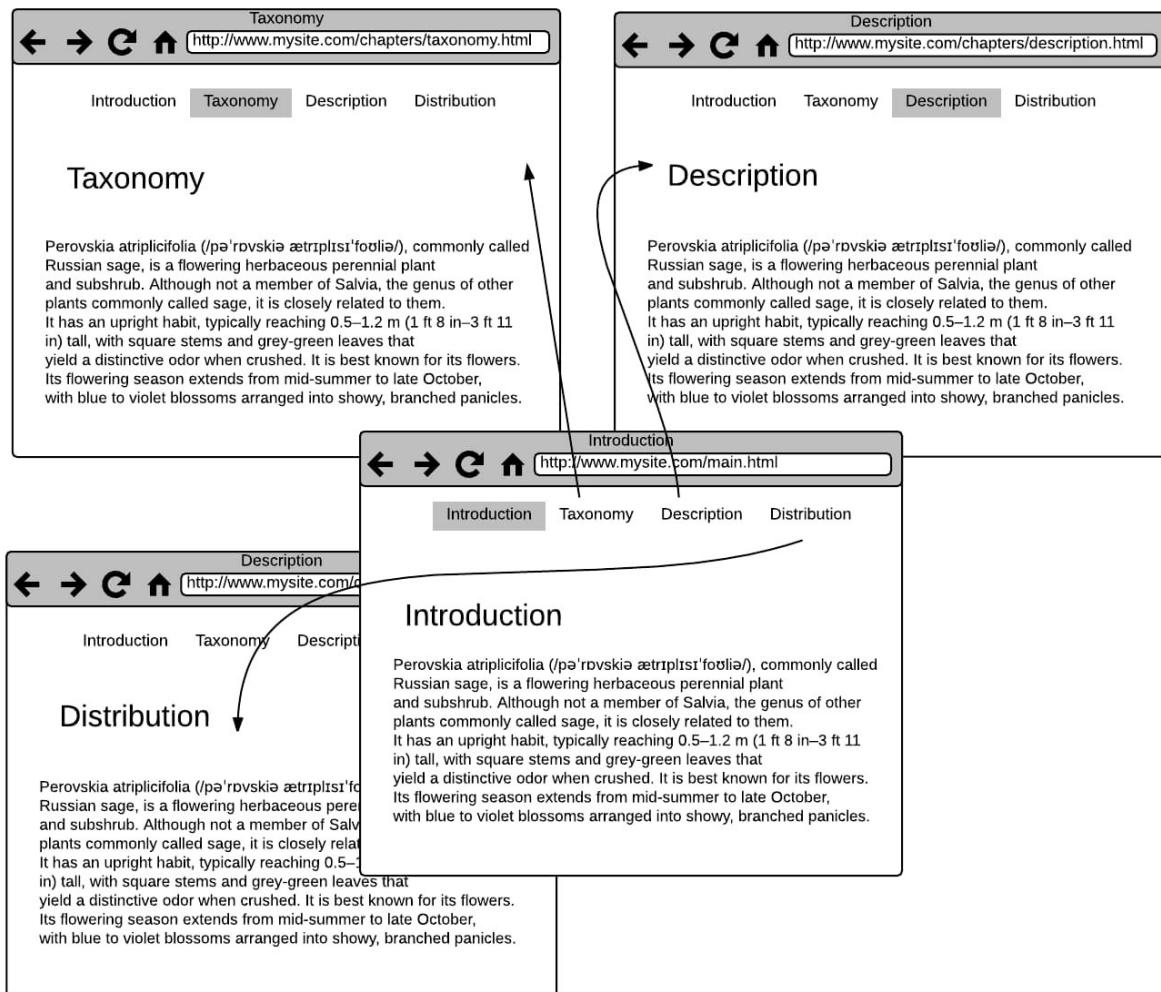
Distribution

Widely distributed across Asia in its native range, *Perovskia atriplicifolia* grows in western China,[4] Pakistan, Afghanistan, Iran, [37] Turkey, and parts of eastern Europe.[38] It is found in steppes and on hillsides,[38] and grows at higher elevations in mountainous regions, including the Himalayas.[39] It has been recorded at 10,000 ft (3,000 m) of altitude in the Karakoram.[40] In Pakistan's Quetta district, it is often found in association with the grass *Chrysopogon aucheri*,[41] and may serve as an indicator species for soils with low calcium carbonate and chloride availability.[42] The harsh habitats preferred by *P. atriplicifolia* are comparable to the sagebrush steppe of North America.[43]

In parts of its range, such as the Harboi, these steppe ecosystems are employed as rangeland for grazing animals such as sheep and goats, although this forage is generally of poor nutritional quality. *P. atriplicifolia* can serve as an important source of phosphorus and zinc,[44] despite being high in poorly-digested material such as neutral detergent fiber and lignin

Site With Many Pages - Distribution Page

This is another view of what we want to achieve:



Site With Many Pages - Site Map

As you can see above, we want a horizontal list of links that would allow us to go from one page to the other. Also, on each page, we need to have the current page link highlighted so that the user knows what page he is in.

Let's add the links on the top of the main, introduction page. The code for the top navigation menu will be:

```

1 <div id="navigation">
2   <a href="main.html">Introduction</a>
3   <a href="chapters/taxonomy.html">Taxonomy</a>
4   <a href="chapters/description.html">Description</a>
5   <a href="chapters/distribution.html">Distribution</a>
6 </div>
```

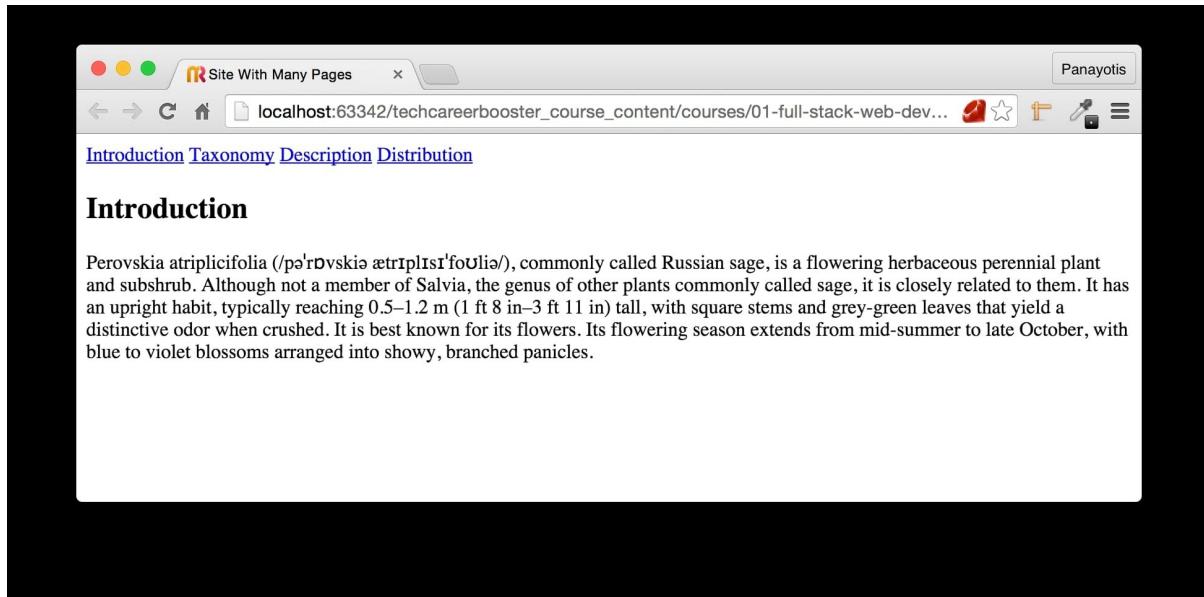
(the above code snippet online)

and the whole `main.html` page will be:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Site With Many Pages</title>
6      </head>
7
8      <body>
9
10         <div id="navigation">
11             <a href="main.html">Introduction</a>
12             <a href="chapters/taxonomy.html">Taxonomy</a>
13             <a href="chapters/description.html">Description</a>
14             <a href="chapters/distribution.html">Distribution</a>
15         </div>
16
17         <div id="introduction">
18             <h2>Introduction</h2>
19
20             <p>
21                 Perovskia atriplicifolia (/pərɒvskiə ətrɪplɪsɪfɒliə/), commonly called\
22                 Russian sage, is a flowering herbaceous perennial plant
23                 and subshrub. Although not a member of Salvia, the genus of other plants \
24                 commonly called sage, it is closely related to them.
25
26                 It has an upright habit, typically reaching 0.5–1.2 m (1 ft 8 in–3 ft 11 \
27                 in) tall, with square stems and grey-green leaves that
28                 yield a distinctive odor when crushed. It is best known for its flowers. \
29                 Its flowering season extends from mid-summer to late October,
30                 with blue to violet blossoms arranged into showy, branched panicles.
31             </p>
32
33         </div>
34
35     </body>
36 </html>
```

(the above code snippet online)

If we save main.html and we reload it, it will look like this:



Site With Many Pages - Main Page with Navigation

You can try clicking the links on “Taxonomy”, “Description” and “Distribution”. You will see that it will take you to the other pages of the site. The whole work is done with the *relative* links such as:

```
1 <a href="chapters/taxonomy.html">Taxonomy</a>
```

(the above code snippet online)

As you can see a *relative* link has an `href` value that does not start from `http://` or `https://`. If neither of them is there, browser assumes that the `href` refers to a page on the same site like the page currently loaded. Note also that, since the target page (`chapters/taxonomy.html`) does not belong to the same folder like the current page (`main.html`) the `href` needs to prefix the target page file name (`taxonomy.html`) with the path from current page, which in our case is `chapters`.

One can now assume, that we can copy / paste the navigation links HTML fragment, and reuse it as is inside the other three pages, the ones that reside inside the `chapters` folder. Let's try that with the taxonomy page. Add the same navigation links code at the top of the taxonomy page, as in the following:

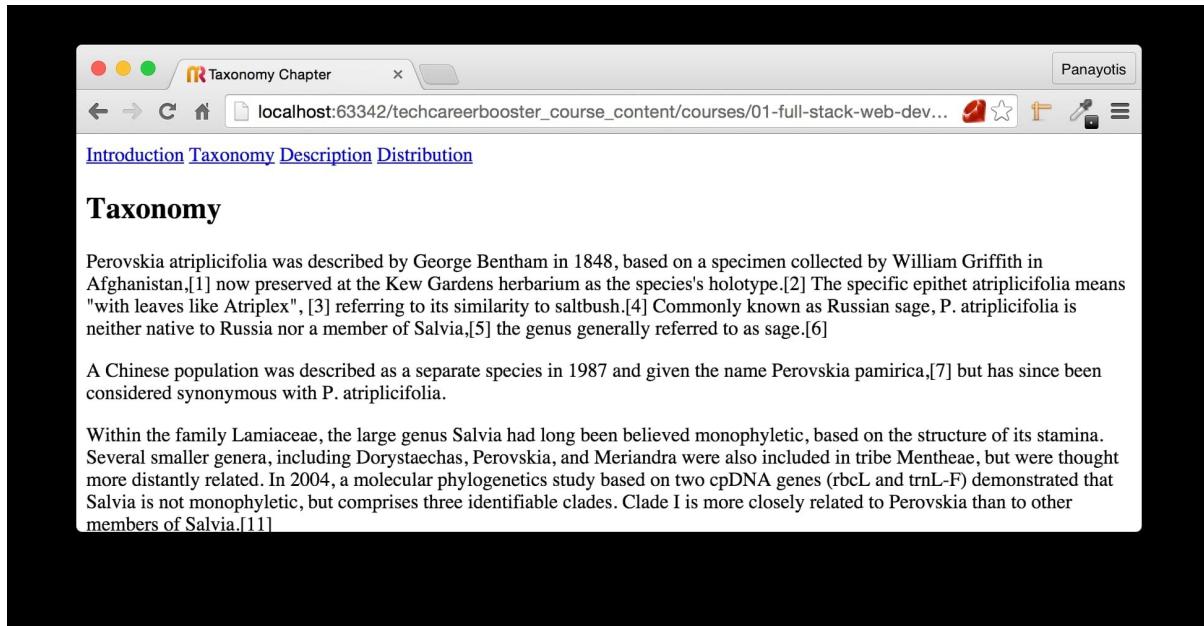
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Taxonomy Chapter</title>
6   </head>
7
8   <body>
9
10    <div id="navigation">
11      <a href="main.html">Introduction</a>
```

```
12      <a href="chapters/taxonomy.html">Taxonomy</a>
13      <a href="chapters/description.html">Description</a>
14      <a href="chapters/distribution.html">Distribution</a>
15  </div>
16
17  <div id="taxonomy">
18      <h2>Taxonomy</h2>
19
20      <p>
21          Perovskia atriplicifolia was described by George Bentham in 1848, based o\
22 n a specimen collected by William Griffith in Afghanistan,[1]
23          now preserved at the Kew Gardens herbarium as the species's holotype.[2] \
24 The specific epithet atriplicifolia means "with leaves like Atriplex", \
25 referring to its similarity to saltbush.[4] Commonly known as Russian\
26 sage, P. atriplicifolia is neither native to Russia nor a member of \
27 Salvia,[5] the genus generally referred to as sage.[6]
28  </p>
29
30  <p>
31          A Chinese population was described as a separate species in 1987 and give\
32 n the name Perovskia pamirica,[7] but has since been considered \
33 synonymous with P. atriplicifolia.
34  </p>
35
36  <p>
37          Within the family Lamiaceae, the large genus Salvia had long been believe\
38 d monophyletic, based on the structure of its stamens.
39          Several smaller genera, including Dorystaechas, Perovskia, and Merianandra \
40 were also included in tribe Mentheae, but were thought \
41 more distantly related. In 2004, a molecular phylogenetics study based on\
42 two cpDNA genes (rbcL and trnL-F) demonstrated that
43          Salvia is not monophyletic, but comprises three identifiable clades. Clad\
44 e I is more closely related to Perovskia than to other \
45 members of Salvia.[11]
46  </p>
47
48  <p>
49          P. atriplicifolia has been the subject of subsequent studies seeking to c\
50 larify the relationships within Mentheae. Further research \
51 combined palynological analysis of pollen grains with rbcL sequencing to \
52 provide additional support for the relationship between
53          Perovskia and Salvia clade I. It also distinguished between P. atriplicif\
54 olia and P. abrotanoides, while confirming their close \
55 relationship.[12] A subsequent multigene study (four cpDNA markers and tw\
56 o nrDNA markers) redrew parts of the Mentheae cladogram,
57 making Rosmarinus a sister group to Perovskia.
```

```
58      </p>
59
60      <p>
61          Several cultivars of P. atriplicifolia have been developed. They are prim\
62 arily distinguished by the height of mature plants and the depth
63          of the leaf-margin incisions.[13] Many of these cultivars, especially tho\
64 se with deeply incised leaves, may actually be hybrids of P.
65          atriplicifolia and P. abrotanoides.[13][14] In that context, some may be \
66 referred to by the hybrid name P. ×hybrida.[14][15]
67      </p>
68
69      <p>
70          The most common cultivar,[16] 'Blue Spire', is among those suspected of b\
71 eing a hybrid.[17][18] It was selected from German plantings
72          by the British Notcutts Nurseries, and first exhibited in 1961.[19][20] '\
73 Blue Spire' grows to approximately 1.2 m (3 ft 11 in), and
74          has large, darker blue flowers.[5][17] In 1993, it received the Royal Hor\
75 ticultural Society's Award of Garden Merit.[21]
76      </p>
77
78      <p>
79          'Filigran' reaches a height of 1.2 to 1.3 m (3 ft 11 in to 4 ft 3 in); th\
80 is tall, sturdy cultivar's name is German for filigree,
81          in reference to its lacy, fern-like foliage.[5][19] 'Little Spire' is sho\
82 rter, with a mature height of only 0.6 m (2 ft 0 in).[16][22]
83          'Longin' is similar in height to 'Blue Spire' but more upright.[5] Allan \
84 Armitage established the late-flowering cultivar 'Mystery of
85 Knightshayes' from a plant at Knightshayes Court.[19] Other cultivars inc\
86 lude 'Blue Haze', 'Blue Mist', 'Hybrida' (also called 'Superba'),
87          'Lace', 'Lisslit', 'Rocketman', and 'WALPPB'
88      </p>
89
90      </div>
91
92  </body>
93
94  </html>
```

(the above code snippet online)

If we save and reload this page, we will see this:



Site With Many Pages - Taxonomy Page

This looks correct. The taxonomy page has been loaded. But what happens if you click the “Introduction” link? Does it take you to the `main.html` page with the `Introduction` header?

No. It doesn't. This is because the `Introduction` anchor has the `href` value that is not valid inside the `chapters/taxonomy.html` page. It has the value `main.html`. But there is no file `main.html` inside the current page folder, i.e. `chapters` folder. The `main.html` file, resides in the parent folder of the `chapters` folder. Hence, inside the `chapters/taxonomy.html` file, the `href` link for `Introduction` needs to have the following HTML code:

```
1  <a href="../main.html">Introduction</a>
```

(the above code snippet online)

This is because the `main.html` file *lives* inside the parent (...) folder of the current page. Take into account again that, when we have taxonomy page loaded, the actual file loaded is `chapters/taxonomy.html`. And the `main.html` file exists in its parent folder.

Try to change that and reload the taxonomy page. Then try to click to `Introduction` link. Does it take you back to the introduction page?

Yes! It does!

Now, click the `Taxonomy` link again to navigate / reload the taxonomy page.

Then, try to click on the `Taxonomy` link. Is everything ok?

No! It's not!

This is because the `Taxonomy` link inside the `chapters/taxonomy.html` file is wrong. It is correct if it were inside `main.html`, but inside `chapters/taxonomy.html` is wrong. Its `href` value is `chapters/taxonomy.html`. When we click this wrong link, browser is trying to find a folder with name “`chapters`” inside the folder where the current page is. But the current page is “`chapters/taxonomy.html`” and there is no other folder “`chapters`” inside the “`chapters`” folder.

Since, Taxonomy link it needs to take us inside the same page as the current one, we can correct the `href` value as follows:

```
1 <a href="taxonomy.html">Taxonomy</a>
```

(the above code snippet online)

If you do the above change inside the `chapters/taxonomy.html` file and reload and then try to visit Taxonomy link again, it will work as expected.

Great. Introduction and Taxonomy links inside the taxonomy page are now working fine. What about the Description link. If you click on that, you will see that it fails. The problem is that its `href` attribute has value equal to `chapters/description.html`. There isn't folder `chapters` relative to the current page (`chapters/taxonomy.html`) and that is the reason we have this failure. File `description.html` lives on the same folder as the `taxonomy.html` and hence we only have to define its name, without the folder `chapters` (which is used to refer to a file only with regards to root folder of the project).

For the same reasons, the link to Distribution fails too.

Correct both Description and Distribution links (inside `taxonomy.html` page) as follows:

```
1 <a href="description.html">Description</a>
2 <a href="distribution.html">Distribution</a>
```

(the above code snippet online)

If you do this correction and reload the taxonomy page, you will see that now all links are working perfectly. They take you to the page that is defined by the link.

We need to add the navigation links HTML fragment to the other two pages too: a) `chapters/description.html` and b) `chapters/distribution.html`. This should be

```
1 <div id="navigation">
2   <a href="../main.html">Introduction</a>
3   <a href="taxonomy.html">Taxonomy</a>
4   <a href="description.html">Description</a>
5   <a href="distribution.html">Distribution</a>
6 </div>
```

(the above code snippet online)

being the same like the HTML fragment we are using for `chapters/taxonomy.html`.

This is not a surprise. Do you understand why? The `chapters/description.html` and `chapters/distribution.html` have the same relative position with the `chapters/taxonomy.html`. Hence, links to pages should be the same.

Styling the links on navigation bar

We have managed to create our 4 pages web site and we have links on top that would allow us to travel from one page to the other. However, according to the mockups, which are our target here, we need to style the links of the navigation bar as follows:

1. All links irrespective of their state (link, visited, active, except hover one...see below) should have the same style: no underline, let's say maroon color.
2. The links should change background color and text color on hover effect. Their background color should become darkblue and their text color should be white.
3. The link that corresponds to current page should have background color maroon and text color white.

So:

1. When current page is “main.html”, we highlight the Introduction link
2. When current page is “chapters/taxonomy.html”, we highlight the Taxonomy link
3. When current page is “chapters/description.html”, we highlight the Description link
4. When current page is “chapters/distribution.html”, we highlight the Distribution link

Let's try to transfer the above requirements to CSS rules:

```

1  a {
2      text-decoration: none;
3      color: maroon;
4  }
5
6  a:hover {
7      background-color: darkblue;
8  }
9
10 .current-page-link {
11     background-color: maroon;
12     color: white;
13 }
```

(the above code snippet online)

Save this file to `stylesheets/main.css` and then create a reference like this:

```
1  <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
```

(the above code snippet online)

inside the `<head>` section of `main.html` page.

Also, make sure that the Introduction link inside the `main.html` page has the class `current-page-link`. This will make sure that link stand out from the others, indicating what is the current page visitor is in. Here it is the final version of the `main.html` page:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Site With Many Pages</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="navigation">
12             <a href="main.html" class="current-page-link">Introduction</a>
13             <a href="chapters/taxonomy.html">Taxonomy</a>
14             <a href="chapters/description.html">Description</a>
15             <a href="chapters/distribution.html">Distribution</a>
16         </div>
17
18         <div id="introduction">
19             <h2>Introduction</h2>
20
21             <p>
22                 Perovskia atriplicifolia (/pərəvskɪə ətrɪplɪsɪfɔɪliə/), commonly called \
23                 Russian sage, is a flowering herbaceous perennial plant
24                 and subshrub. Although not a member of Salvia, the genus of other plants \
25                 commonly called sage, it is closely related to them.
26                 It has an upright habit, typically reaching 0.5–1.2 m (1 ft 8 in–3 ft 11 \
27                 in) tall, with square stems and grey-green leaves that
28                 yield a distinctive odor when crushed. It is best known for its flowers. \
29                 Its flowering season extends from mid-summer to late October,
30                 with blue to violet blossoms arranged into showy, branched panicles.
31             </p>
32
33         </div>
34
35     </body>
36 </html>

```

(the above code snippet online)

After the above changes, if you reload the `main.html` page, you will see the links styled exactly as we have requested.

Now, if you click to travel to another page, for example to taxonomy page, you will see that the style of the links is the default one and not the one we have on introduction page. This is because, the taxonomy file does not load the CSS file with the CSS rules.

So, `chapters/taxonomy.html` needs to reference the `stylesheets/main.css` file from within its `<head>` section.

The correct link is this:

```
1 <link rel="stylesheet" href="..../stylesheets/main.css" type="text/css">
```

(the above code snippet online)

Do you see the .. that is used to go to the parent folder of the current page folder?

Also, don't forget to make the Taxonomy link the one that has the class `current-page-link` now.

The final version of the `chapters/taxonomy.html` file should be:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Taxonomy Chapter</title>
6     <link rel="stylesheet" href="..../stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10
11     <div id="navigation">
12       <a href="..../main.html">Introduction</a>
13       <a href="taxonomy.html" class="current-page-link">Taxonomy</a>
14       <a href="description.html">Description</a>
15       <a href="distribution.html">Distribution</a>
16     </div>
17
18     <div id="taxony">
19       <h2>Taxonomy</h2>
20
21       <p>
22         Perovskia atriplicifolia was described by George Bentham in 1848, based on a specimen collected by William Griffith in Afghanistan,[1]
23         now preserved at the Kew Gardens herbarium as the species's holotype.[2] The specific epithet atriplicifolia means "with leaves like Atriplex",[3] referring to its similarity to saltbush.[4] Commonly known as Russian sage, P. atriplicifolia is neither native to Russia nor a member of Salvia,[5] the genus generally referred to as sage.[6]
24       </p>
25
26       <p>
27         A Chinese population was described as a separate species in 1987 and given the name Perovskia pamirica,[7] but has since been considered synonymous with P. atriplicifolia.
28       </p>
29
30
31
32
33
34
35
36
```

```
37      <p>
38          Within the family Lamiaceae, the large genus Salvia had long been believed\
39  d monophyletic, based on the structure of its stamens.
40          Several smaller genera, including Dorystaechas, Perovskia, and Merianandra \
41  were also included in tribe Mentheae, but were thought
42          more distantly related. In 2004, a molecular phylogenetics study based on\
43  two cpDNA genes (rbcL and trnL-F) demonstrated that
44          Salvia is not monophyletic, but comprises three identifiable clades. Clad\
45  e I is more closely related to Perovskia than to other
46          members of Salvia.[11]
47      </p>
48
49      <p>
50          P. atriplicifolia has been the subject of subsequent studies seeking to c\
51 larify the relationships within Mentheae. Further research
52          combined palynological analysis of pollen grains with rbcL sequencing to \
53 provide additional support for the relationship between
54          Perovskia and Salvia clade I. It also distinguished between P. atriplicif\
55 olia and P. abrotanoides, while confirming their close
56          relationship.[12] A subsequent multigene study (four cpDNA markers and tw\
57 o nrDNA markers) redrew parts of the Mentheae cladogram,
58          making Rosmarinus a sister group to Perovskia.
59      </p>
60
61      <p>
62          Several cultivars of P. atriplicifolia have been developed. They are prim\
63 arily distinguished by the height of mature plants and the depth
64          of the leaf-margin incisions.[13] Many of these cultivars, especially tho\
65 se with deeply incised leaves, may actually be hybrids of P.
66          atriplicifolia and P. abrotanoides.[13][14] In that context, some may be \
67 referred to by the hybrid name P. ×hybrida.[14][15]
68      </p>
69
70      <p>
71          The most common cultivar,[16] 'Blue Spire', is among those suspected of b\
72 eing a hybrid.[17][18] It was selected from German plantings
73          by the British Notcutts Nurseries, and first exhibited in 1961.[19][20] '\
74 Blue Spire' grows to approximately 1.2 m (3 ft 11 in), and
75          has large, darker blue flowers.[5][17] In 1993, it received the Royal Hor\
76 ticultural Society's Award of Garden Merit.[21]
77      </p>
78
79      <p>
80          'Filigran' reaches a height of 1.2 to 1.3 m (3 ft 11 in to 4 ft 3 in); th\
81  is tall, sturdy cultivar's name is German for filigree,
82          in reference to its lacy, fern-like foliage.[5][19] 'Little Spire' is sho\
```

```

83 rter, with a mature height of only 0.6 m (2 ft 0 in).[16][22]
84 'Longin' is similar in height to 'Blue Spire' but more upright.[5] Allan \
85 Armitage established the late-flowering cultivar 'Mystery of
86 Knightshayes' from a plant at Knightshayes Court.[19] Other cultivars include \
87 'Blue Haze', 'Blue Mist', 'Hybrida' (also called 'Superba'),
88 'Lace', 'Lisslit', 'Rocketman', and 'WALPPB'
89 </p>
90
91 </div>
92
93 </body>
94
95 </html>
```

(the above code snippet online)

`chapters/description.html` and `chapters/distribution.html` need to have the correct reference to the CSS file in their `<head>` section too.

```
1 <link rel="stylesheet" href="..../stylesheets/main.css" type="text/css">
```

(the above code snippet online)

This is exactly the same like the one used inside `chapters/taxonomy.html`.

The final version of the `chapters/description.html` should be:

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Description Chapter</title>
6     <link rel="stylesheet" href="..../stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10    <div id="navigation">
11      <a href="..../main.html">Introduction</a>
12      <a href="..../taxonomy.html">Taxonomy</a>
13      <a href="..../description.html" class="current-page-link">Description</a>
14      <a href="..../distribution.html">Distribution</a>
15    </div>
16
17    <div id="description">
18      <h2>Description</h2>
19
20      <p>
21        Perovskia atriplicifolia is a deciduous perennial subshrub with an erect \
```

22 to spreading habit.[13][27] Superficially, it resembles a much
23 larger version of lavender.[28] Multiple branches arise from a shared roo\
24 tstalk,[8] growing to a height of 0.5–1.2 m (1 ft 8 in–3 ft 11 in),
25 [8][22] with occasional specimens reaching 1.5 m (4 ft 11 in).[4] The mat\
26 ure plant may be 0.6–1.2 m (2 ft 0 in–3 ft 11 in) across.[4]
27 The rigid stems are square in cross-section,[4] and are covered by a indu\
28 mentum formed by stellate, or star-shaped, trichomes and oil droplets.[27]
29 Especially during autumn, these hairs give the stems a silvery appearance\
30 .[29]
31 </p>
32
33 <p>
34 The grayish-green leaves are arranged in opposite pairs,[13][30] and atta\
35 ched to the stems by a short petiole.[27] They are generally 3–5 cm
36 (1.2–2.0 in) long and 0.8–2 cm (0.3–0.8 in) wide,[27] although narrower i\
37 n some populations.[8] The overall leaf shape is oblate, a rounded
38 shape longer than it is wide, to lanceolate, shaped like the head of a la\
39 nce.[27] They are pinnatipartite,[8] with a deeply incised
40 leaf margin that may be either wavy or sharp-toothed; even within a singl\
41 e community of *P. atriplicifolia*, there can be considerable
42 variation in the details of leaf shape.[27] Leaves near the top of branch\
43 es may merge into bracts.[27] The foliage is aromatic,
44 especially when crushed,[4] with a fragrance described as sage-like,[5] a\
45 blend of sage and lavender,[16] or like turpentine.[31]
46 </p>
47
48 <p>
49 The flowering season of *P. atriplicifolia* can be as long as June through \
50 October,[27] although populations in some parts of its range,
51 such as China, may bloom in a much more restricted period.[8] The inflore\
52 scence is a showy panicle, 30–38 cm (12–15 in) long,[4] with
53 many branches.[32] Each of these branches is a raceme, with the individua\
54 l flowers arranged in pairs called verticillasters.[8] Each
55 flower's calyx is purple, densely covered in white or purple hairs, and a\
56 bout 4 mm long. The corolla is tube-shaped, formed from a
57 four-lobed upper lip and a slightly shorter lower lip; the blue or violet\
58 blue petals are about 1 cm long.[8][32] The style has been
59 reported in both an exserted-extending beyond the flower's tube-form and \
60 one contained within the flower;[32] all known examples of *P.*
61 *atriplicifolia* in cultivation have exserted styles.[13] Gardening author \
62 Neil Soderstrom describes the appearance of the flowers from a
63 distance as "like a fine haze or fog".[33]
64 </p>
65 <p>
66 Fruits develop about a month after flowering,[8] and consist of dark brow\
67 n oval nutlets, about 2 mm × 1 mm.

```
68      </p>
69
70      </div>
71
72      </body>
73
74  </html>
```

(the above code snippet online)

And the final version of the chapters/distribution.html is:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Distribution Chapter</title>
6          <link rel="stylesheet" href="../stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div id="navigation">
11             <a href="../main.html">Introduction</a>
12             <a href="taxonomy.html">Taxonomy</a>
13             <a href="description.html">Description</a>
14             <a href="distribution.html" class="current-page-link">Distribution</a>
15         </div>
16
17         <div id="distribution">
18             <h2>Distribution</h2>
19
20             <p>
21                 Widely distributed across Asia in its native range, Perovskia atriplicifolia
22                 grows in western China,[4] Pakistan, Afghanistan, Iran,[37]
23                 Turkey, and parts of eastern Europe.[38] It is found in steppes and on hillsides,[38]
24                 and grows at higher elevations in mountainous regions,
25                 including the Himalayas.[39] It has been recorded at 10,000 ft (3,000 m) of altitude
26                 in the Karakoram.[40] In Pakistan's Quetta district, it is often found in association
27                 with the grass Chrysopogon aucheri,[41] and may serve as an indicator species for soils
28                 with low calcium carbonate and chloride availability.[42] The harsh habitats preferred by P.
29                 atriplicifolia are comparable to the sagebrush steppe of North America.[43]
30             </p>
31
32             <p>
33                 In parts of its range, such as the Harboi, these steppe ecosystems are employed
34                 as rangeland for grazing animals such as sheep and goats, although
```

```
36      this forage is generally of poor nutritional quality. P. atriplicifolia c\
37  an serve as an important source of phosphorus and zinc,[44] despite being
38      high in poorly-digested material such as neutral detergent fiber and lign\
39  in
40      </p>
41      </div>
42
43  </body>
44
45 </html>
```

(the above code snippet online)

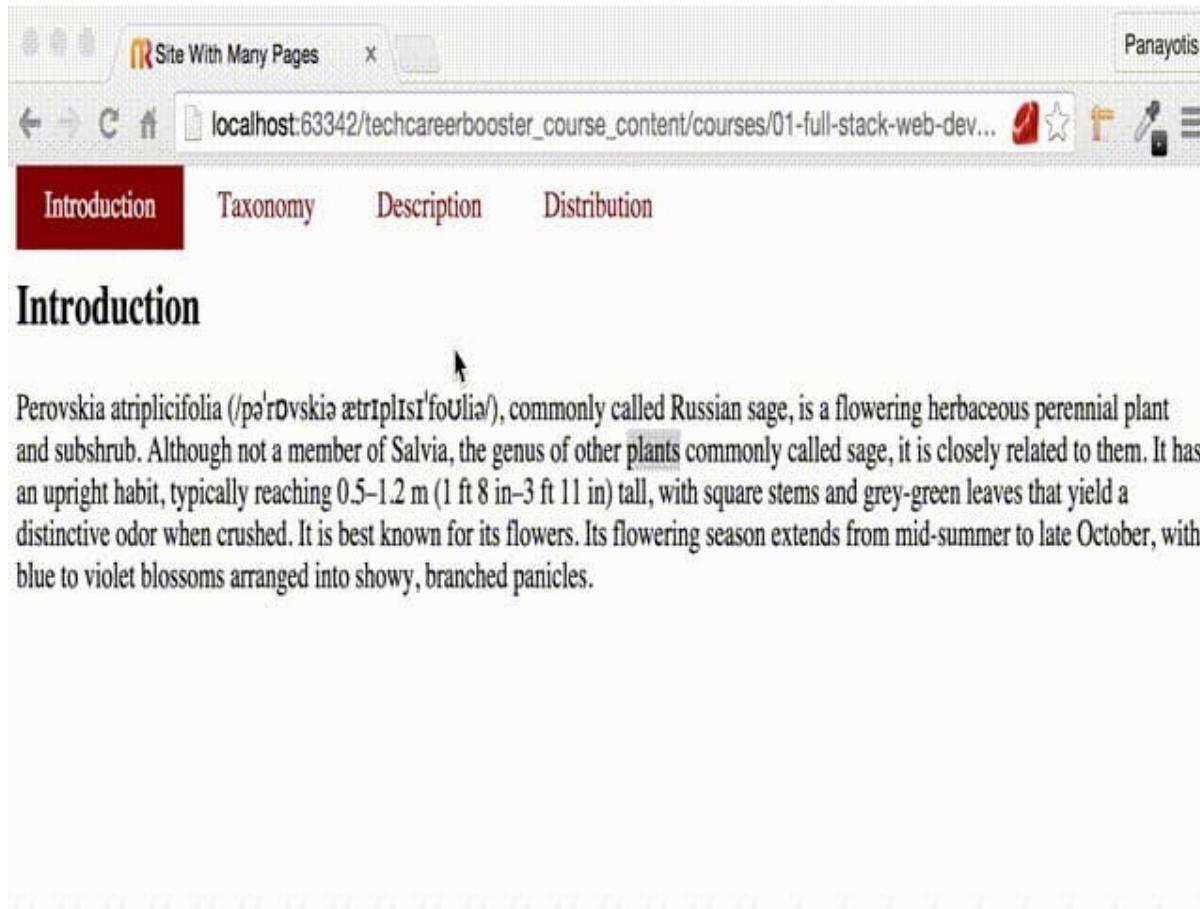
Saving all that content above and reload your pages, you will be able to start linking on the navigation bar from one page to another page, and the navigation links will all have the same style fulfilling the styling requirements that we have set above.

One thing that we can do to improve the style a little bit more, is to give some padding on the links, so that the current page link is displayed better.

```
1 a {
2     text-decoration: none;
3     color: maroon;
4     padding: 10px 20px;
5 }
6
7 a:hover {
8     background-color: darkblue;
9     color: white;
10 }
11
12 .current-page-link {
13     background-color: maroon;
14     color: white;
15 }
```

(the above code snippet online)

We have only added some padding for the links. Now they look much better:



Site With Many Pages -Styled Menu

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

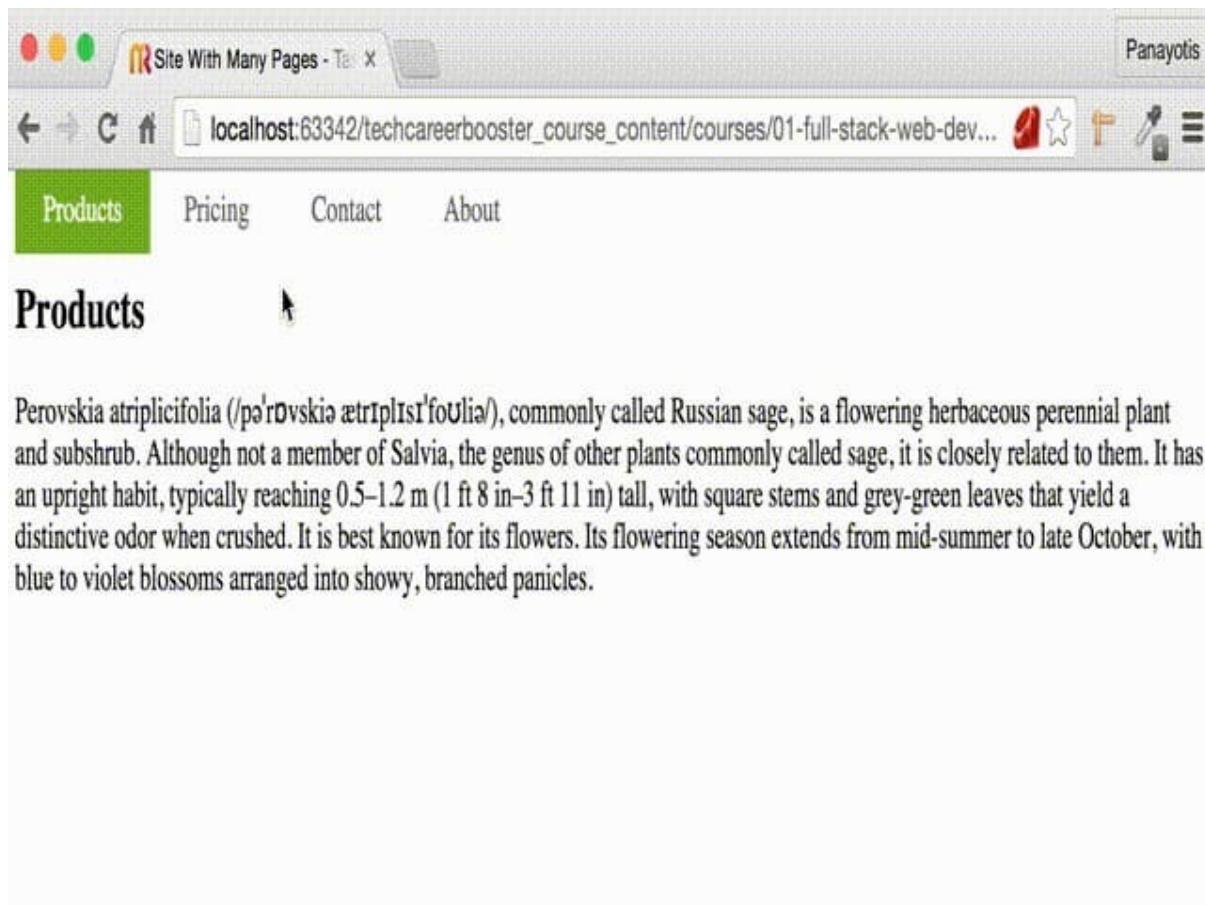
Task:

Task details:

Read all the instructions here, before you start implementing your task.

1. Create a set of 4 pages that would belong to the same site.
2. All pages should have a navigation bar at the top with 4 links to each page.
3. Pages should be:
 1. Products, inside the `main.html`
 2. Pricing, inside the `sections/pricing.html`
 3. About, inside the `sections/about.html`

4. Contact, inside the `sections/contact.html`
4. The content of each page, below the top navigation bar, should have a header `<h2>` and a paragraph `<p>` with the main text. `<h2>` should contain the name of the page so that we know which page we have visited. The content of the `<p>` can be any dummy (e.g. Lorem ipsum...) text. You can enclose any element into any `<div>` element, if you want.
5. Style the links of the navigation bar as follows:
 1. Links should not be underlined.
 2. The current page should be highlighted.
 3. On mouse over the link should be highlighted with a highlight mode different from the one used for current page.
 4. Don't use the same colors as the ones we used for the pages we built while studying the chapter.
 5. Don't use the class `current-page-link`. Use the class `active-page` instead.
6. The style sheet rules should exist inside a file `stylesheets/main.css`.
7. Write inside your CSS file, comments that explain what each rule is doing.
8. Here is what your site might look like:



Chapter 20 - Deploying Our Work

Summary



Netlify

In this chapter we learn how to deploy our site to a hosting provider server. We are using [Netlify](#) to host our site. You will be asked to deploy all your, major, work until now.

Learning Goals

1. Learn about static sites.
2. Learn about static sites hosting providers.
3. Create an account on Netlify.
4. Learn how to deploy a static site that has only one `index.html` file.
5. Learn how to deploy a static site that has both an `index.html` file and CSS file, stored in another folder.
6. Learn about CSS optimization, bundling and minification.
7. Learn how you can disable CSS optimization, for development and debugging reasons.

Deploying Our Work

We have created a lot of HTML pages until now. It's about time to let the world be able to have a look at them. This will expose ourselves and our work and will be our incentive to make our work be very good. Also, we will start understanding little-by-little what does release to production mean.

Static Sites

Currently, we have created static pages for static sites. They are called `static` because they do not build their content at run-time, dynamically. The HTML content preexists inside `.html` files. The content is served as it is and it is not changed on the fly by the server.

Note: We will learn how to change the content of our HTML pages on the client side. This can be done with JavaScript. However, this would be a technique that takes place on the browser, rather than on the server sending the content to browser. That is why one may consider, even in those cases, the site to be static.

Sites are hosted on Servers

Sites are hosted on servers that are 24x7 switched on, publicly available, as part of the global Internet. So, in order for our site to be publicly available, its content needs to be uploaded into such a server.

Companies Offering Hosting

There are many companies that offer hosting for static Web sites. One such company is Netlify and this is the one that we are going to use in this course for our static sites.

Netlify - Create an Account

So, your responsibility now is to create an account with <https://www.netlify.com>. This is very easy. You visit the home page of the service and you click on Start Here. You need to register with your email and provide a password. After that, you will need to confirm your email. Then you are done.

See a short video that demonstrates this process:

[Create a Netlify Account](#)

Deploy our first Site

We are about to deploy our first site. Deploying to Netlify basically means uploading the contents of our site to Netlify. Netlify has a very handy drag-n-drop functionality that allows us to upload our content.

Deploy our Hello World Site

First we will deploy our Hello World Site. In [chapter 3](#), we have created the content of this site.

Important: Each static site, needs to have a file with name `index.html`. This file should exist in the root folder of your site. This is the file that the server is using as a start-navigation file, in order to serve your site to your visitors. So, **name** your hello world HTML page with the filename `index.html` before trying to upload that to Netlify.

Take this file and upload it to your Netlify account, in the area that Netlify tells you to upload your file. Assuming that the file has the name “`index.html`”, watch the following video to see what you have to do:

[How you can upload your hello world site to Netlify](#)

Updating our Site

With Netlify is very easy to update the content of your site. You only have to drag-n-drop the new content in the area that corresponds to your site. Suppose that we want to update our hello world site with the following code:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Hello World Page</title>
6      </head>
7
8      <body>
9          <h1>Hello World</h1>
10         <div>
11             <small>Copyright © ACME 2016</small>
12         </div>
13     </body>
14 </html>
```

(the above code snippet online)

which basically adds a small copyright notice.

Save this file with the name **index.html** and follow the steps on the following video to upload it to Netlify.

Important: On Netlify, do not create a new site every time you want to update a new one.

[How you can update your hello world site on Netlify](#)

Deploying a Site with Stylesheets/CSS

Until now we have uploaded a site that was composed of only 1 file, the **index.html** file. Now, we are going to upload a site that contains both an **index.html** file (which is absolutely necessary for the static sites as we said earlier) and another file **main.css** that resides inside the folder **stylesheets**. So, the structure of the folders and files for our site is as follows:

```

1  -- [stylesheets]
2      |
3          |-- main.css
4  -- index.html
```

(the above code snippet online)

For **index.html** use the following content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Task for divs</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="left-header" class="header-column">
12             <h3>Left Header</h3>
13             <p>
14                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
15 ipsa mauris tellus,
16                 porttitor erat proin ac, quisque vel, dui nec aliquam.
17             </p>
18         </div>
19
20         <div id="right-header" class="header-column">
21             <h3>Right Header</h3>
22             <p>
23                 Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, \
24 ipsa mauris tellus,
25                 porttitor erat proin ac, quisque vel, dui nec aliquam.
26             </p>
27
28         </div>
29
30         <div id="first-left-column" class="central-column">
31             <h3>First Left Column</h3>
32             <p>
33                 Interdum arcu, lorem aenean elit mauris sed, ultricies eu, sed fel\
34 is nunc, porta eu leo in ac bibendum. Justo egestas
35                 porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
36 eos vel porttitor amet erat nunc, sed penatibus turpis.
37                 Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
38 uada, taciti vitae malesuada ac porttitor eget cum,
39                 magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
40 mollis dis. Blandit sollicitudin luctus fringilla placerat
41                 quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
42 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
43                 placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
44 ie erat mattis nullam nostra. Dignissim orci suspendisse in
45                 commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \
46 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu
```

```
47      risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\
48      praesent laoreet, lobortis torquent magna habitasse ut
49          pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\
50      ris molestie maecenas cum arcu hendrerit, in per nunc,
51          ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \
52 Nunc vestibulum lectus ut
53      </p>
54  </div>
55
56  <div id="second-left-column" class="central-column">
57      <h3>Second Left Column</h3>
58      <p>
59          Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\
60      is nunc, porta eu leo in ac bibendum. Justo egestas
61          porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
62      eos vel porttitor amet erat nunc, sed penatibus turpis.
63          Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
64      uada, taciti vitae malesuada ac porttitor eget cum,
65          magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
66      mollis dis. Blandit sollicitudin luctus fringilla placerat
67          quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
68      t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
69          placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
70      ie erat mattis nullam nostra. Dignissim orci suspendisse in
71          commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \
72      mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu
73          risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\
74      praesent laoreet, lobortis torquent magna habitasse ut
75          pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\
76      ris molestie maecenas cum arcu hendrerit, in per nunc,
77          ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \
78 Nunc vestibulum lectus ut
79      </p>
80  </div>
81
82  <div id="first-right-column" class="central-column">
83      <h3>First Right Column</h3>
84      <p>
85          Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\
86      is nunc, porta eu leo in ac bibendum. Justo egestas
87          porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
88      eos vel porttitor amet erat nunc, sed penatibus turpis.
89          Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
90      uada, taciti vitae malesuada ac porttitor eget cum,
91          magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
92      mollis dis. Blandit sollicitudin luctus fringilla placerat
```

```
93      quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
94 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
95      placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
96 ie erat mattis nullam nostra. Dignissim orci suspendisse in
97      commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \
98 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu
99      risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\
100 praesent laoreet, lobortis torquent magna habitasse ut
101      pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\
102 ris molestie maecenas cum arcu hendrerit, in per nunc,
103      ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \
104 Nunc vestibulum lectus ut
105      </p>
106      </div>
107
108      <div id="second-right-column" class="central-column">
109          <h3>Second right Column</h3>
110          <p>
111              Interdum arcu, lorem aenean elit mauris mauris sed, ultricies eu, sed fel\
112 is nunc, porta eu leo in ac bibendum. Justo egestas
113              porttitor sociis libero, et lectus elementum neque massa pede sit, hymena\
114 eos vel porttitor amet erat nunc, sed penatibus turpis.
115              Ut nunc pulvinar nulla. Quis integer aliquam donec suspendisse nibh males\
116 uada, taciti vitae malesuada ac porttitor eget cum,
117              magna sed, nostra quis fringilla dolor ante, mattis id id interdum ipsum \
118 mollis dis. Blandit sollicitudin luctus fringilla placerat
119              quis pellentesque, wisi pede in metus neque etiam tellus, lacus feugiat u\
120 t bibendum mi, pede ut pulvinar purus lacus, etiam non eu
121              placerat pede quam. Dui ullamcorper vivamus aenean turpis, amet in molest\
122 ie erat mattis nullam nostra. Dignissim orci suspendisse in
123              commodo consectetur elementum, suspendisse nunc sem amet lorem interdum \
124 mauris. Dolor pariatur nunc id tincidunt, eros molestie arcu
125              risus pellentesque molestie et, nisl in lorem adipiscing, fusce per donec\
126 praesent laoreet, lobortis torquent magna habitasse ut
127              pharetra blandit. Ultricies ut amet. Facilisis urna donec dolor. Urna mau\
128 ris molestie maecenas cum arcu hendrerit, in per nunc,
129              ac arcu proin consectetur at, amet venenatis tellus curabitur lobortis. \
130 Nunc vestibulum lectus ut
131          </p>
132          </div>
133
134          <div id="footer">
135              <h3>Footer</h3>
136
137              <p>
138                  You can contact us by sending an email to contact@techcareerbooster.com
```

```
139      </p>
140
141      <p>
142          Copyright © TCB
143      </p>
144  </div>
145
146  </body>
147 </html>
```

(the above code snippet online)

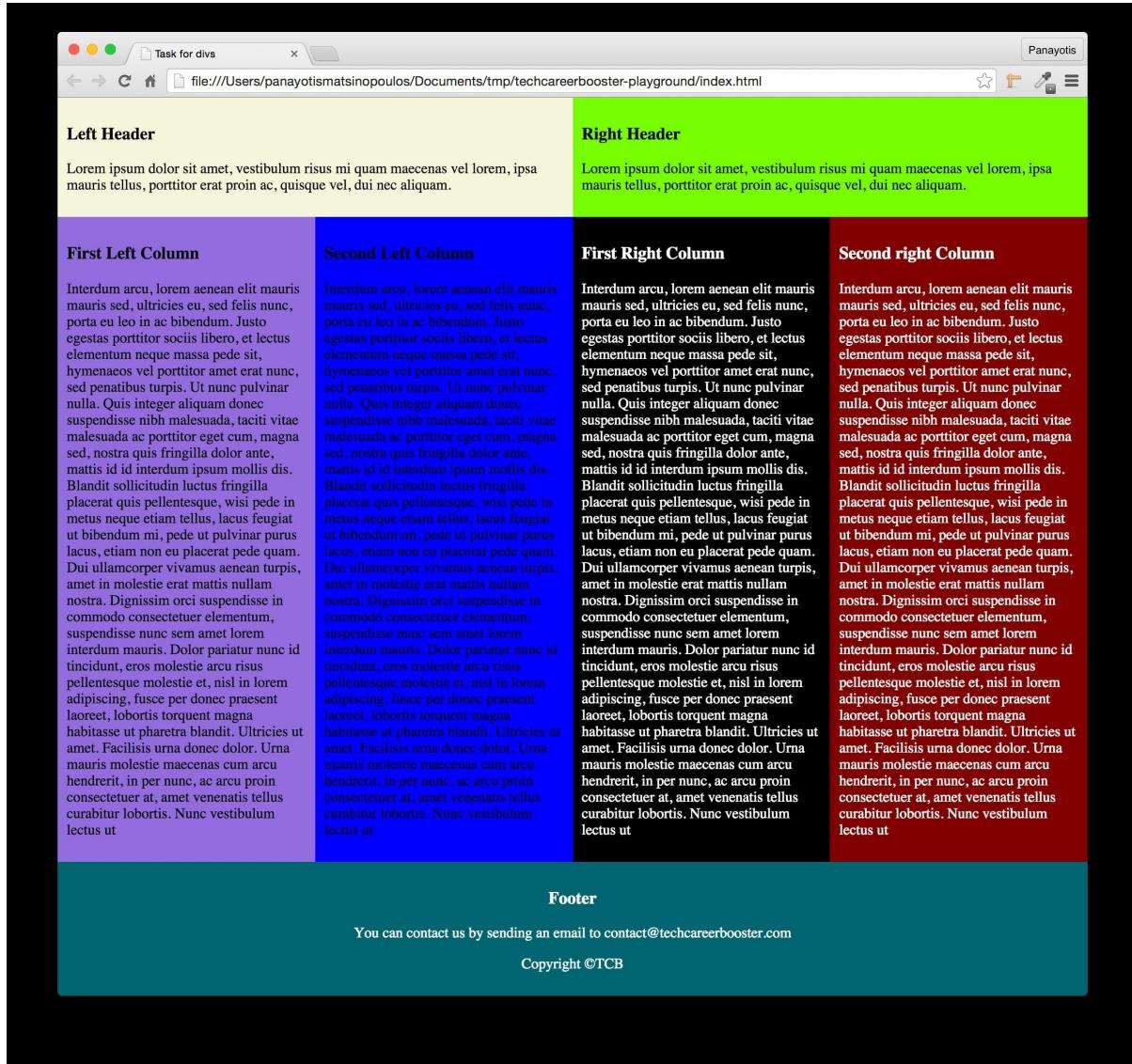
and for the `stylesheets\main.css` file use the content:

```
1  * {
2      box-sizing: border-box;
3  }
4
5  body {
6      margin: 0;
7  }
8
9  div {
10     padding: 10px;
11 }
12
13 .header-column {
14     width: 50%;
15     float: left;
16 }
17
18 #left-header {
19     background-color: #F5F5DC;
20 }
21
22 #right-header {
23     background-color: #7CFC00;
24 }
25
26 .central-column {
27     width: 25%;
28     float: left;
29 }
30
31 #first-left-column {
32     background-color: mediumpurple;
33     clear: left;
```

```
34 }
35
36 #second-left-column {
37     background-color: blue;
38 }
39
40 #first-right-column {
41     background-color: black;
42     color: White;
43 }
44
45 #second-right-column {
46     background-color: maroon;
47     color: White;
48 }
49
50 #footer {
51     text-align: center;
52     background-color: #006470;
53     color: White;
54     clear: left;
55 }
```

(the above code snippet online)

If you load the `index.html` file on your browser, you should see the following:



Site with CSS

Now, we are going to create a new site on Netlify and upload the content above. The important things that you need to take care of are the following:

- We are creating a new site
- We need to select on our file browser (Finder for Mac, Windows File Browser for Windows, File Explorer for Linux) all the files and folders that comprise our site
- Drag the selection of previous step and drop it on the Netlify placeholder that accepts site content for upload.

Watch the video below, before you try that yourself:

[Deploy a site with index and stylesheets folder](#)

Deploying to Netlify using the command line

Netlify offers a command line tool that can be used for you to deploy your site. You can read about it [here](#). However, we will not expand further on that.

Netlify Optimizes our CSS

Netlify, and this is done by many other host providers, optimizes our CSS with the following things that it does:

1. Delivers the CSS content from a file that does not reside on the same host server as the main HTML content.
2. Compresses the CSS content (for example, by removing white spaces and turning the whole content into one line).

If you inspect the source code of the page of your Netlify deployed site, you will experience these two optimizations above. Here is the source code image:

```
html>
<head>
  <meta charset="utf-8">
  <title>Task for divs</title>
  <link rel='stylesheet' type='text/css' href='http://netlify.scdn4.secure.raxcdn.com/css/18718daae74fd3f974ce4f220d46ca616cbe7181/stylesheets/main.css' />
</head>
<body>
  <div id="left-header" class="header-column">
    <h3>Left Header</h3>
  </div>
  Lorem ipsum dolor sit amet, vestibulum risus mi quam maecenas vel lorem, ipsa mauris tellus,
```

Netlify Changes the CSS Reference

As you can see in this picture, Netlify serves the `stylesheets/main.css` file from their optimized CDN network (Content Delivery Network), and not from the URL of your production site. CDNs are optimized to serve static content, like CSS and JavaScript files.

You can copy the URL that is used by Netlify as the CSS reference and paste it on your browser address. This will fetch the CSS that your production site is using. You will see something like that:



The screenshot shows a browser window with the URL `netlify.scdn4.secure.raxcdn.com/css/18718daae74fd3f974ce4f220d46ca616cbe7181/stylesheets/main.css` in the address bar. The page content is a single line of CSS code, indicating that the browser has fetched the compressed version of the CSS file from the CDN.

Netlify Compresses CSS Content

As you can see above, the CSS content is compressed (compare it to the original content of your `stylesheets\main.css` file). It has the blanks removed, it is one line (although displayed wrapped by the browser) and it has other optimizations too.

Watch the following video to understand again the process of inspecting the source code and the CSS compressed content.

[Inspect Source Code and Use Developer Tools To See Optimized CSS](#)

Disable Optimization of your CSS content

Sometimes it is useful to ask Netlify to NOT compress the CSS content. We want Netlify to leave the CSS content as is when we want to do some kind of debugging on deployed site, using the developer tools. Or when we want our site to serve development-mode or staging-mode purposes and not production-mode purposes. In that case, compressed CSS content might not help.

Important On production sites, we always want the hosting provider, like Netlify, to compress our CSS content and serve it from their special CDN server.

If we want to disable CSS optimization, we need to go to our Netlify site settings, on the Post Processing section, click Edit and remove the Bundle CSS and Minify CSS options.

Information: Bundle CSS is used when we have more than 1 CSS file. It concatenates the files into one in order to reduce the number of browser requests. Minify CSS compresses the content of the CSS file.

Go ahead and do this change for your site. Then visit your site and using the developer tools. Inspect the CSS reference (in the head section of your file) and the actual CSS content.

Watch this video to understand what I mean:

[Inspect Source Code and Use Developer Tools To See Non-Optimized Version of your CSS](#)

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

Deploying to Netlify

Important 1: Use the Open Source PRO subscription level for your projects. It is free.

Important 2: For all the sites that you have created, make sure that the settings are set to optimize the CSS (Bundle CSS and Minify CSS are checked).

1. Deploy on Netlify, the site that you have created as task on chapter “Minimum HTML Document”.
2. Deploy on Netlify, the site that you have created as task on chapter “Heading Tags”.
3. Deploy on Netlify, the site that you have created as task on chapter “About White Space”.

4. Deploy on Netlify, the site that you have created as task on chapter HTML Entities.
5. Deploy on Netlify, the site that you have created as task on chapter “Paragraphs Horizontal Lines and Comments”.
6. Deploy on Netlify, the site that you have created as task on chapter “Lists”.
7. Deploy on Netlify, the site that you have created as task on chapter “Tables”.
8. Deploy on Netlify, the site that you have created as task on chapter “CSS Selectors”.
9. Deploy on Netlify, the site that you have created as task on chapter “More Advanced CSS Selectors”.
10. Deploy on Netlify, the site that you have created as task on chapter “Talk About Color”.
11. Deploy on Netlify, the site that you have created as task on chapter “Formatting Parts of Text”.
12. Deploy on Netlify, the site that you have created as task on chapter “HTML Links”.

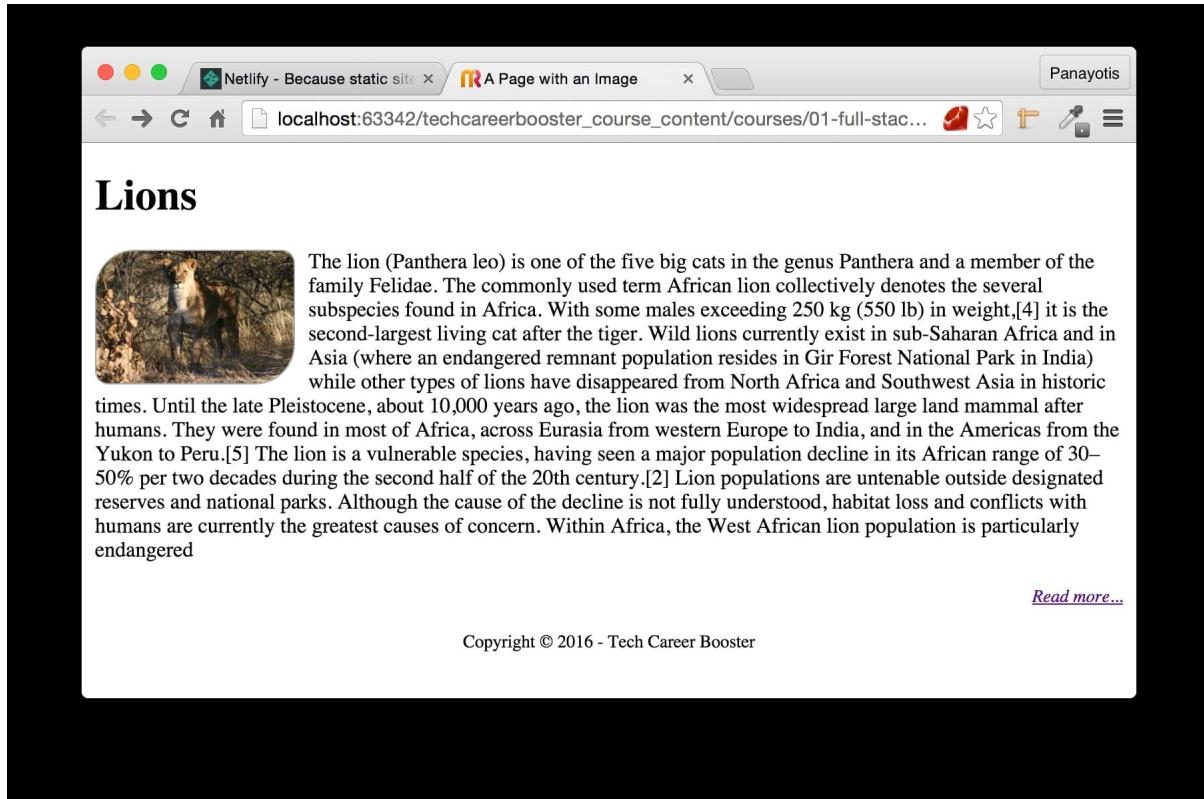
So, at the end of this task, you should have 14 sites deployed on your Netlify account (2 from the content of the chapter and 12 as result of this task)

Chapter 21 - Images

Summary

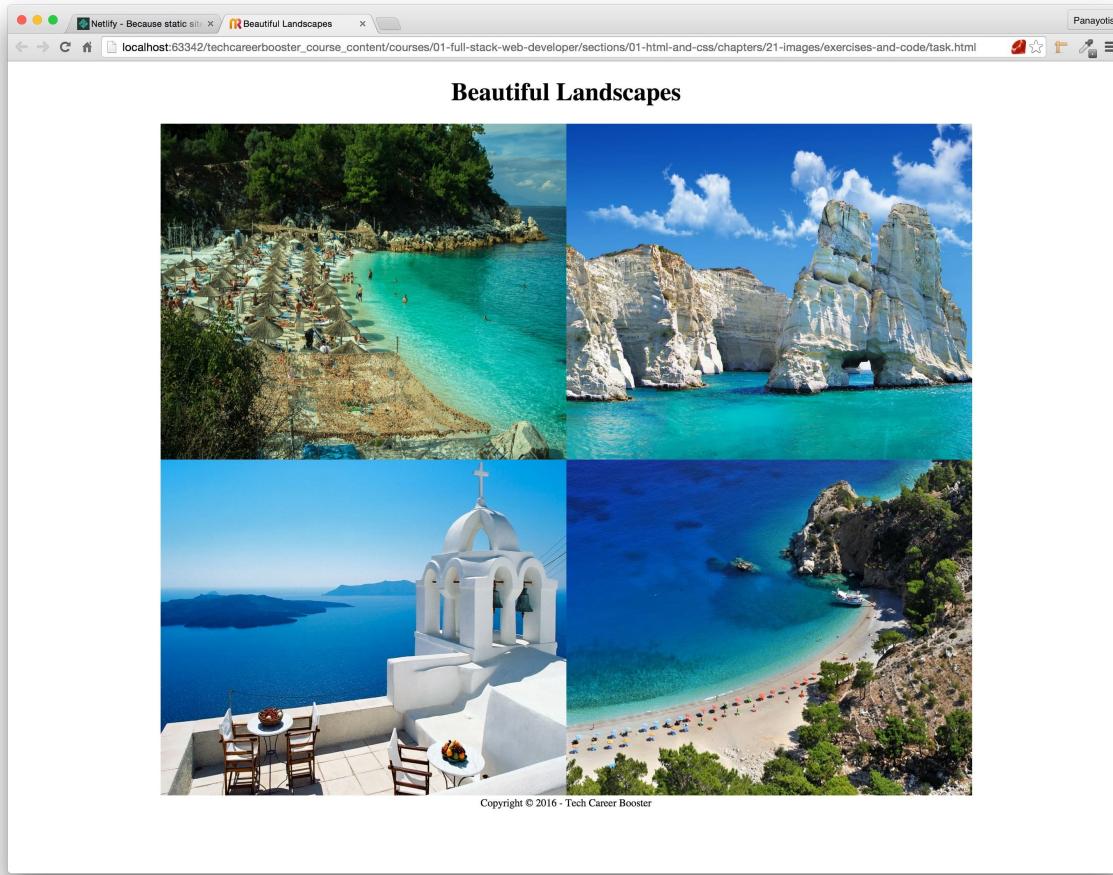
In this chapter you start adding images to your Web pages. Hence, you start making your site much more attractive.

You will build a site like the following:



Web Page with Image

and like this one:



Site With Images

Learning Goals

1. Learn about the most popular image file formats.
2. Learn about the `` tag.
3. Learn about image sizes.
4. Learn about resizing images with `width` and `height` attributes.
5. Learn about resizing images with `style` attribute.
6. Learn about using images as links.
7. Learn about putting text around images.
8. Learn about image transparency / opacity.
9. Learn about image borders and changing the corners of the images.
10. Learn about catching the mouse over event with CSS rules.
11. Learn about images and how their position is affected by line height.
12. Using images from our site.

Images

Besides text, the first simplest visual element is an image. Images will make our site more attractive. Images can be drawings, or photos.

Image Files

Images are digitally stored as files. There are various different formats that are used to store an image:

1. PNG
2. JPG
3. BMP
4. GIF
5. TIFF
6. ICO
7. SVG

and more.

In the context of web development, the most popular formats are:

1. PNG, which is basically used for small images that we want to make sure are of top quality.
Also, PNG support transparent color, which is very useful.
2. JPG, which is basically used for large image files. It has very good compression, although with some loss in quality. But this loss in quality is not enough to be easily perceived by the human eye. So, they are ideal for Web sites photos.
3. ICO, which are used to store small icons images. They can be used as favicons, the image that is displayed by the browser next to the tab title.

If you want to learn more about the different file formats, you can google that. One nice answer is given [here](#).

In Summary: For big photo images, prefer JPG. For small prefer PNG. In any case that you want transparent color, you will have to use PNG format.

 tag

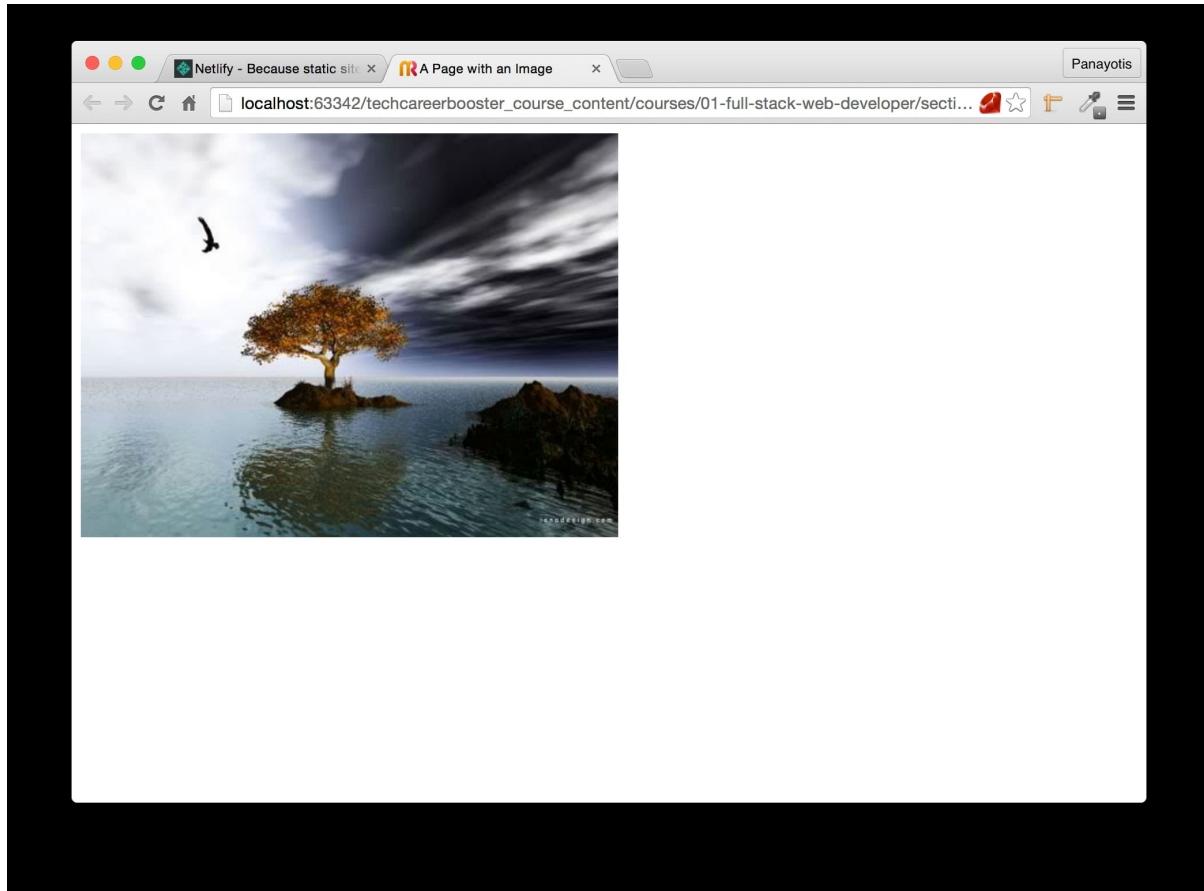
The tag is used to place an image in an HTML page. Let's try that one here:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Page with an Image</title>
6      </head>
7
8      <body>
9
10         
11         alt="Tree in the middle of the sea" title="Tree in the middle of the sea"/>
```

```
12
13  </body>
14
15  </html>
```

(the above code snippet online)

Save the above page into a file named `index.html` and load the file on your browser. You will see the following result:



HTML page with 1 image inside

If you inspect the `img` you will see that this is an `inline` element. This is very important. Text before and text after image are printed adjacent to the image.

The `src` attribute

The `src` attribute is like the `href` on `link` elements in `head` sections, or like the `href` in `anchor` elements. It says where the image file exists. In the previous example, it has the value `http://www.goodlightscraps.com/content/nature/nature-images-86.jpg`. It means that it is hosted inside another server (`www.goodlightscraps.com`) and we are asking the browser to fetch it using the protocol `http`. The actual path inside that remote server is `content/nature/nature-images-86.jpg`.

Image files can reside in folders inside our site, or in remote servers. When local, on our site, we

just mention the relative path to the image file. Like we did with CSS files. If on remote servers, we can access them using `http` or `https`.

Important: If our site is served using SSL (`https` protocol) then it's important that all the content of the site, HTML pages, CSS files, JavaScript files, images and other, to be served either from our site (hence served with same protocol `https`) or from another server that also serves these assets with the same protocol, `https` and not `http`. Otherwise, the user browser will issue warnings to end user. That our secure site includes access to non-secure resources. These warnings might prevent user from continuing navigation to our site.

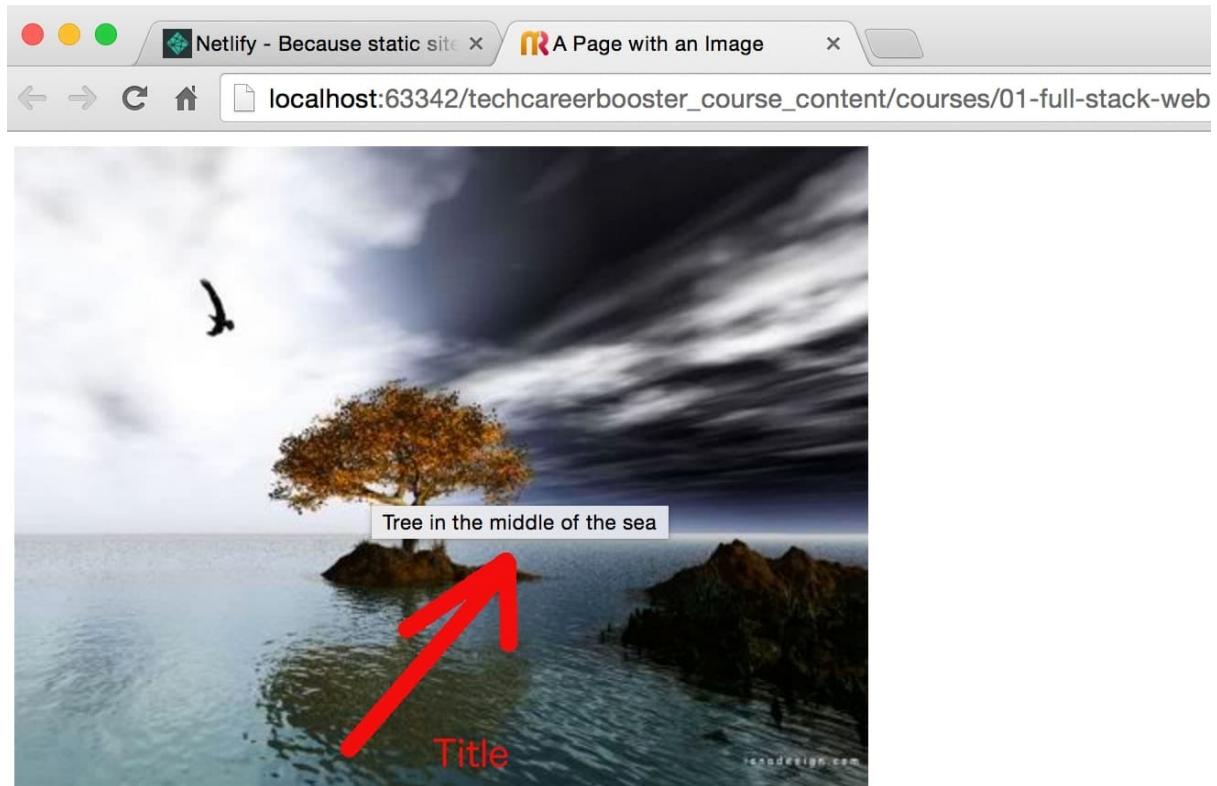
We will talk about locally stored images later on. For now, let's continue with images that live on other remote servers.

The `alt` and the `title` attributes

The `alt` and the `title` attribute accompany the image and they function as follows:

1. `alt` should always contain a descriptive text of the image.
 1. It is used by screen reading software to announce the existence of the image and what is its content.
 2. It is displayed when user has disabled the display of images.
 3. It is displayed when the browser cannot fetch, for any reason, the image.
2. `title` is used to give a descriptive text to the image and it is displayed, as a tooltip, when we mouse over the image.

This is how the title is displayed when you mouse over the image that is referenced by the HTML page we just built.



Title attribute and how it is displayed

Information: Screen reading software is used by visually impaired people that are using computers. It is used to announce to user every element of the screen, so that the user can navigate and use the computer.

Image size

You already know that image size is calculated in pixels. Browser, when displaying an image, it will display it in its original image size. Try with the last HTML page that we developed. Inspect the image element with the developer tools. You will see the size of the image being 466px x 350px. The width comes always first.

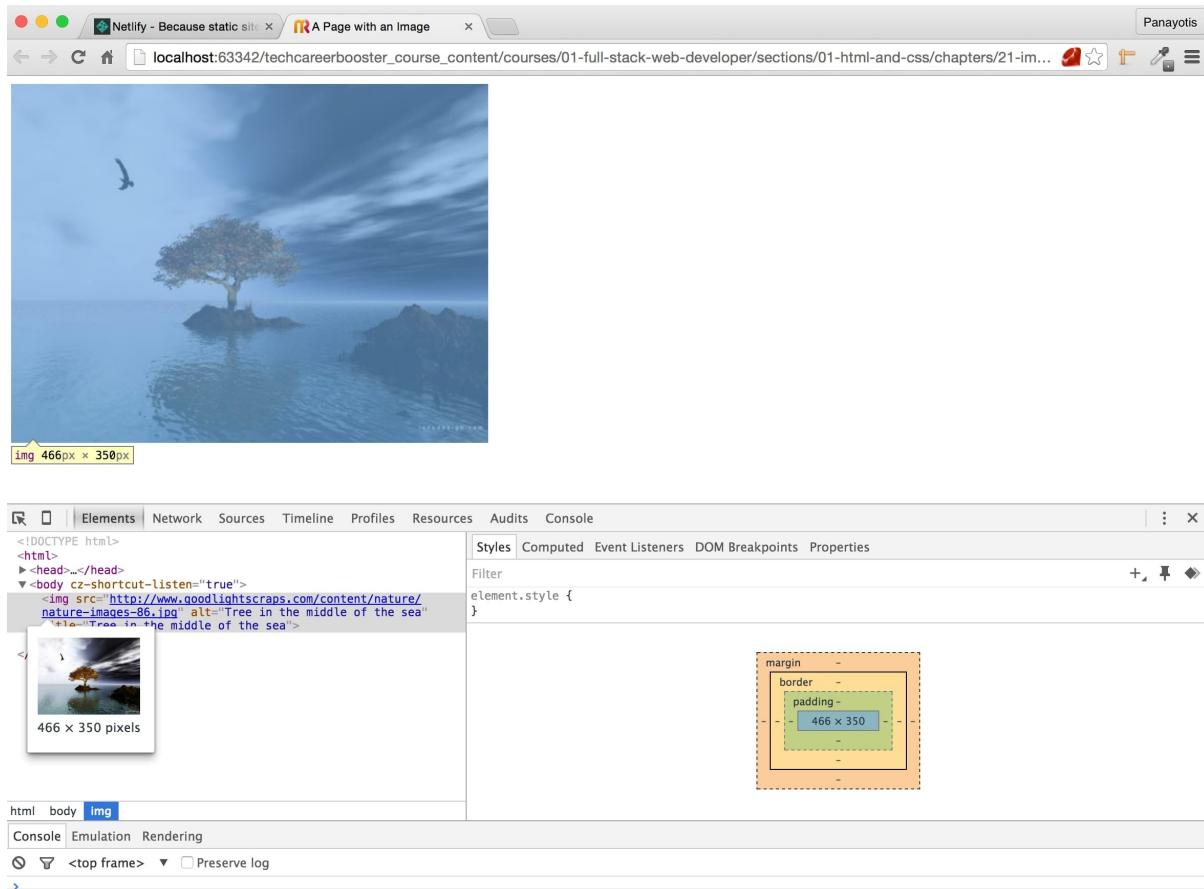


Image Size Inspection with Developer Tools

The size that you see above, is the original size of the image. (You can verify that if you download the image and inspect its size).

Let's try another HTML file that references a really large image:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Page with an Image</title>
6      </head>
7
8      <body>
9
10         
13
14     </body>
15
16 </html>
```

(the above code snippet online)

If you save this file as an `index.html` file and load the page on your browser you will see this:

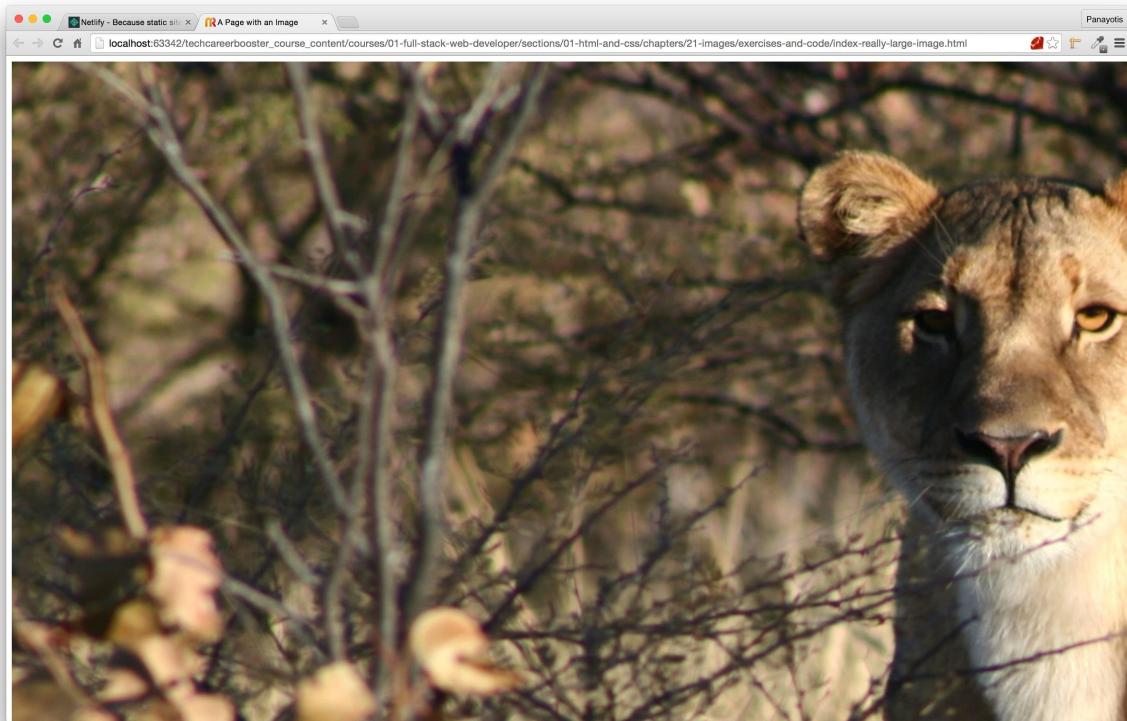
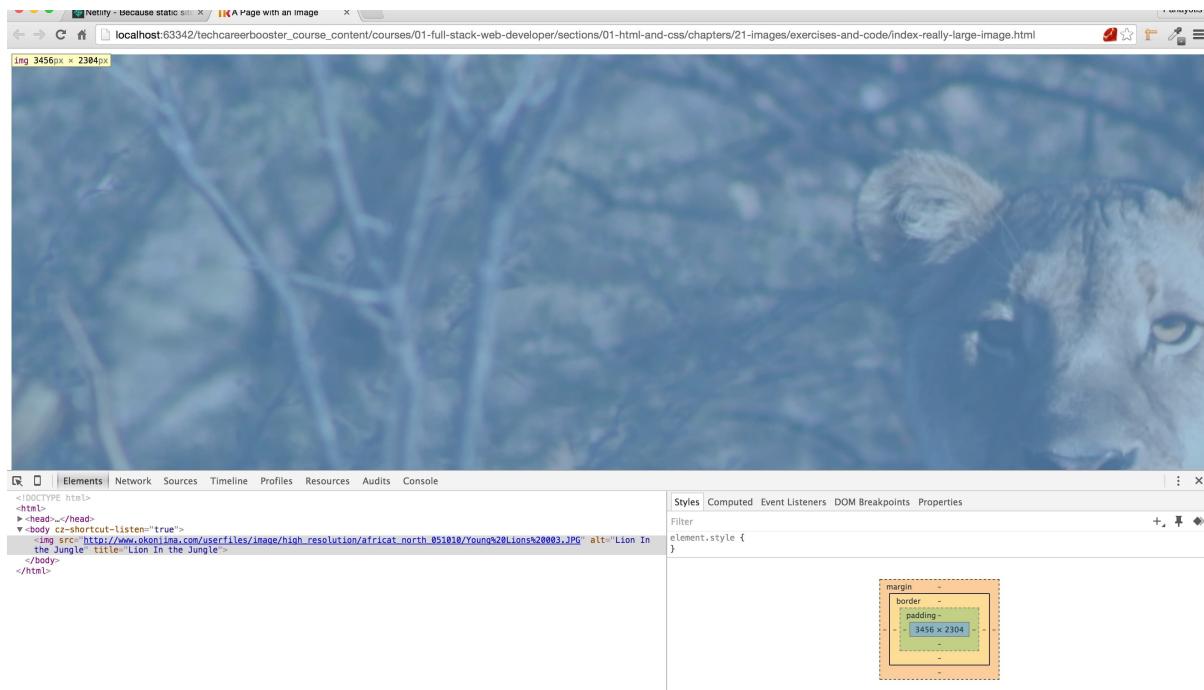


Image With Very Big Size

As you can see, the image cannot fit into the browser window. And if you inspect the image with developer tools, you will see something like this:



Large Image Inspection With Developer Tools

The image is huge: 3456px X 2304px.

width and height attributes

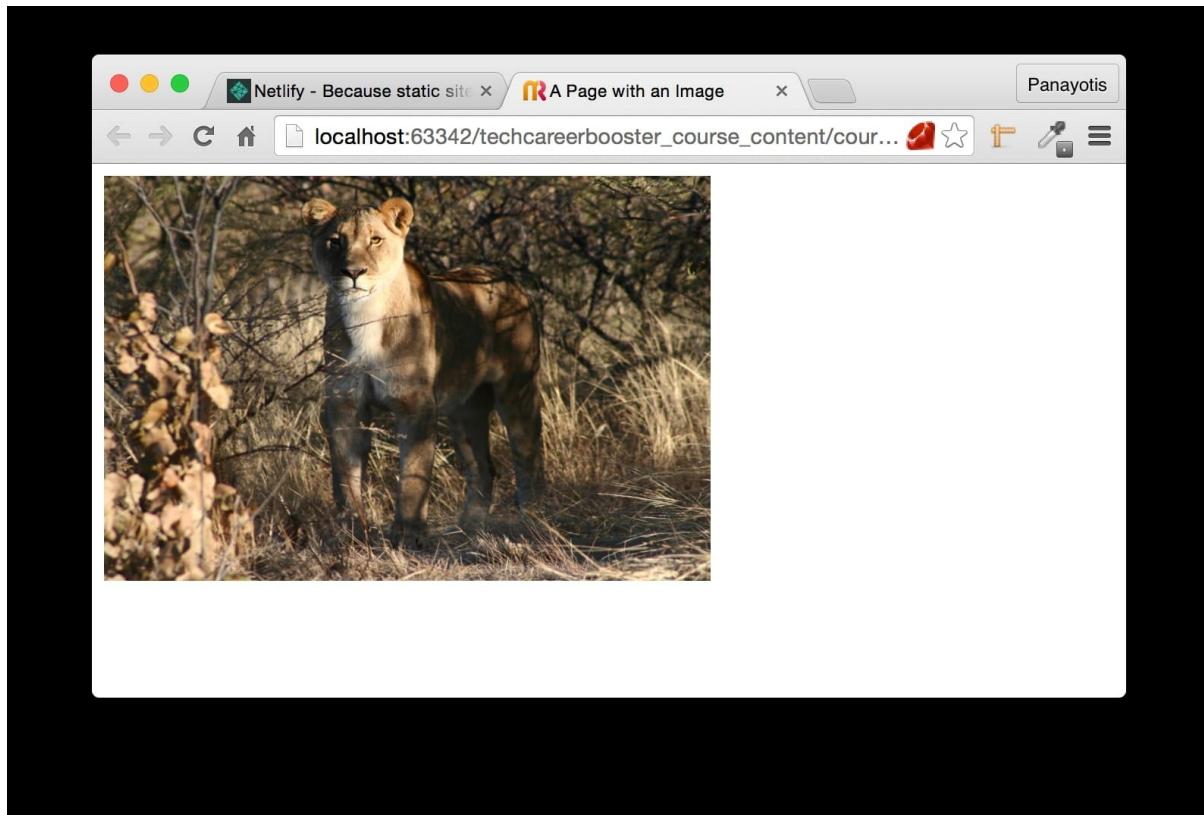
You can use the `width` and `height` attributes in order to ask the browser to resize the image. You can either specify one of the two dimensions, and browser will calculate automatically the other, keeping the aspect ratio of the original sizes (`width / height`), or you can specify both dimensions.

Let's try by specifying the `width` attribute only:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Page with an Image</title>
6      </head>
7
8      <body>
9
10         
14
15     </body>
16
17 </html>
```

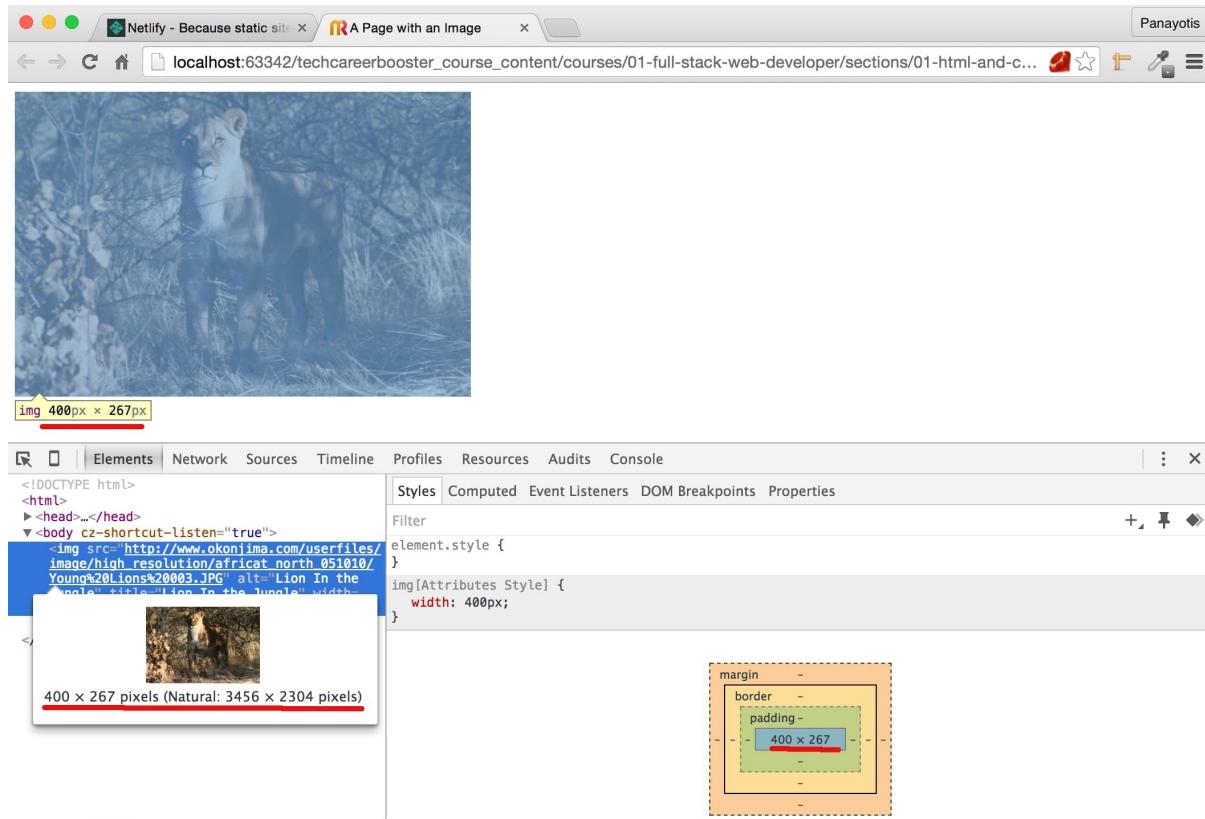
(the above code snippet online)

We have set the `width` attribute to have the value `400`. This means `400px`. If you reload your page, it will look like this:



Large Image Set Width Attribute

Having set the `width` attribute to `400` we ask the browser to reset the width of the original image to `400px` and recalculate the height accordingly, so that it can keep the aspect ratio of the original image. So, height should be $2304 * 400 / 3456 = 266,667$ or `267` to round on nearest integer. Let's prove that by inspecting the new image element using the developer tools. We should see the image size draw to be `400px X 267 px`.



Developer Tools Display Original and Resized Dimensions

As you can see in the above picture, developer tools give you both the original size of the image and the one calculated after the resize (due to the width setting). (make sure you mouse over the “img” element and over the link inside that in the Elements area on the developer tools). Also, you can see that the browser correctly calculated the height of the image, given the 400px width.

Of course, we can certainly set both `width` and `height`. And we can also “destroy” the aspect ratio of the image. Look at the following example:

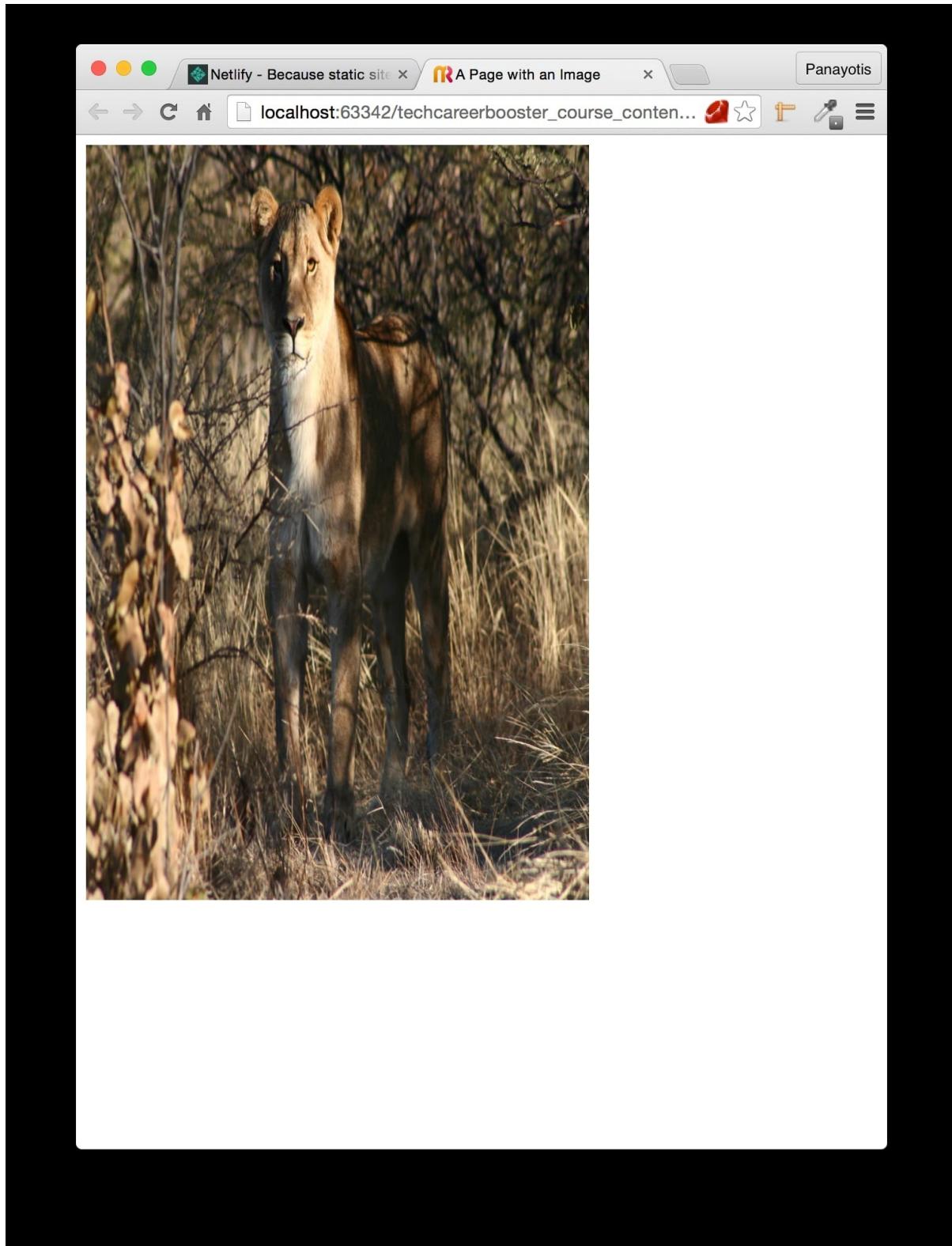
```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Page with an Image</title>
6      </head>
7
8      <body>
9
10         
14
15     </body>
```

```
16  
17  </html>
```

(the above code snippet online)

Save this file and load it on your browser. You will see something like this:



Setting width and height to values that do not respect the aspect ratio of the original image

As you can see, the resizing that we did (`width` equal to `400px` and `height` equal to `600px`) “destroyed” the aspect ratio of the original image. And the image does not look properly. Or it may be what you really wanted to do, case in which it does look “ok”.

Styling using the `style` attribute

Another way you can specify the `width` and/or `height` of an image, is to use the `style` attribute (which is an attribute that can be used on any HTML element) and give it a value that would determine the new `width` and `height` of the image.

Let's take the previous example, and instead of using the `width` and `height` attributes, we will use the `style` attribute:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>A Page with an Image</title>
6    </head>
7
8    <body>
9
10   
14
15  </body>
16
17 </html>
```

(the above code snippet online)

If you load this page on your browser, it will be exactly the same as before, when we used the `width` and `height` attributes with the same values.

So, what's the preferred way to do that? It's the second one with the `style` attribute. And in fact, we prefer this to exist inside a CSS file rather than as an attribute of the `img` element.

Images as Links

We have learned about anchors (`<a>`) and about images (``). We have also learned that we can turn anything into a link. This includes images of course.

Let's try to create the following simple page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Page with an Image</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <a href="https://en.wikipedia.org/wiki/Lion">
12             
16         </a>
17
18     </body>
19
20 </html>
```

(the above code snippet online)

In the above code, you can see how we have included an `` element inside a `<a>` element.

Save the above HTML content to an image with name `index.html`. Also, save the following CSS rule inside a file `stylesheets/main.css`:

```
1  #lion {
2      width: 400px;
3 }
```

(the above code snippet online)

If you load this page on your browser, you will see the following:

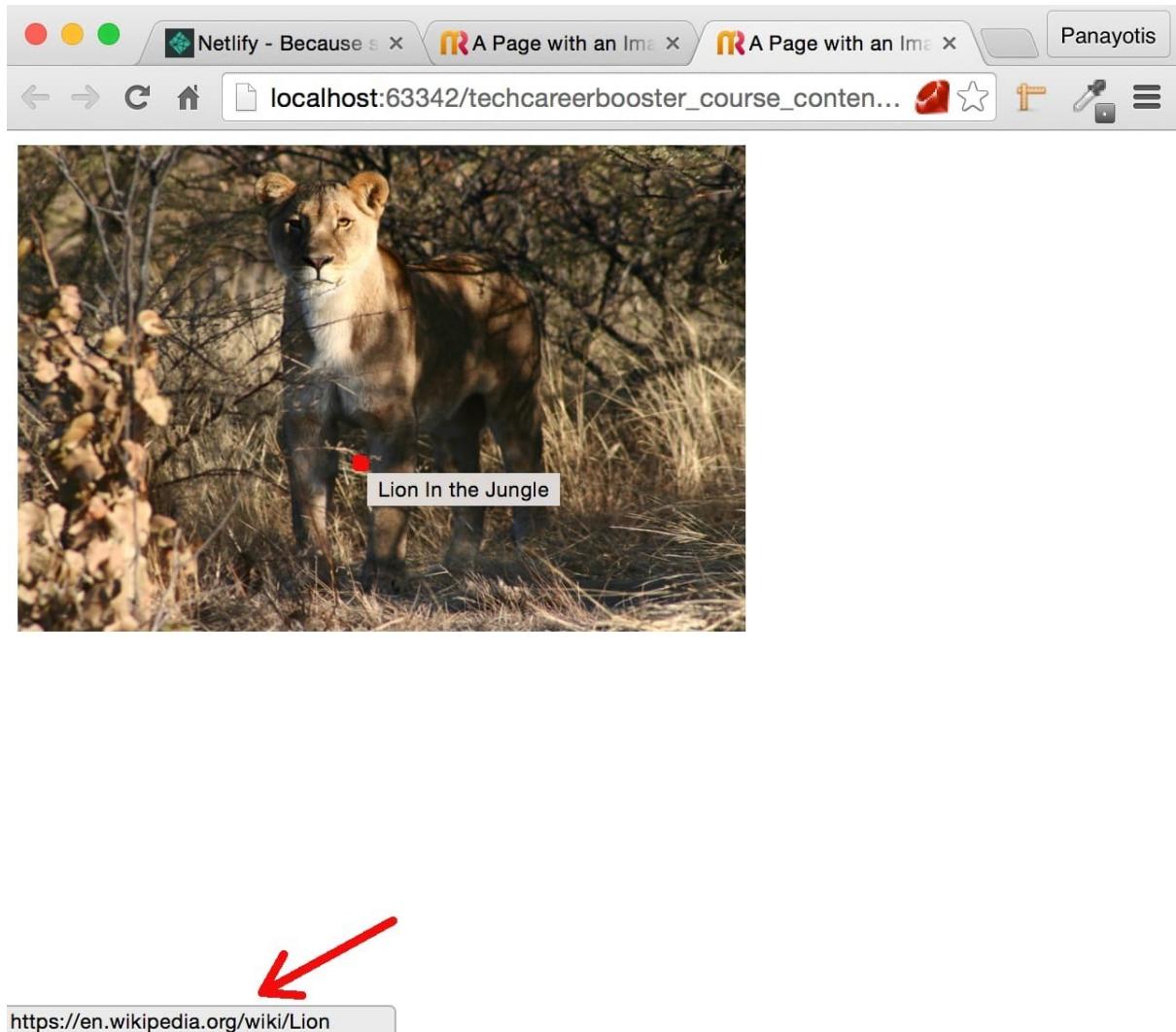


Image as a Link

As you can see from the picture above, when you move your mouse over the image, the mouse pointer changes shape to the standard anchor/link mouse shape (a hand) and the status bar of your browser displays the target Web address.

Text Around Images

We have already said that `` tag is an inline element. Text before the tag and text after the tag are placed in adjacent positions. But images are big if compared to text left and right to them. How can we make the text be placed naturally and nice next to images?

Let's try to implement the page that you see here:

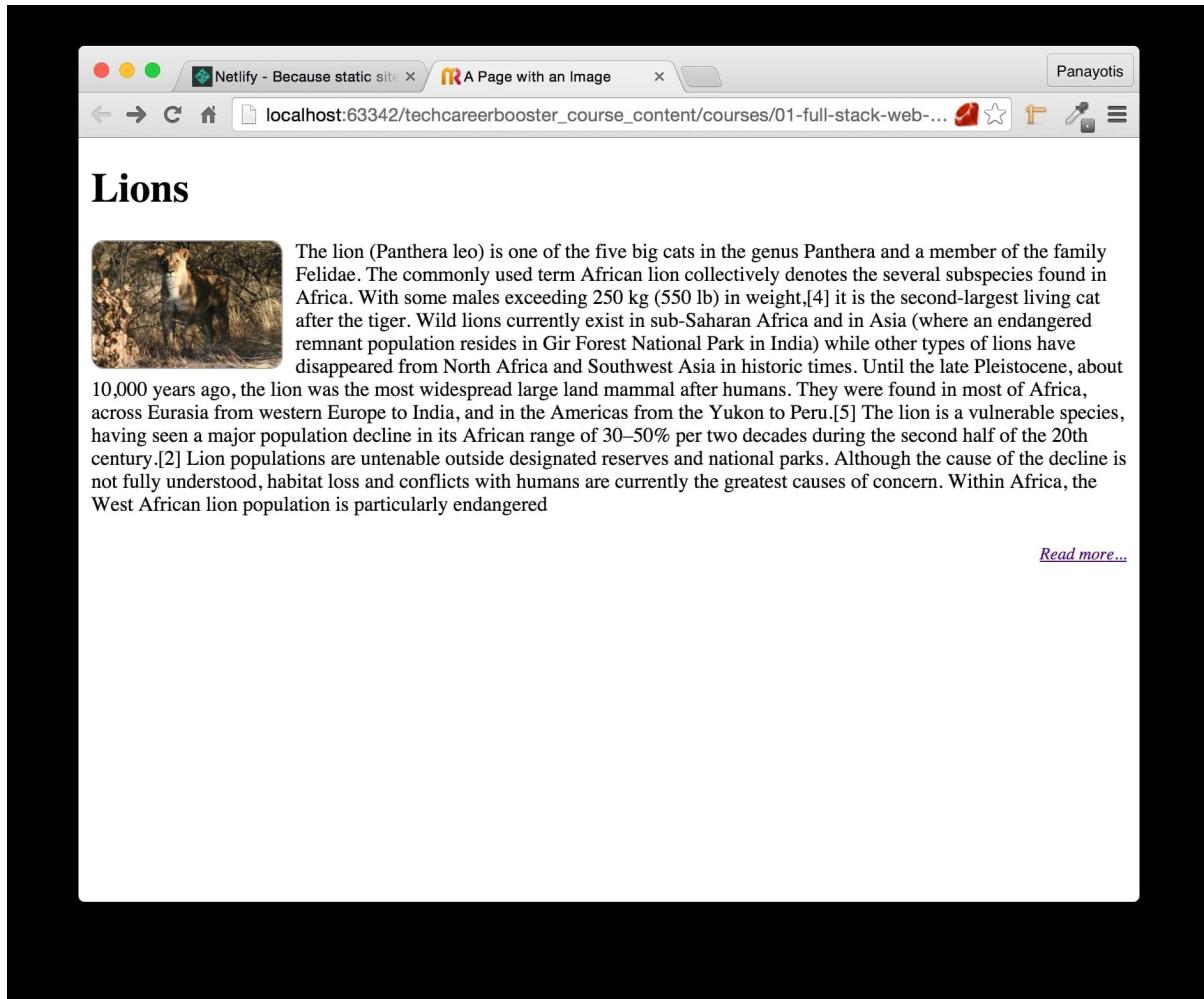


Image and Text Around It

The important things that you need to pay attention to are:

1. The image stands on the left and the text goes around it.
2. The image has a border around it with dark gray color.
3. The image border is rounding at the borders.

Also, we will implement the next things on this page:

1. The image will be a link to the lions page on Wikipedia.
2. The "Read more..." will be a link to lions page on Wikipedia.
3. When the mouse goes over the image, we will change the image opacity / transparency so that the user perceives that it is a link and that he can click on it to visit the corresponding target appearing at the browser status bar.

This is how the page should be when we move the mouse around and click on links

Let's start. We will first put the HTML content in place:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Page with an Image</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h1>Lions</h1>
11         <p>
12             <a href="https://en.wikipedia.org/wiki/Lion">
13                 
17             </a>
18
19             The lion (Panthera leo) is one of the five big cats in the genus Panthera a\
20 nd a member of the family Felidae. The commonly used term African
21         lion collectively denotes the several subspecies found in Africa. With some\
22         males exceeding 250 kg (550 lb) in weight,[4] it is the second-largest
23         living cat after the tiger. Wild lions currently exist in sub-Saharan Afric\
24 a and in Asia (where an endangered remnant population resides in Gir Forest
25         National Park in India) while other types of lions have disappeared from No\
26 rth Africa and Southwest Asia in historic times. Until the late Pleistocene,
27         about 10,000 years ago, the lion was the most widespread large land mammal \
28 after humans. They were found in most of Africa, across Eurasia from western
29         Europe to India, and in the Americas from the Yukon to Peru.[5] The lion is\
30         a vulnerable species, having seen a major population decline in its African
31         range of 30–50% per two decades during the second half of the 20th century.\\
32 [2] Lion populations are untenable outside designated reserves and national parks\
33 .
34             Although the cause of the decline is not fully understood, habitat loss and\
35 conflicts with humans are currently the greatest causes of concern. Within Afric\
36 a,
37             the West African lion population is particularly endangered
38         </p>
39
40         <p>
41             <a href="https://en.wikipedia.org/wiki/Lion"><em><small>Read more...</small>\n\
42 ></em></a>
43         </p>
44
45     </body>
46
```

47 </html>

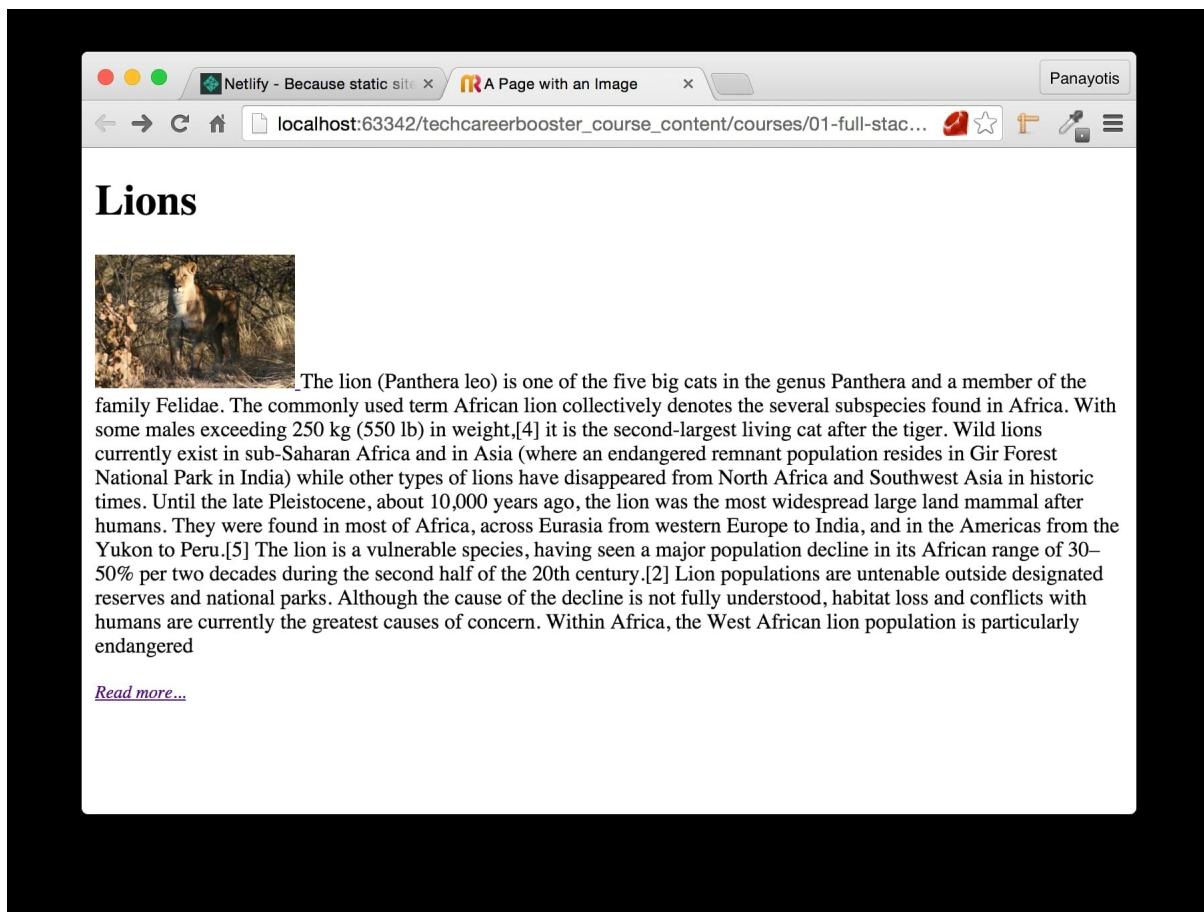
(the above code snippet online)

Create the above HTML content and save it into a file named `index.html`. Create also the file `stylesheets/main.css`. To start with, add the following rules inside it:

```
1 * {  
2     box-sizing: border-box;  
3 }  
4  
5 body {  
6     margin: 0 10px 0 10px; /* only give a left and right margin */  
7 }  
8  
9 #lion {  
10    width: 150px;  
11 }
```

(the above code snippet online)

If you save both files and load this page on a browser, you will see something like this:



Page with Lion - First Try

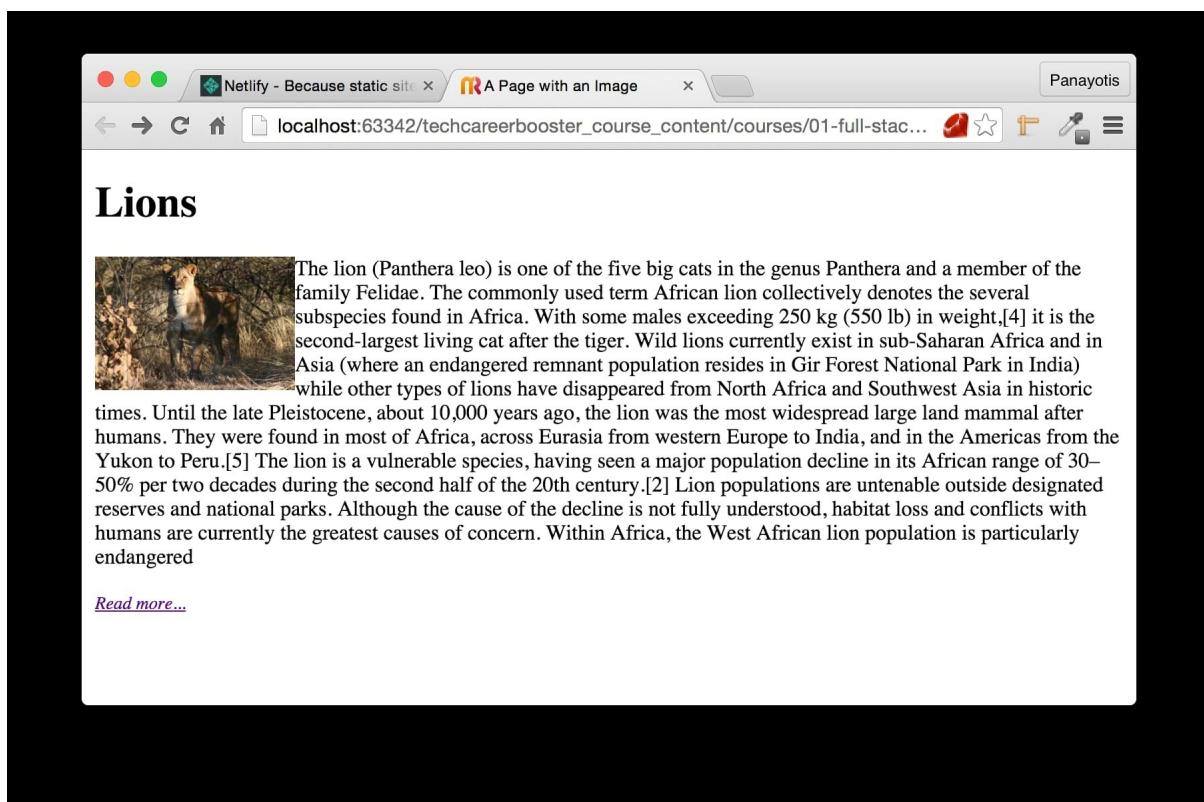
The problem above is that the text does not start at the top of the image. But it starts at the bottom. There is one easy fix to that. We only have to add the `float` property with value `left` to the image element. This will make the image float to the left, i.e. positioned to the first top left available space. Next element will flow around it. So, make the following change. On the image related CSS rule add the `float: left;` rule. So, our CSS should be:

```

1  * {
2      box-sizing: border-box;
3  }
4
5  body {
6      margin: 0 10px 0 10px; /* only give a left and right margin */
7  }
8
9  #lion {
10     width: 150px;
11     float: left;
12 }
```

(the above code snippet online)

Save and reload your page. You will see the following:



Page with Image Float Left Introduced

Much better. We are close to the final result. The image stays on the correct position and the text flows around it. Things that we need to correct here to reach the final result are:

1. Text to the right of the image should not be that close to the image boundary. We can fix that by introducing a little bit of right margin to the image.
2. The “Read more...” link should be aligned to the right. We can do that by introducing the class `text-right` with the correct rule and assign this class to the 2nd `<p>` element that contains the “Read more...” link.
3. We need to add an image border. We can do that with the `border` property.
4. We need to make the image have rounded corners. This is done with the property `border-radius`. This is a new property for you. We will give more explanations later on.
5. We need to make the image be a little bit more transparent when we move the mouse over it. This is done with setting the property `opacity`. Details, later on.

Here is the final CSS rule file:

```

1  * {
2      box-sizing: border-box;
3  }
4
5  body {
6      margin: 0 10px 0 10px;
7  }
8
9  #lion {
10     width: 150px;
11     float: left;
12     margin-right: 10px;
13     border: 1px solid darkgray;
14     border-radius: 10px;
15 }
16
17 /* Here is how we apply a CSS rule when the mouse goes over the element with id "\lion", which is the id
18 of our <img> element. */
19 #lion:hover {
20     opacity: 0.8;
21 }
22
23
24 .text-right {
25     text-align: right;
26 }
```

(the above code snippet online)

Don't forget to add the class `text-right` to the 2nd `<p>` element inside the HTML content.

Save all the files (`index.html` and `stylesheets/main.css`) and reload your page. You will see exactly what we had as our final target. Move the mouse over the image. You will see it losing a

little bit of its opacity (hence becoming a little bit more transparent). Also, you will see the dark gray border around the image and its border corners being rounded.

Let's give some more details for the new CSS rules and properties that we have introduced above:

1. `#lion:hover`. This is the CSS selector that applies CSS rules when we move the mouse over the element with id `lion`, a.k.a. over the image in our HTML page.
2. `opacity`. Takes values from `0.0` up to `1.0`. `0.0` means without any opacity, hence being completely transparent and `1.0` with full opacity, hence being without any transparency. When an image starts being transparent, that means that it reveals the colors that are behind it coming in forward. In our example, the image is placed on top of a white page, i.e. when you increase its transparency you see the colors becoming more white. Try to set this property to various values from `0.0` up to `1.0` and play with that in order to understand how it works. **Note:** If you want to support IE 8 or earlier, you may need to also add the property `filter` with value `alpha(opacity=80)`. In that case, your CSS rule would have been:

```
1  #lion:hover {  
2      opacity: 0.8;  
3      filter: alpha(opacity=80); /* support of opacity for IE8 or earlier */  
4 }
```

`opacity`, when used in `filter` property, takes values from 0 to 100 and represents a percentage.

3. `border-radius`. This is used to make the border of an element having rounded corners. It is the shorthand property for `border-top-left-radius`, `border-top-right-radius`, `border-bottom-right-radius` and `border-bottom-left-radius`.
 1. When you only set one value, you actually set equal value to all of the 4 corners.
 2. When you set two values, first value is for top-left and bottom-right corners, and second value is for top-right and bottom-left corners.
 3. When you set three values, first value is for top-left, second value is for top-right and bottom-left, third value is for bottom-right.
 4. When you set four values, first value is for top-left, second value is for top-right, third value is for bottom-right and fourth value is for bottom-left.

Try, for example, to set the `border-radius` property of the `#lion` rule to be `border-radius: 30px 10px;`. If you do that, you will see the following result:

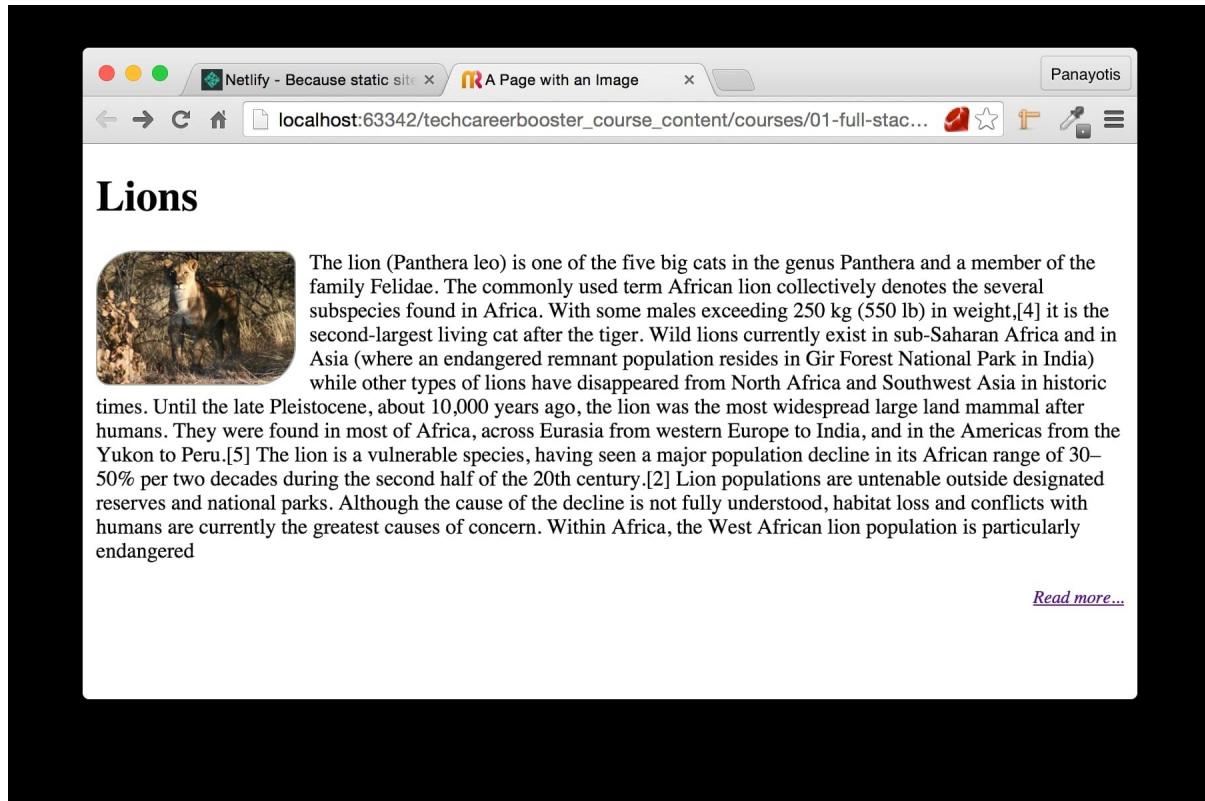


Image with Different Top Left and Bottom Right Corners

Do you see the difference between top-left and top-right corners? Top-left and bottom-right have the same corner size (30px), and they differ from top-right and bottom-left which have 10px corner size.

Images and line height

As we have said before, images are inline elements. If you put text and then an image and then some text, these will appear one next to each other, like if image was part of the text:

Write the following HTML page:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <link rel="stylesheet" href="stylesheets/main.css" type="">
6     <title>A Page with Text and Image</title>
7   </head>
8
9   <body>
10    <div>
11      This is a phrase left to an image.This is a phr
12 ase right to the image.
13
```

```
14      </div>
15  </body>
16
17 </html>
```

(the above code snippet online)

Save this file as `index.html` and then create the CSS file `stylesheets/main.css` that is referenced by this file:

```
1  * {
2      box-sizing: border-box;
3 }
4
5 body {
6     margin: 0 20px;
7     padding: 0;
8     font-size: 28px;
9 }
10
11 div {
12     background-color: #FFE4B5;
13 }
14
15 div, img {
16     margin: 0;
17     padding: 0;
18 }
```

(the above code snippet online)

This CSS file above, does nothing more than removing margins and padding that might affect our discussion here. And make the font-size a little bit larger in order to be able to see easier the line heights. We have also given a background color to the containing / parent `<div>` in order to make sure that we visually identify the space that the text and image occupy.

If you load this page on your browser, you should see something like this:

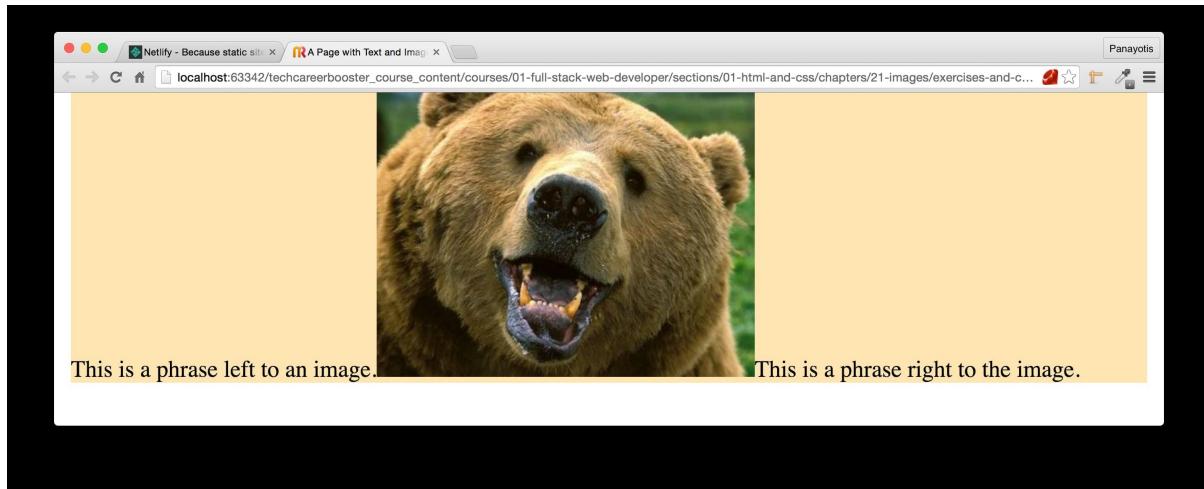
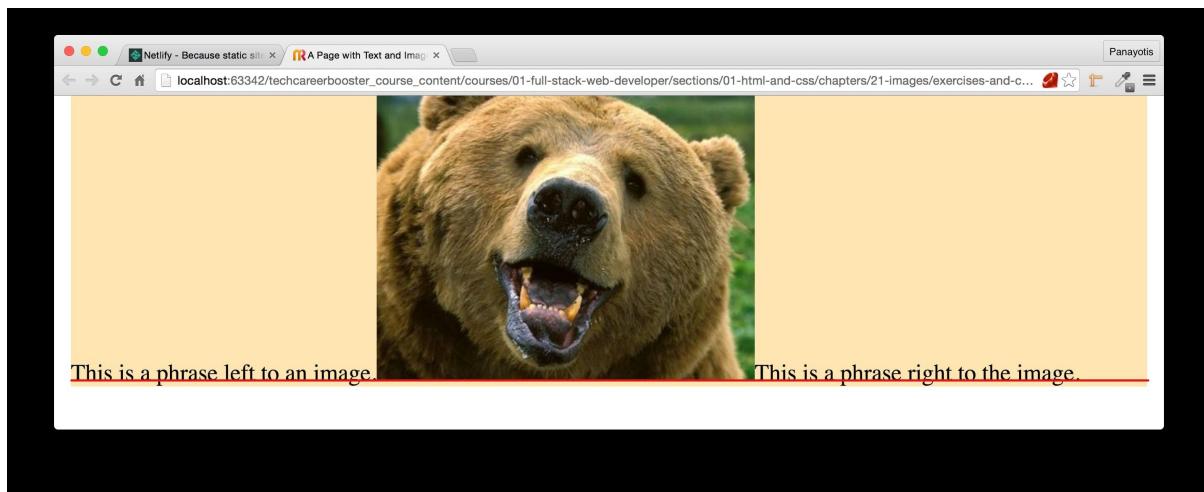


Image with some text left and right to it

If you look carefully, you will see that the image bottom side, does not align with the bottom side of the container. It aligns with the baseline of the text. There is still free space below. The free space below, spans, not only below the image, but below the whole div content (text + image). What is this space? It is the space for the descender characters. We have talked about that in [chapter 16](#). And, now, we encounter it again. But, here we want to stress the fact that the browser, treats the position of the image as if it were a character without descender portion, and places it on the same line height like the baseline of the text.

Here is the baseline marked with red color:



Baseline Marked with Red Color

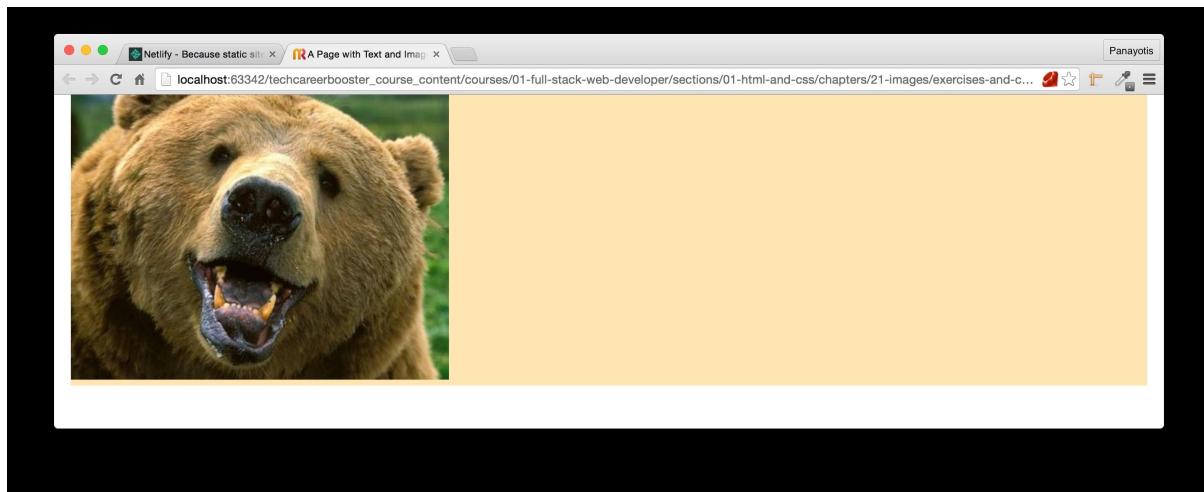
The general rule here is that images are inline elements and inline elements are positioned so that their bottom aligns with the text baseline. The text baseline is a line that is used by browser to position text and it is there **even if there is not actual text inside**. In other words, when the browser encounters inline elements, even if they are not actual text, it uses an internal calculation to calculate the **line height** that should be used to accommodate the inline elements. This line height is calculated so that the final value is enough to hold the inline elements and any characters that might appear next to them, including of course the ascenders and descenders characters.

Let's do the following experiment. On the previous page, we are going to remove the texts left and right to the image. We will only leave the image inside the div. Here we are:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <link rel="stylesheet" href="stylesheets/main.css" type="">
6          <title>A Page with Text and Image</title>
7      </head>
8
9      <body>
10         <div>
11             
13         </div>
14     </body>
15
16 </html>
```

(the above code snippet online)

Make sure you save the above, and reload it on your browser. What you will see is the following:



We have a series of solutions that we can apply. They will all work on this particular case, but some of them might not be working in another context. That's why you need to know all of them and pick up the one that best suits your needs on the particular context you encounter this "problem" in.

1. *Solution 1:* Set the `display` property of the `img` to `block`.
2. *Solution 2:* Set the `line-height` property of the containing `div` to `0`;
3. *Solution 3:* Set the `vertical-align` property of the `img` tag to `top` (sometimes `top` will not work; try also one of `bottom`, `baseline`, `middle`, `text-bottom`)
4. *Solution 4:* Set the `font-size` of the `div` to `0`.

For example, try the first solution:

```
1  * {
2      box-sizing: border-box;
3  }
4
5  body {
6      margin: 0 20px;
7      padding: 0;
8      font-size: 28px;
9  }
10
11 div {
12     background-color: #FFE4B5;
13 }
14
15 div, img {
16     margin: 0;
17     padding: 0;
18 }
19
20 img {
21     display: block;
22 }
```

(the above code snippet online)

If you save this and reload your page, you should see the following:

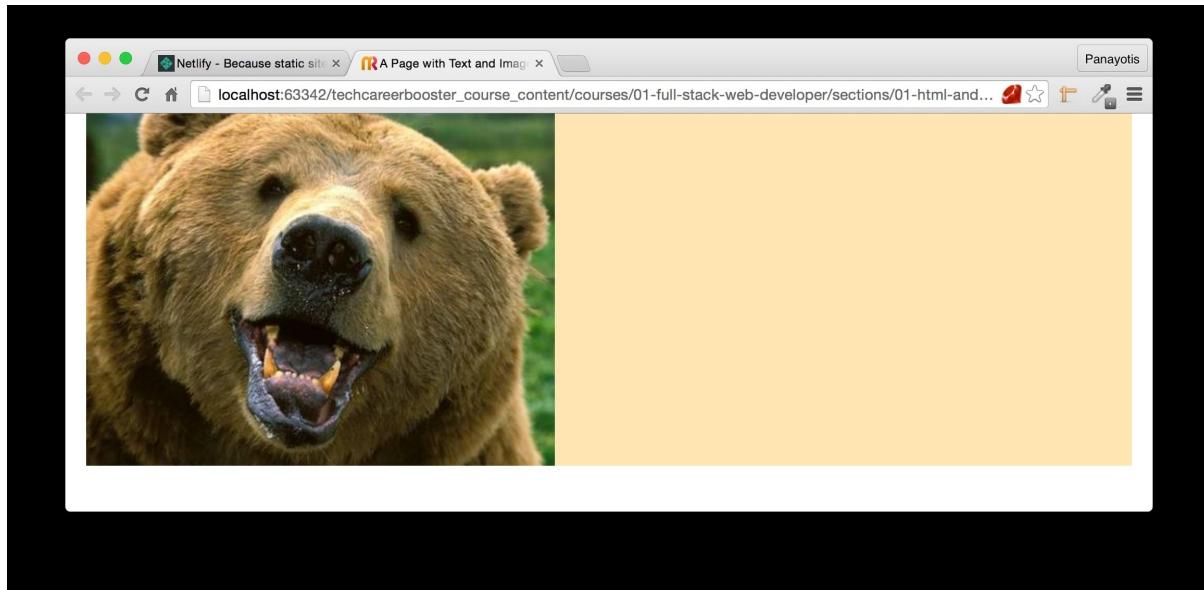


Image Inside a Div without Blank Space Below

Problem solved! The image bottom is aligned to the div bottom.

Using images stored on our site

In the previous examples, we referenced some images that existed in remote servers, not from our site. If we want to use images from our site, we first need to save them locally, where the pages of our site are.

Usually, we save the images of our site inside a folder named “images”. Let’s do that.

Download the next two images:

1. <https://dvirfixler.files.wordpress.com/2014/03/stunning-photography-of-wild-animals-11.jpg>
2. <http://images6.fanpop.com/image/photos/34300000/Big-Cats-wild-animals-34365429-900-599.jpg>

and save them locally inside the folder “images” in the root of your site. Name the first image “animal1.jpg” and the second image “animal2.jpg”. In other words, you need to have something like that in your file system:

```
1 - [images]
2   |
3     |- animal1.jpg
4     |- animal2.jpg
```

(the above code snippet online)

Next create the HTML file with the following content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Page with Local Images</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <h1 class="text-center">Wild Animals</h1>
12
13         <div id="left-column" class="column">
14             
15         </div>
16
17         <div id="right-column" class="column">
18             
19         </div>
20
21     </body>
22
23 </html>
```

(the above code snippet online)

Save this file with the name `index.html`.

Then create the CSS file with the following content:

```
1  /* Use border-box model for box sizing. Hence, width and height take into account \
2   padding and borders. */
3  * {
4      box-sizing: border-box;
5  }
6
7  /* Remove any body margin */
8  body {
9      margin: 0;
10 }
11
12 /* Make all images of the same size */
13 img {
14     width: 420px;
15     height: 350px;
16 }
```

```
18 /* Make all columns of the same size, 50% of their container */
19 .column {
20   width: 50%;
21 }
22
23 /* Left column should float left but with text alignment right */
24 #left-column {
25   float: left;
26   text-align: right;
27 }
28
29 /* Right column should float left, with text alignment left - which is the default */
30 t */
31 #right-column {
32   float: left;
33 }
34
35 /* Helper class to center text */
36 .text-center {
37   text-align: center;
38 }
```

(the above code snippet online)

Save this content to file `stylesheets/main.css`.

Before proceeding, read carefully the CSS comments and make sure you understand the CSS rules that we are using.

Hence, your files and folders should be like this:

```
1 - [images]
2   |
3   |- animal1.jpg
4   |- animal2.jpg
5 - [stylesheets]
6   |
7   |- main.css
8 - index.html
```

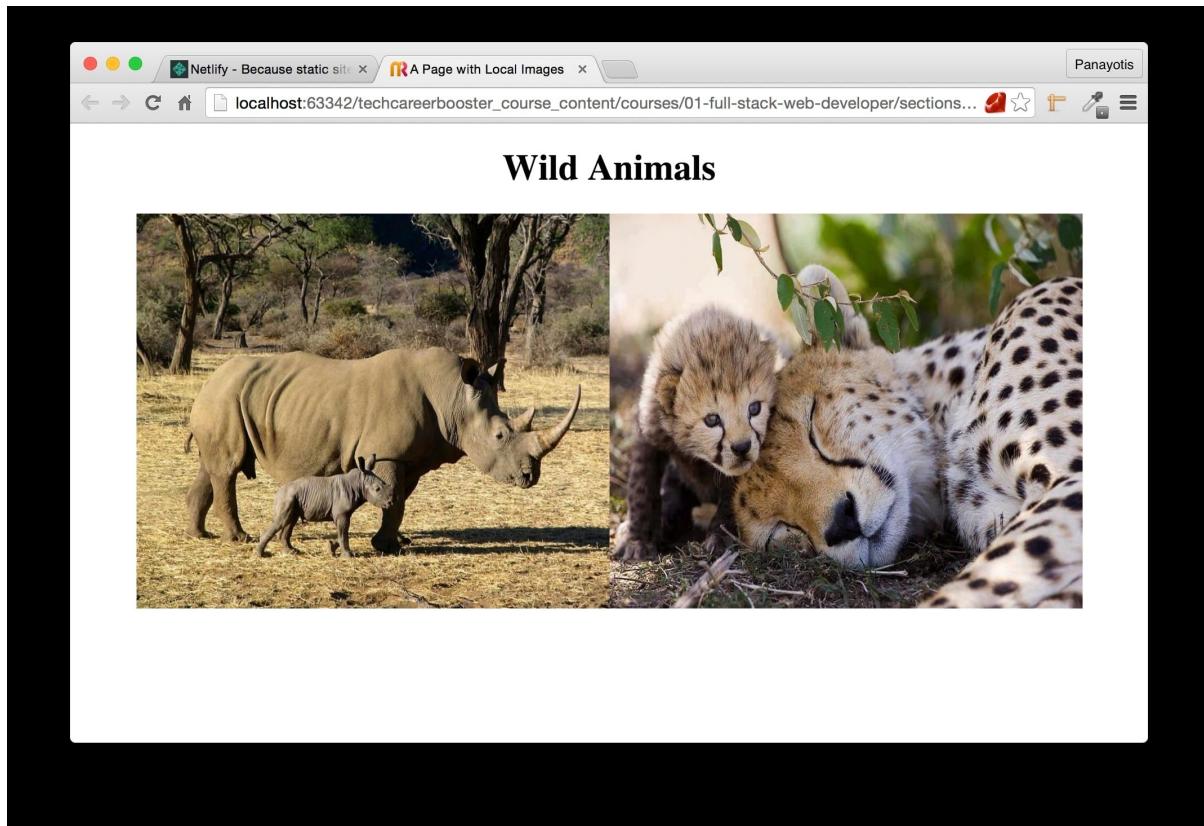
(the above code snippet online)

Do you see how the HTML page content references the images that reside on the same site, inside the folder “images”?

```
1 ...
2 
3 ...
4 
```

(the above code snippet online)

Load the `index.html` page on your browser. You should see something like this:



Two Images Locally Stored On Site

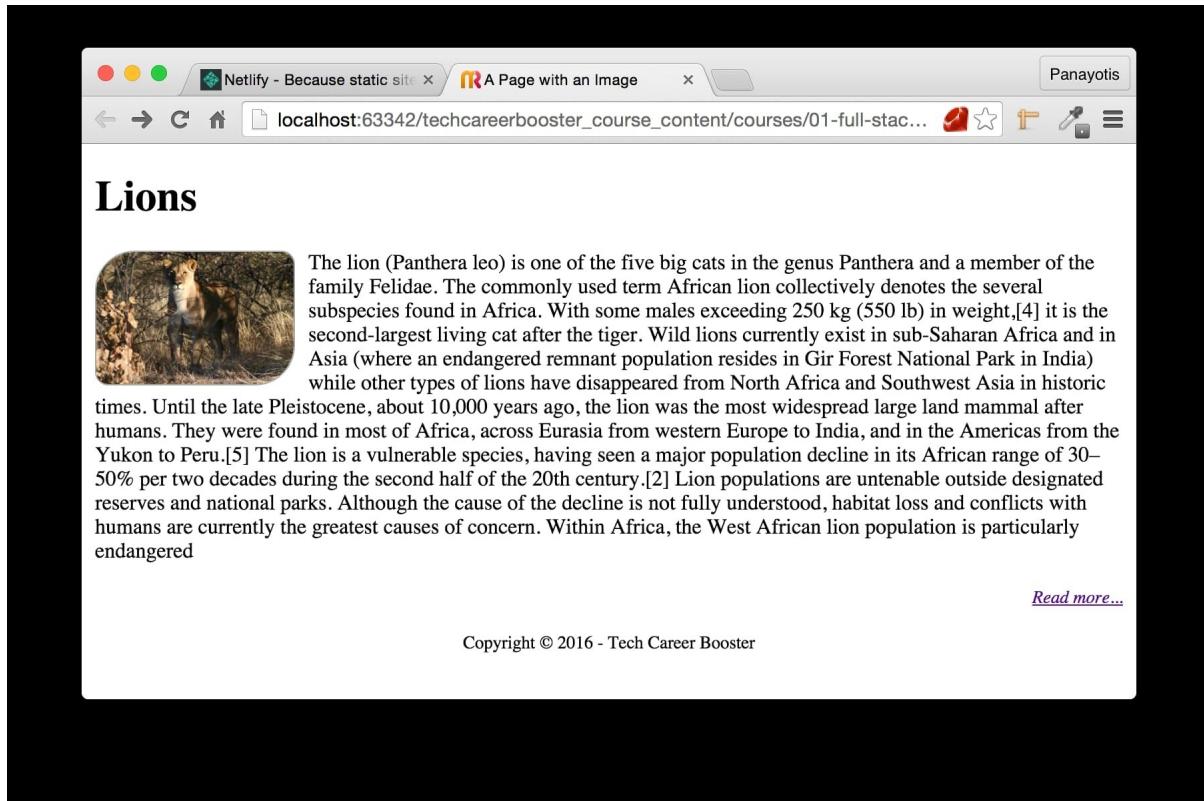
Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

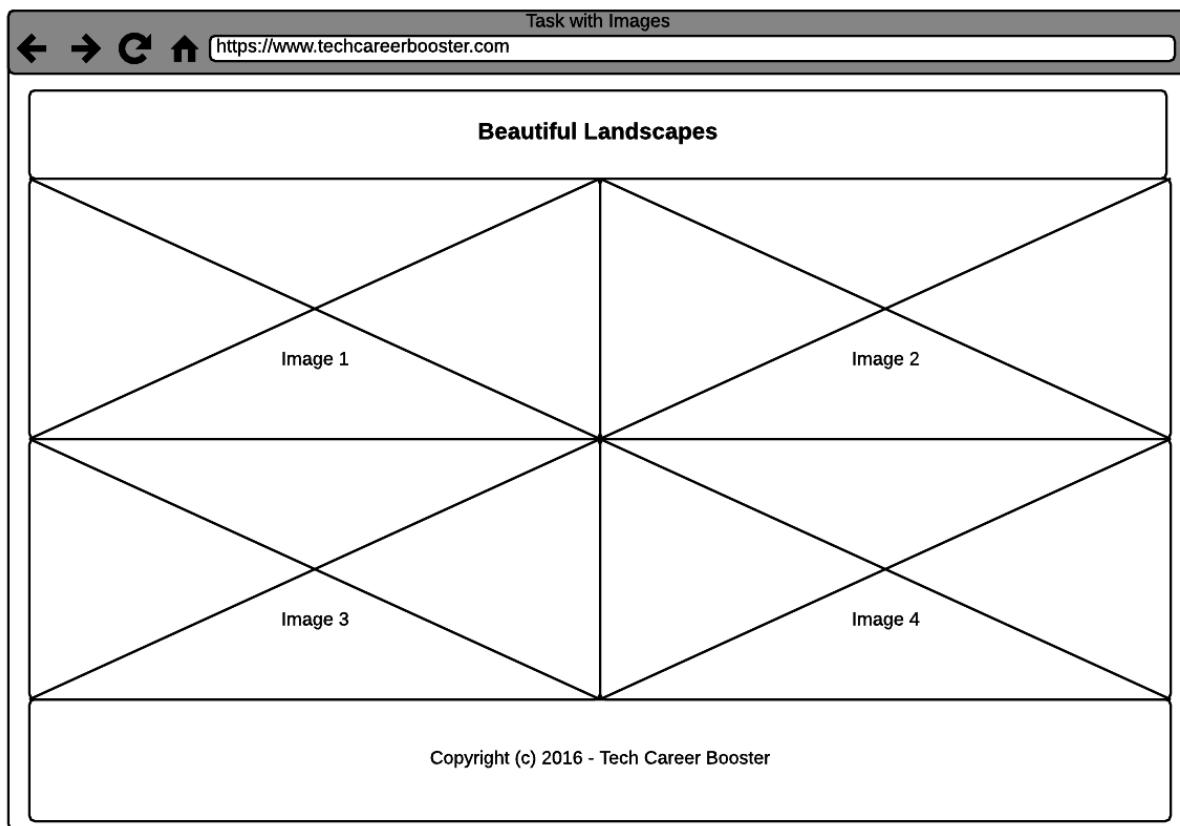
Task details:

1. Add a copyright notice at the bottom of the site that we have built in this chapter. This should be the final result (Use your own name instead of "Tech Career Booster"):



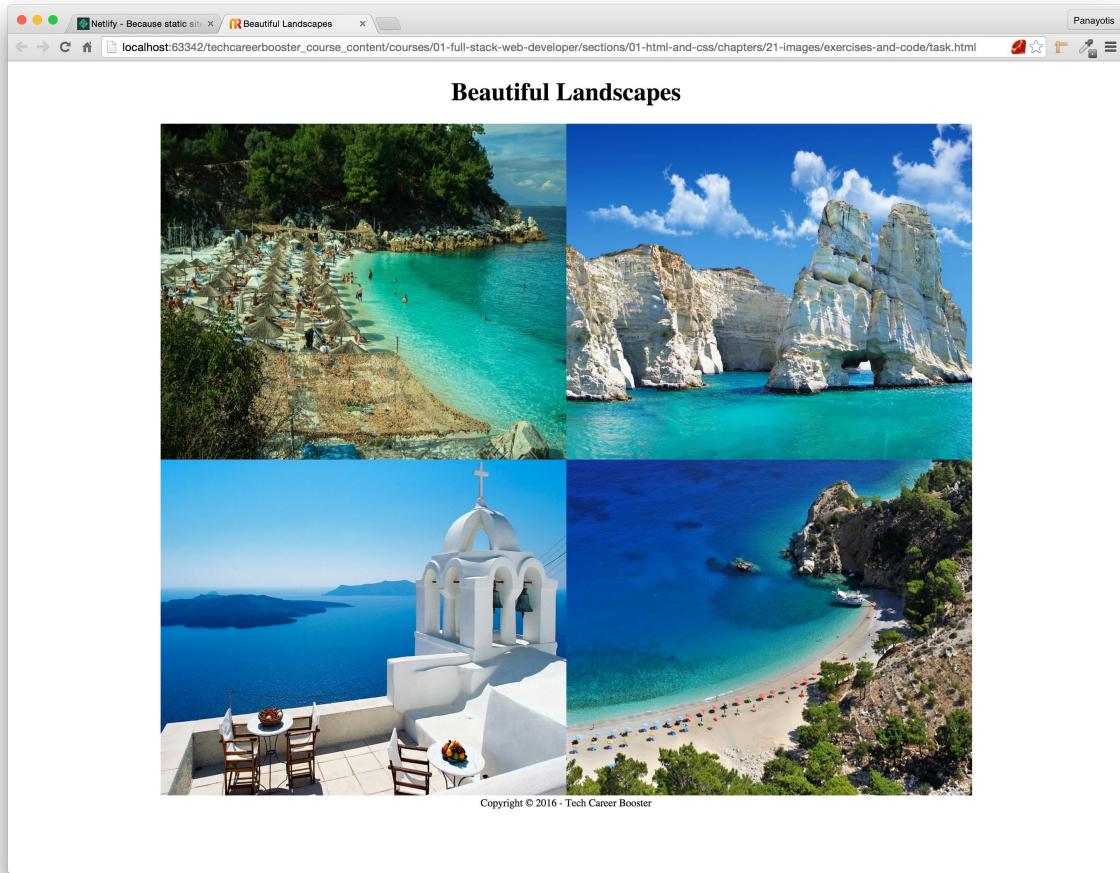
Page With Copyright Notice

2. Upload this site to your Netlify account.
3. Create the following web site and upload to your Netlify account.
 1. Here, we are presenting a mock up.



Images Task - Site with 4 Images

2. You should implement a site that satisfies the above mock up.
3. Make sure that when the user moves his mouse over any of the images, the image opacity is decreased by 30%.
4. Use any list of images that you want.
5. The images should be downloaded and stored locally on your site. This is a sample implementation:



Task With Images

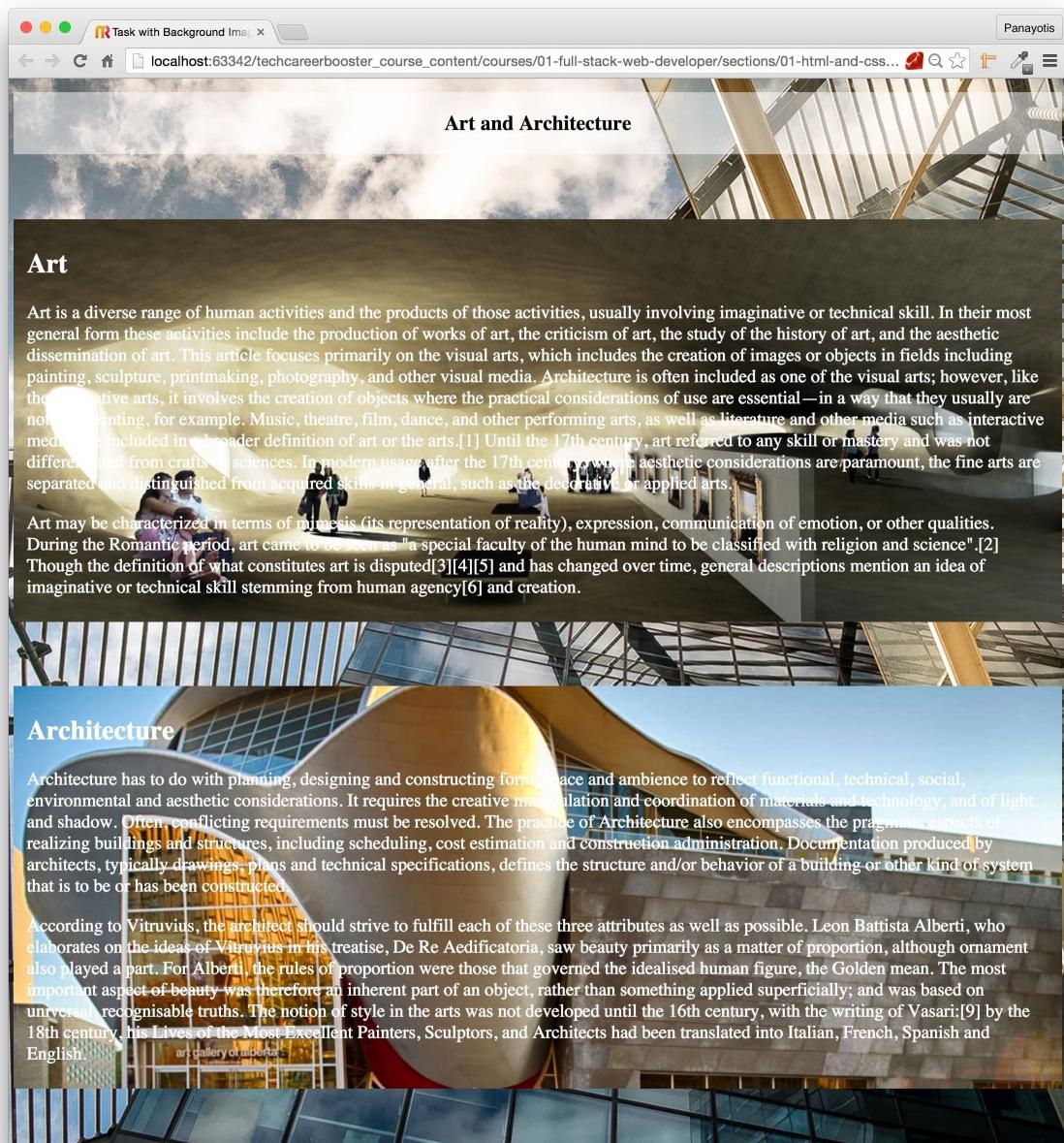
6. Images width and height should be fixed and the same for all images.
7. The images should be links to the google map point of the landscape that you have selected. For example, this is the link "<https://www.google.gr/maps/place/Thasos/@40.6846043,24.578337,10z>" for the first image (top-left) of the sample above.

Chapter 22 - Image Backgrounds

Summary

In this chapter, you will learn how you can use images as background of your page and parts of it.

You will build sites like this:



Site With Background Images

Learning Goals

1. Learn to specify the background of the body element.
2. Learn about the background image related CSS properties:
 1. background-color
 2. background-image
 3. background-repeat
 4. background-attachment
 5. background-position
 6. background-size
3. Learn how to have a background image for the whole page of your site.
4. Learn about how large images are better for the background image of your site.
5. Use a background image for parts of your page.
6. Use background color for sections of your page.
7. Make sure the background image stays still and does not scroll with the page content scrolling.
8. Learn how to be DRY in your code and do not reuse with copy / paste.
9. Learn about the background shorthand property.

Image Backgrounds

We have learned how to introduce images to our site, turn them to links and possibly put text around them. However, there are cases in which we want the image to be used as a background. Look at this site here for example:

The screenshot shows the homepage of pamediaekopes.gr. The background features a scenic view of traditional windmills on a rocky island with a castle in the distance. Overlaid on this background is a dark blue search bar with white text. The search bar contains fields for 'From' (London, Heathrow (LHR)), 'To' (Lisbon, Lisboa (LIS)), 'Departure - Return' (Mon 01/02 - Fri 19/02), 'Passengers' (1 Adult), and a green 'Search' button. Below the search bar, there's a section titled 'Best offers for flights from London in the coming weeks' with four cards: Athens (From £107,65), Lisbon (From £112,37), Prague (From £112,88), and Iasi (From £176,19). The website has a navigation bar with links for FLIGHTS, HOTELS, FERRIES, and CARS.

A Background Image Example

As you can see in the above picture (you can [visit the real site too if you click here](#)), there is an image that has other elements covering parts of its area. Hence, the image is on the background, whereas the elements are on the foreground.

Let's start little by little, with the basics.

Image as a body background

We are going to use this image here:



Small Dog Image

to cover the whole page on our site. Note, however, that the image is of fixed size, and this, in particular, is a small image. It has width 300px and height 400px.

When you want to cover the whole page, window, with an image, one first idea might be to use the image as a background of the body element.

Let's do that:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Image as a body background</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9
10     </body>
11 </html>

```

(the above code snippet online)

A very simple web page. Actually, it does not have content on the body. However, it does use a stylesheet file. Create the following CSS file:

```

1 body {
2     background-image: url("../images/cute-medium-sized-dog.jpg");
3 }

```

(the above code snippet online)

As you can see, the CSS is very simple. It is using the `background-image` CSS property to specify the image that should be used for background cover on the particular element, i.e. the `body` element.

Important: If you haven't done that already, download the image `cute-medium-sized-dog.jpg` from the images of the current chapter. Save the image inside the folder "images" on the root folder of your project.

One of the values that the `background-image` property takes is the call to `url` function with argument the path to the image file. Note that we are using a relative path. If you see the file structure:

```

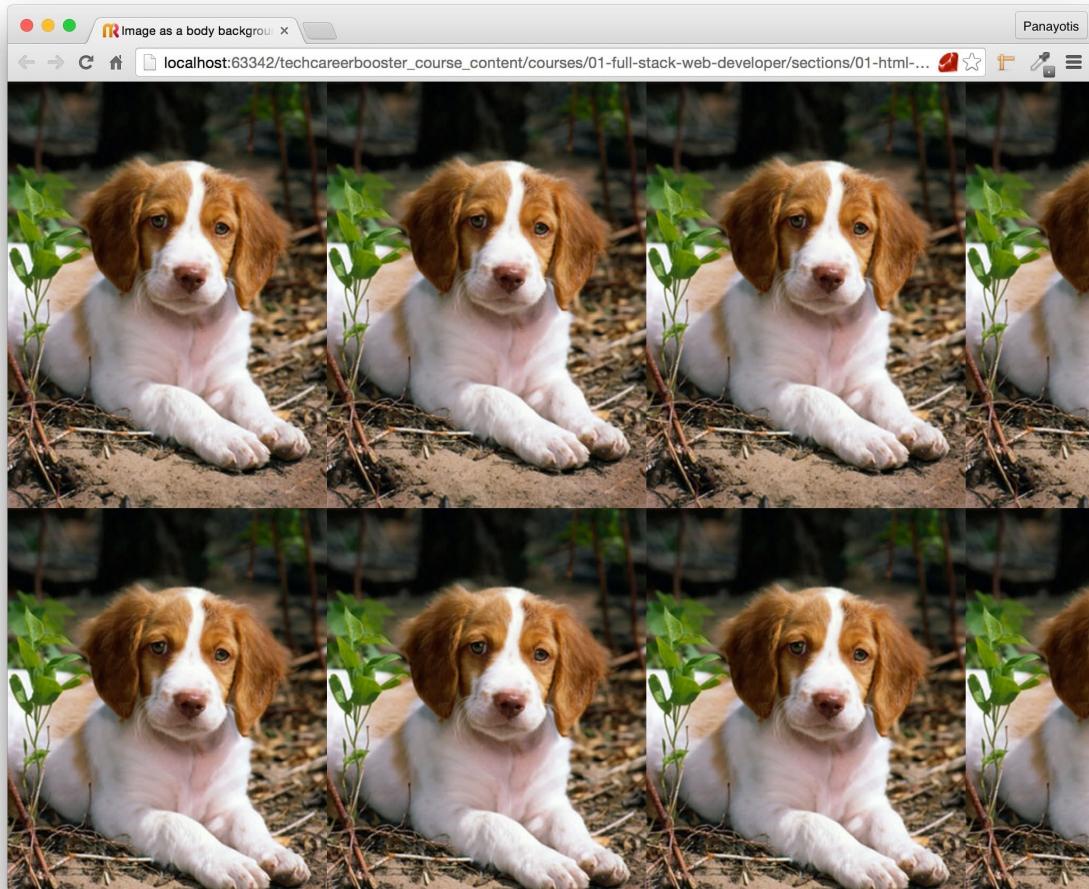
1 - [images]
2   |
3   | - cute-medium-sized-dog.jpg
4 - [stylesheets]
5   |
6   | - main.css
7 - index.html

```

(the above code snippet online)

you will see that in order for the `main.css` file to reference a file inside the `images` dir, needs to travel one directory up (`..`) and then go inside `images` directory.

If you correctly create the above files and then load the `index.html` page on your browser, you will see the following result:



Page with Body Background

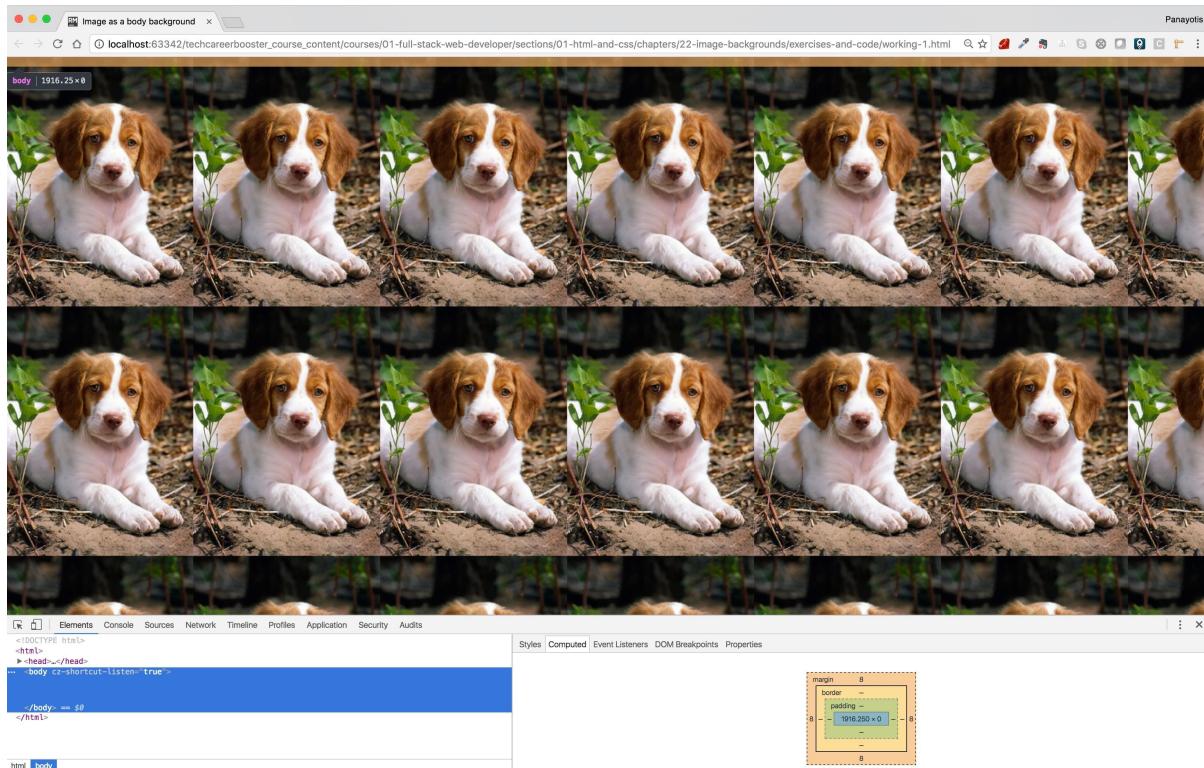
As you can see, due to the fact that the image is small, it is repeated horizontally and vertically, in order to cover the whole body area.

Let's Be More Accurate

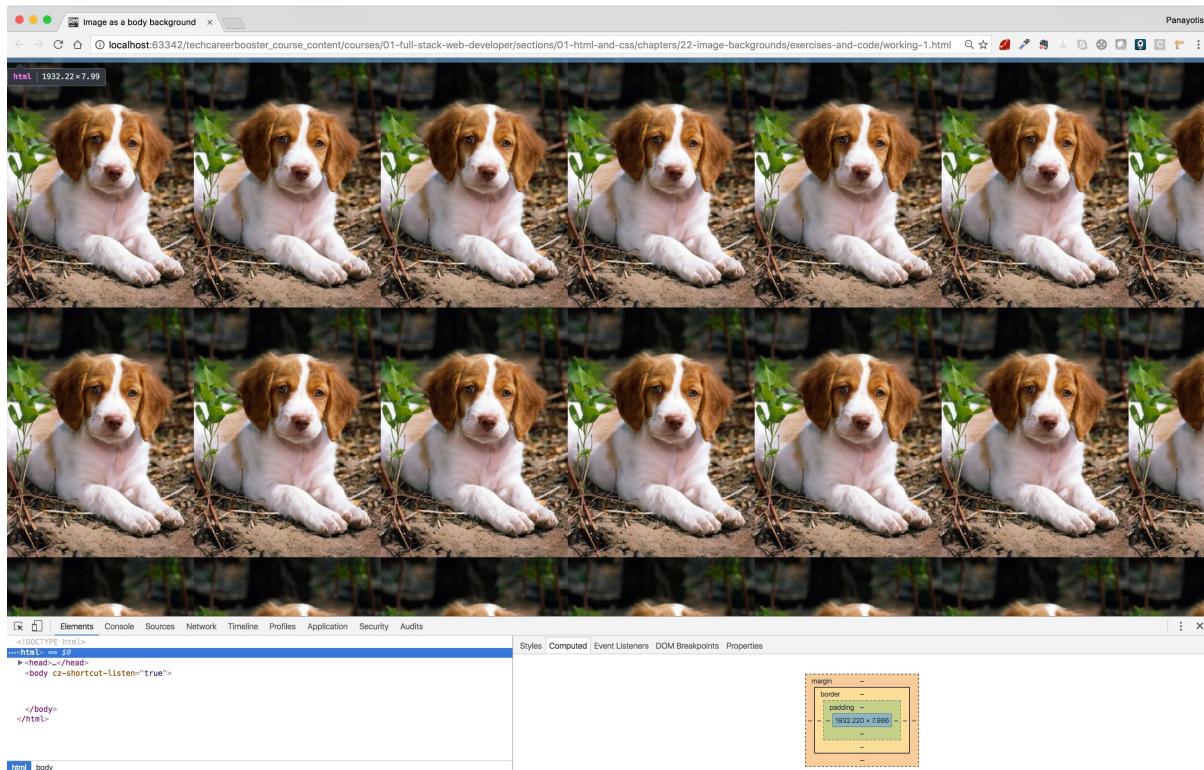
In the “CSS Box Model” chapter we have learned that the `body` background properties propagate to `html`, if `html` does not specify its own background properties. And actually, from `html` it propagates to `canvas`, to the whole browser drawing area.

This is what has happened for our page above too.

If you inspect, with developer tools, the `body` element, you will see that it does not have a height:

**Body Has 0 Height**

And the `html` has a very small height:

**Html Has Small Height**

However, you can see that the small image runs from left to right and top to bottom and covers

the whole drawing area, the canvas area.

Do we want to have the background properties propagate to the parent element and then to canvas? We could work with that, but maybe it is a good idea if we could stop the background properties from propagation and have them work on the body element only. One reason this might be good for is in order to have more accurate control on the styling of our page.

How To Stop Propagation

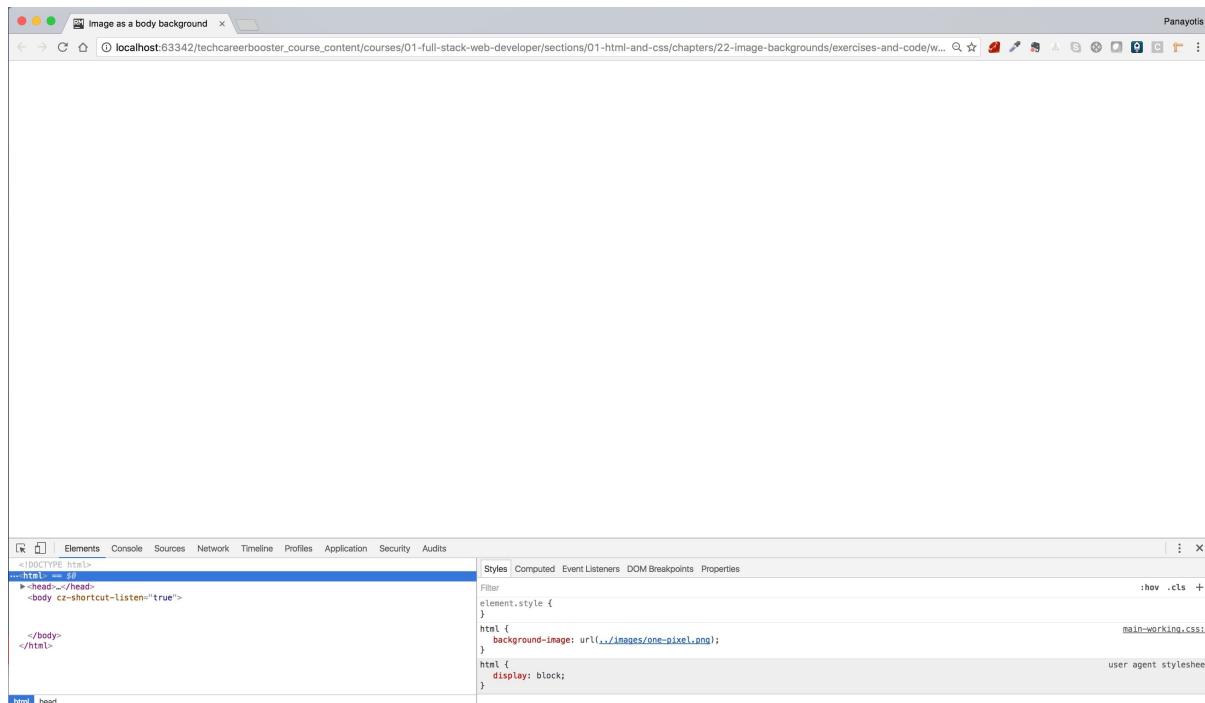
How can we stop propagation? We have to make sure that `html` has its own background properties. We can use, for example a one pixel image: Try the following CSS for your page:

```

1 html {
2     background-image: url('..../images/one-pixel.png');
3 }
4
5 body {
6     background-image: url("..../images/cute-medium-sized-dog.jpg");
7 }
```

(the above code snippet online)

If you reload your page on your browser, you will see this:



HTML Element With One Pixel Background Image

What we have done was to set the `background-image` property on the `html` to a 1 pixel white image. Alternatively, we could have used the `background-color` property. Let's do that:

```
1 html {  
2     background-color: White;  
3 }  
4  
5 body {  
6     background-image: url("../images/cute-medium-sized-dog.jpg");  
7 }
```

(the above code snippet online)

Body Without Content is Not Displayed

Why do we see the page now being white and we don't see the body background image? Because, body does not have any content. Hence, its height is 0, and, consequently, the background image is not drawn.

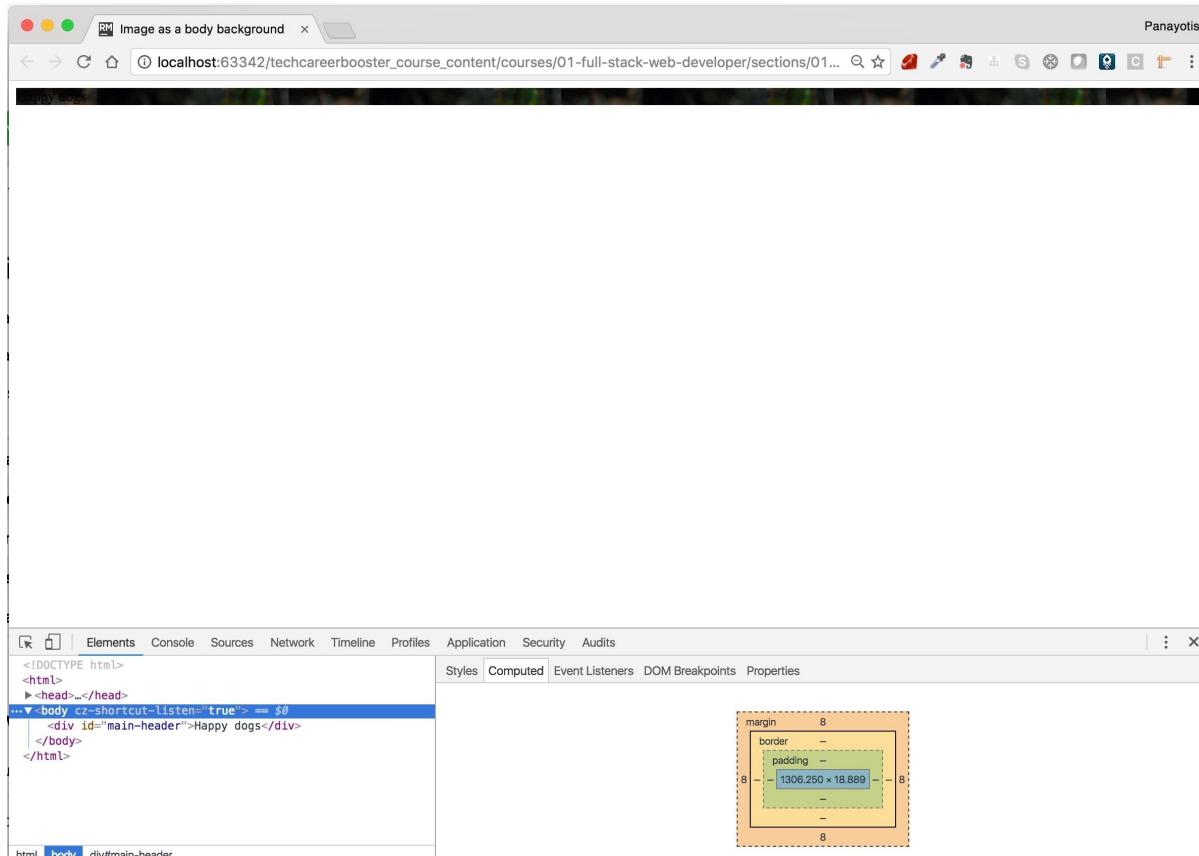
Adding Content to Body

We are now going to add some content to body so that browser draws it clearly on page. Let's change the HTML of our page to:

```
1 <!DOCTYPE html>  
2 <html>  
3     <head>  
4         <meta charset="utf-8">  
5         <title>Image as a body background</title>  
6         <link rel="stylesheet" href="stylesheets/main.css" type="text/css">  
7     </head>  
8     <body>  
9         <div id="main-header">Happy dogs</div>  
10    </body>  
11 </html>
```

(the above code snippet online)

If you save the above and reload the page on your browser, you will see something like this:



Body is Now Visible

The body has enough height to draw its content, the new div with id `main-header`. But, of course, this is not enough to clearly display the whole image background.

Make body High Enough

Which means that in order for the image background to cover the whole area again, under the control of the body element, we need to make the body have enough `height`.

Let's make our CSS as follows:

```

1 html {
2   height: 100%;
3   background-color: white;
4 }
5
6 body {
7   min-height: 100%;
8   background-image: url("../images/cute-medium-sized-dog.jpg");
9 }
```

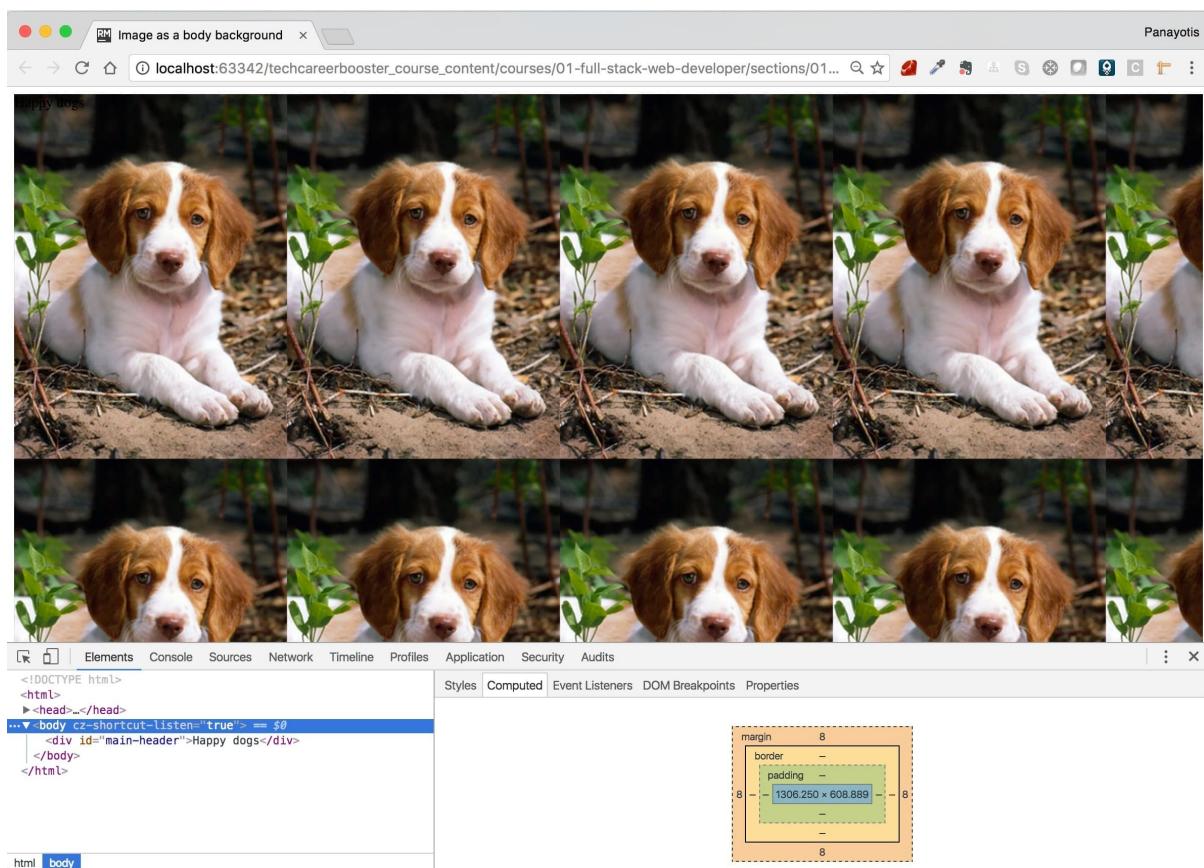
(the above code snippet online)

What we did? We set the height for `html` to 100% so that it covers the whole canvas area. And we set the body `min-height` to be 100% too. These two make the body height be at least equal to the height of the `html` element.

Question: Why do I have to set `min-height` to body and not `height`? It is because the `min-height` will make sure that the background properties work ok, in the case in which the body content is long enough to require the user to scroll down in order to see the whole content. Visit these two pages here and scroll to see the difference.

1. Page with body `min-height` 100%
2. Page with body `height` 100%

Having done that, if we reload the page on our browser, we will see this:



Body With Full Height Background Image is Now Visible

Remove Margin

But, there is a small detail we need to take care of. The body has the default margin. You can see that 8 pixels margin around it. Let's remove that.

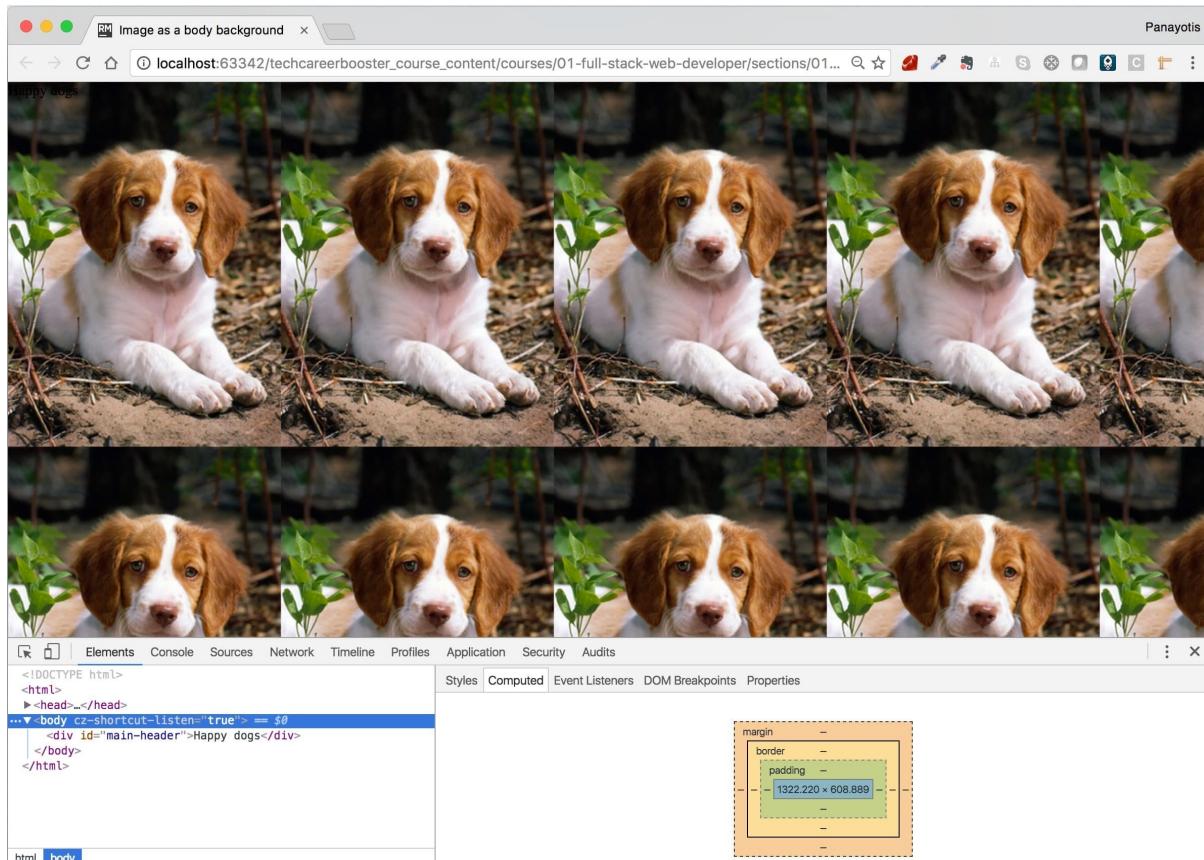
```

1 html {
2     height: 100%;
3     background-color: white;
4 }
5
6 body {
7     min-height: 100%;
8     background-image: url("../images/cute-medium-sized-dog.jpg");
9     margin: 0;
10}

```

(the above code snippet online)

If you save and reload the page on your browser, you will get this:



Body Margin 0 And Background Image

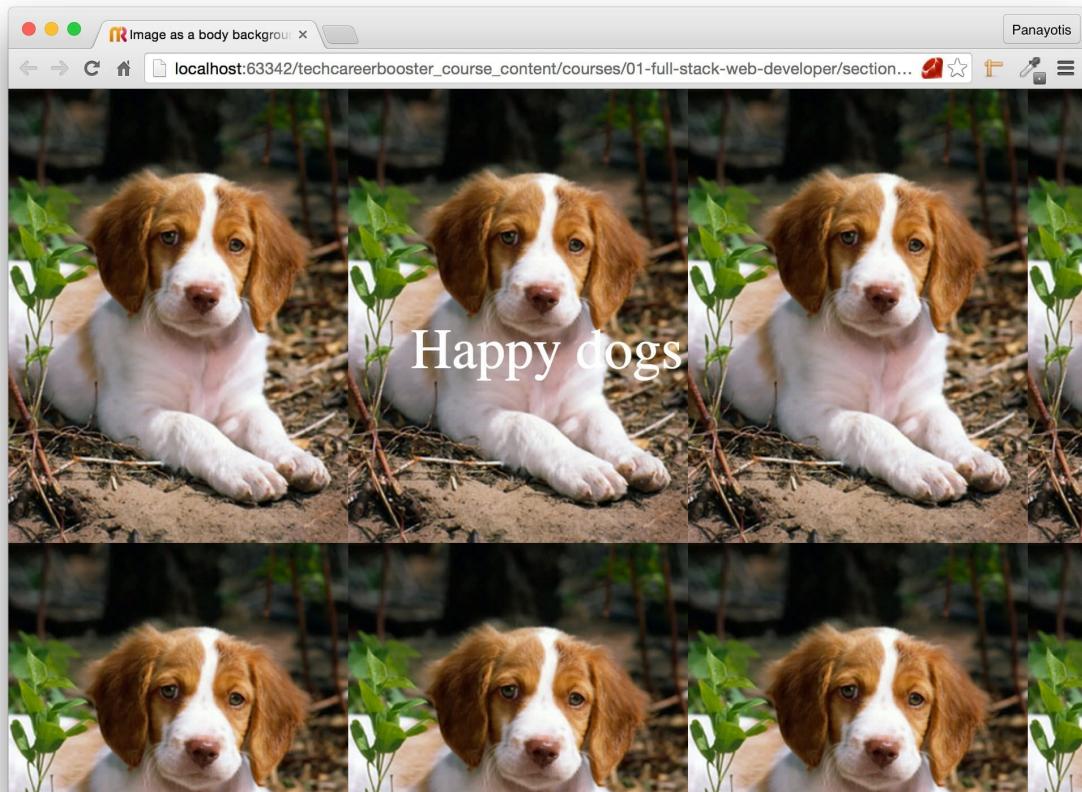
Where is The Happy Dogs Text?

It's black and you cannot easily see them. Let's change the CSS to have them be easier to see:

```
1 html {  
2     height: 100%;  
3     background-color: White;  
4 }  
5 body {  
6     min-height: 100%;  
7     background-image: url("../images/cute-medium-sized-dog.jpg");  
8     margin: 0;  
9 }  
10  
11 #main-header {  
12     color: White;  
13     font-size: 50px;  
14     text-align: center;  
15     padding-top: 200px;  
16 }
```

(the above code snippet online)

If you save those files and reload your `index.html` page, you will see this:



Background and Text on Top

As you can see, the “Happy Dogs” text has been put on top of the image(s). Image(s) work as a background.

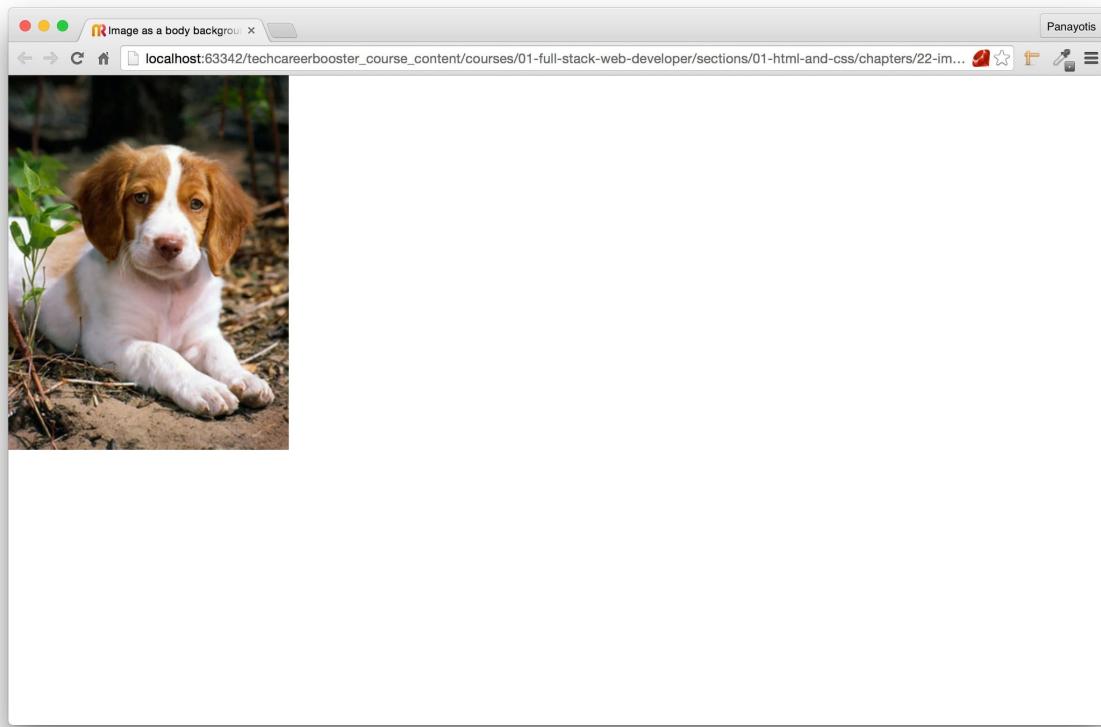
One Image to Cover Whole Page

How can we have one image only, covering the whole page. Can we have one instance of the dog image covering the whole viewport? Let's try the following change in our CSS rules:

```
1 html {  
2     height: 100%;  
3     background-color: White;  
4 }  
5 body {  
6     min-height: 100%;  
7     background-image: url("../images/cute-medium-sized-dog.jpg");  
8     background-repeat: no-repeat;  
9     margin: 0;  
10 }  
11  
12 #main-header {  
13     color: White;  
14     font-size: 50px;  
15     text-align: center;  
16     padding-top: 200px;  
17 }
```

(the above code snippet online)

We have used the property `background-repeat` with value `no-repeat`. Let's save that and reload the page. What you will see is this:

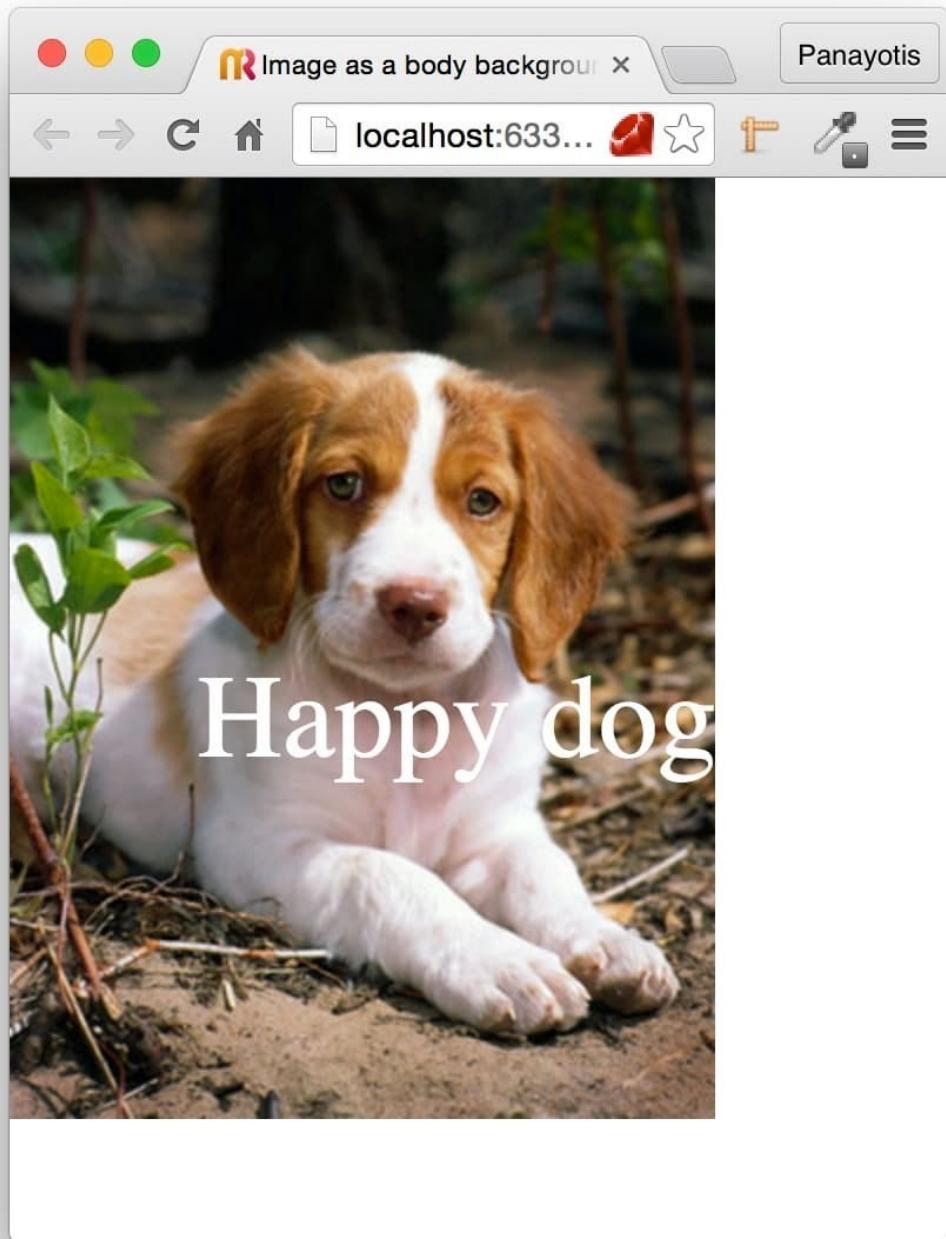


Background Image No Repeat

As you can see, the `background-repeat` with value `no-repeat` instructs the browser to not repeat the image. The property `background-repeat` takes the values:

1. `repeat`, which is the default value and repeats the image both horizontally and vertically.
2. `repeat-x`, which repeats the image horizontally.
3. `repeat-y`, which repeats the image vertically.
4. `no-repeat`, which does not repeat the image.

Ok. Cool. We have done 1 step to have only one instance of the image. But hold on. Where is the text. It's there, but you cannot see that because it is white and it is on the center of the page. If you shrink your browser window, you will see it.



Background Image No Repeat Shrunk Window

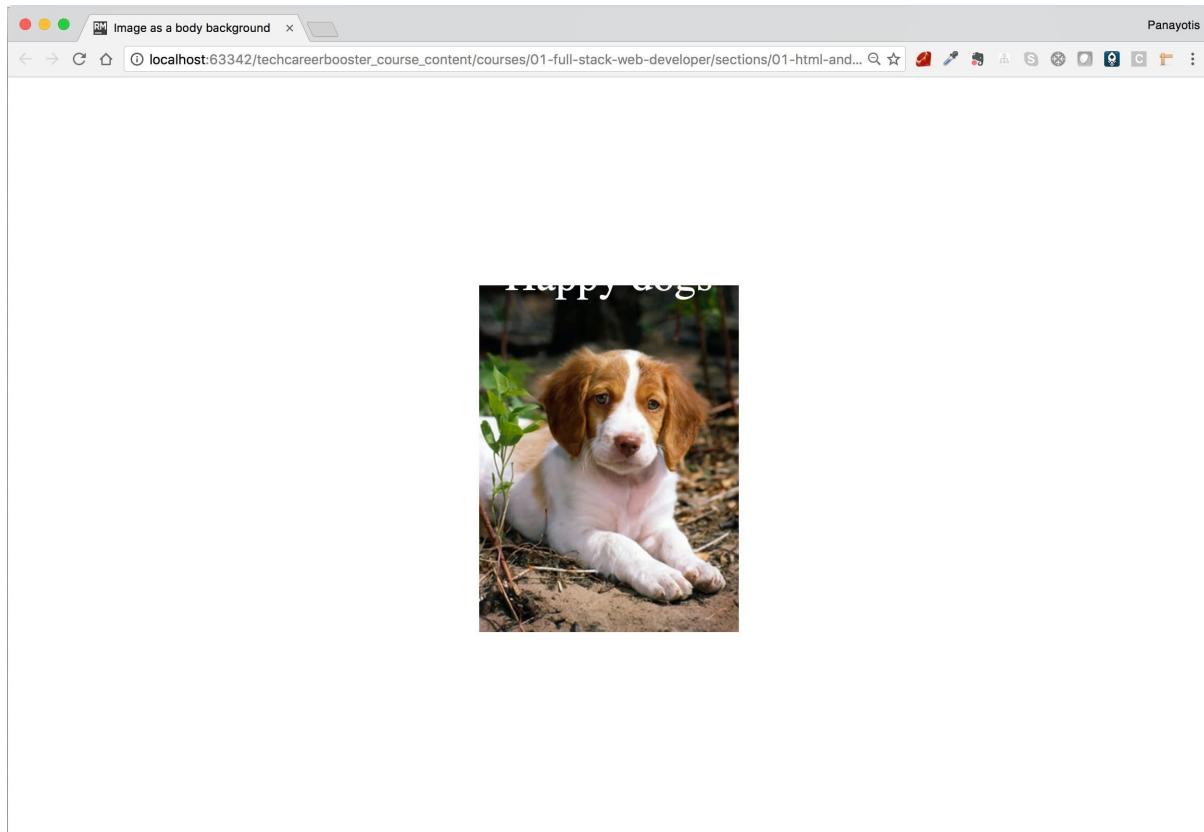
Center Image

The next step is to ask the browser to bring the image in the center of the window. Let's amend our CSS as follows:

```
1 html {  
2     height: 100%;  
3     background-color: White;  
4 }  
5 body {  
6     min-height: 100%;  
7     background-image: url("../images/cute-medium-sized-dog.jpg");  
8     background-repeat: no-repeat;  
9     background-position: center;  
10    margin: 0;  
11 }  
12  
13 #main-header {  
14     color: White;  
15     font-size: 50px;  
16     text-align: center;  
17     padding-top: 200px;  
18 }
```

(the above code snippet online)

If you save and reload the page, you will see this:



Background Image No Repeat Centered

We have used the `background-position` property with the value `center`. This property is used to position the background image. Do a google search to find out what are the possible values this property can take. For our purpose here, the `center` value positions the image on the center of its containing element.

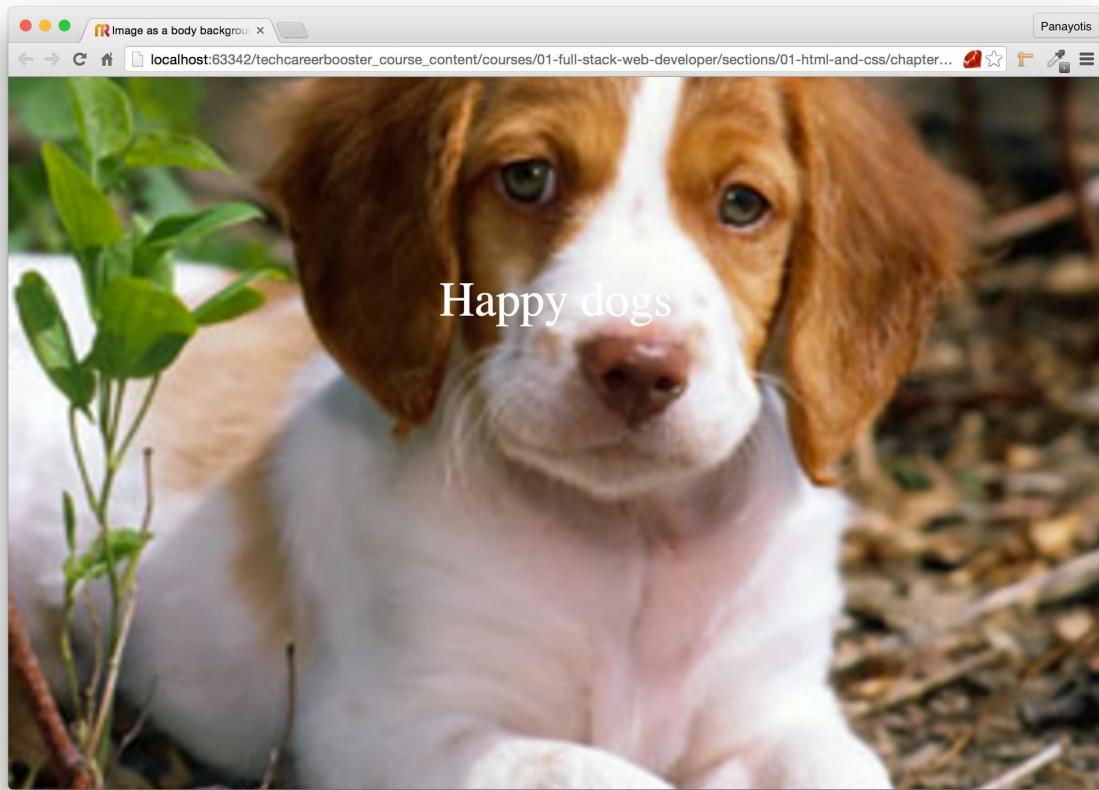
Cover The Whole Body

Almost perfect! One thing that we have to do now is to make sure that the background image covers the whole browser window. Let's use the following CSS content:

```
1 html {  
2     height: 100%;  
3     background-color: White;  
4 }  
5 body {  
6     min-height: 100%;  
7     background-image: url("../images/cute-medium-sized-dog.jpg");  
8     background-repeat: no-repeat;  
9     background-position: center;  
10    background-size: cover;  
11    margin: 0;  
12 }  
13  
14 #main-header {  
15     color: White;  
16     font-size: 50px;  
17     text-align: center;  
18     padding-top: 200px;  
19 }
```

(the above code snippet online)

If you save and reload your page, you will see the following:



Background Image No Repeat Centered Cover

Fantastic! The background image covers the whole browser window as we initially wanted it to be. This has been achieved with the CSS property `background-size`. The `cover` value instructs the browser to scale the background image to be as large as possible, so that it covers the whole body area.

Adding Text On the Page

Now, let's suppose that our page has enough content to expand below the bottom edge of our browser window. Let's assume that our HTML page is now this:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Image as a body background</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9          <div id="main-header">Happy dogs</div>
10         <p>
11             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\
```

```
12 por incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis no\
13 strud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis \
14 aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat\
15 nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu\
16 i officia deserunt mollit anim id est laborum.  
17     </p>  
18     <p>  
19         Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\
20 por incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis no\
21 strud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis \
22 aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat\
23 nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu\
24 i officia deserunt mollit anim id est laborum.  
25     </p>  
26     <p>  
27         Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\
28 por incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis no\
29 strud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis \
30 aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat\
31 nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu\
32 i officia deserunt mollit anim id est laborum.  
33     </p>  
34     <p>  
35         Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\
36 por incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis no\
37 strud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis \
38 aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat\
39 nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu\
40 i officia deserunt mollit anim id est laborum.  
41     </p>  
42     <p>  
43         Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\
44 por incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis no\
45 strud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis \
46 aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat\
47 nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu\
48 i officia deserunt mollit anim id est laborum.  
49     </p>  
50     <p>  
51         Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tem\
52 por incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis no\
53 strud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis \
54 aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat\
55 nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qu\
56 i officia deserunt mollit anim id est laborum.  
57     </p>
```

```
58     </body>
59 </html>
```

(the above code snippet online)

with the following stylesheets/main.css file:

```
1  html {
2      height: 100%;
3      background-color: White;
4  }
5  body {
6      min-height: 100%;
7      background-image: url("../images/cute-medium-sized-dog.jpg");
8      background-repeat: no-repeat;
9      background-position: center;
10     background-size: cover;
11     margin: 0;
12 }
13
14 #main-header {
15     color: White;
16     font-size: 50px;
17     text-align: center;
18     padding-top: 200px;
19 }
20
21 p {
22     padding: 30px 20px;
23     font-size: 18pt;
24     color: White;
25 }
```

(the above code snippet online)

What we have done was to add some text inside p elements and then style the p elements.

Now, the page is longer than the browser window. We need to scroll to see the content that expands at the bottom of the page. This, unfortunately, makes the background image scroll too:

Scrolling of the background image

Make the Image Fixed

So, how can we have the image fixed and not scrolling when we scroll the content of the page? We need to use another CSS property, which is called background-attachment. Here it is how:

```
1 html {
2     height: 100%;
3     background-color: White;
4 }
5 body {
6     min-height: 100%;
7     background-image: url("../images/cute-medium-sized-dog.jpg");
8     background-repeat: no-repeat;
9     background-position: center;
10    background-size: cover;
11    background-attachment: fixed;
12    margin: 0;
13 }
14
15 #main-header {
16     color: White;
17     font-size: 50px;
18     text-align: center;
19     padding-top: 200px;
20 }
21
22 p {
23     padding: 30px 20px;
24     font-size: 18pt;
25     color: White;
26 }
```

(the above code snippet online)

The `background-attachment` with value `fixed` instructs the browser to keep the background image on fixed position and not to scroll it with the HTML page content scrolling. If you reload your page and try to scroll down, you will see the image staying still. Very nice!

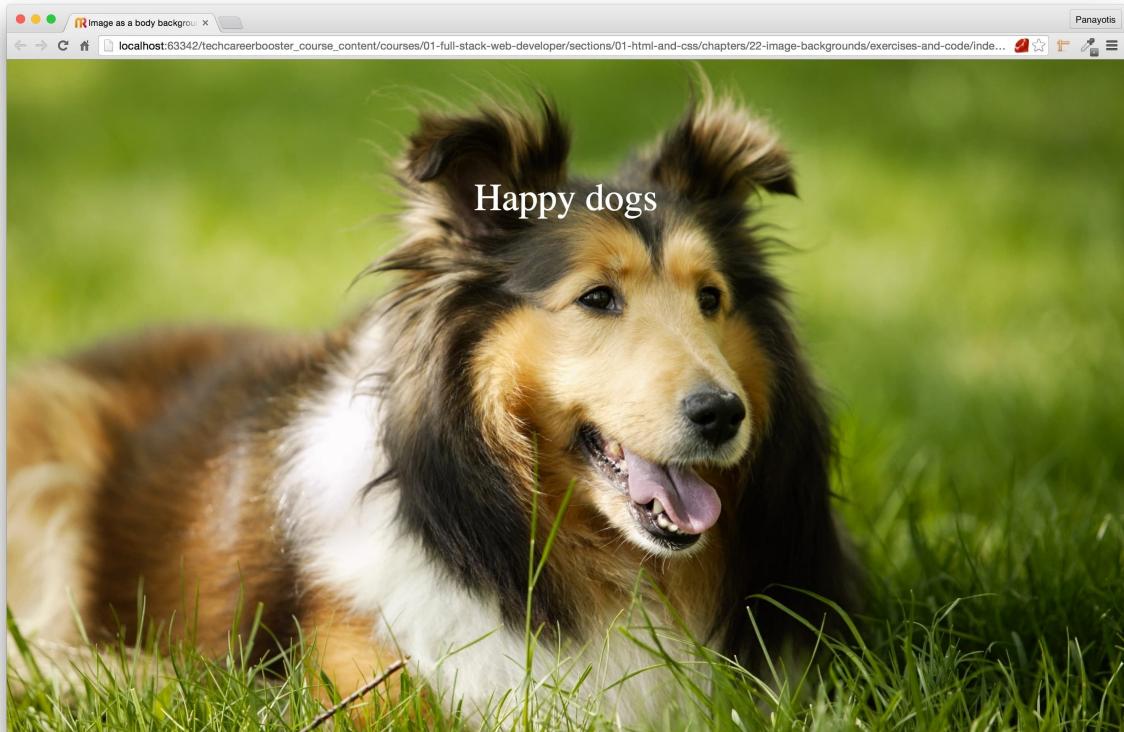
[Video showing scrolling of text with image fixed](#)

Use A Big Image

We have managed to use a background image for our site, but the image that we used is small with regards to our browser window size. This makes the image, being enlarged by the browser, look fuzzy. And this phenomenon, is worse when you enlarge your browser window.

In order to have the image be looking crispy and with high quality, nowadays, when the displays have high resolution and the browser windows large dimensions, you need to use high quality background images with increased width and height sizes. Try something that is bigger than 2000 x 1500. Let's see that with the image `nice-looking-dog.jpg`, which is a high resolution image. Change the `background-image` CSS property to point to that image and reload your page.

This is what you will finally see:

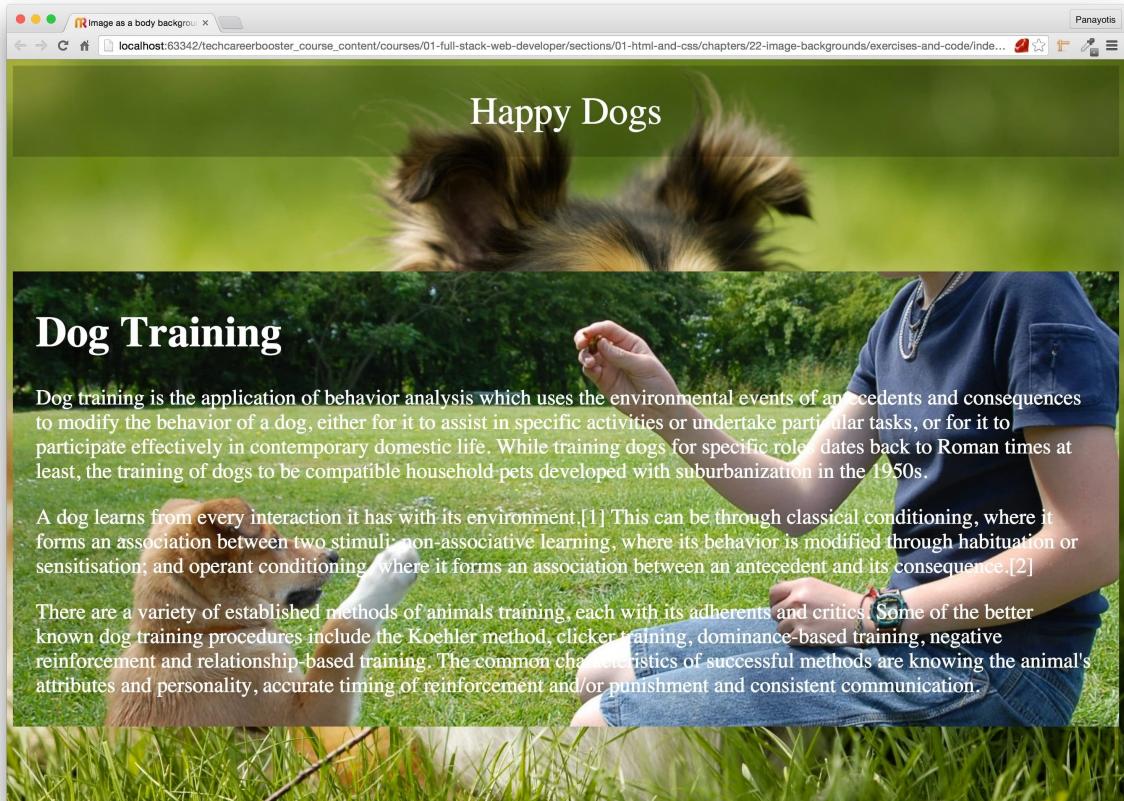


Background Image Fixed Cover High Quality

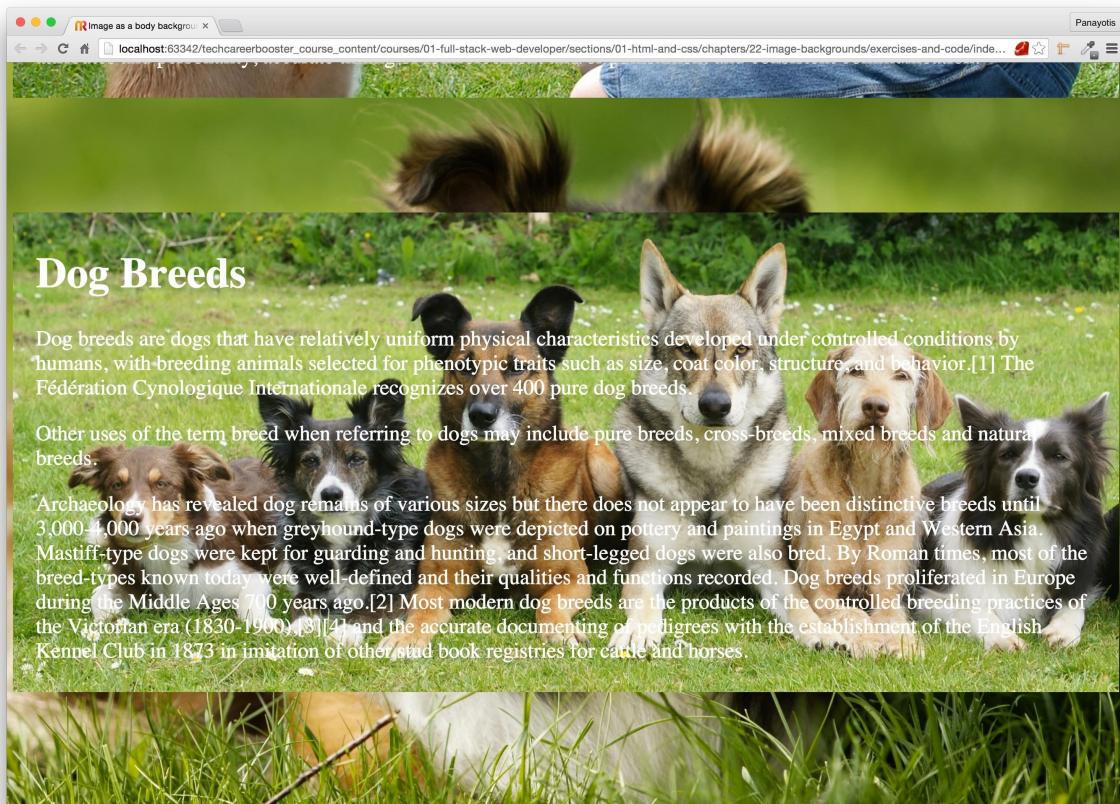
Background Images on Parts of Page

We have created a nice background image for the whole page of our site. But how can we attach background images to parts of the site?

Let's suppose that we want to create the following site. Watch the screenshots and the video:



Background Image On Parts Of Page - First Top Part of the Page



Background Image On Parts Of Page - Second Bottom Part of the Page

As you can see above, we have enhanced the site that we originally built, so that it has 2 parts of the page covered with other background image and some text inside.

How the site displays its content when we scroll from top to bottom

So, let's try to build this site now.

First we need to deal with the "Happy Dogs" header title. It needs to be at the top of the page. So, our HTML content is like this:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Image as a body background</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9          <div id="main-header">Happy Dogs</div>
10     </body>
11 </html>
```

(the above code snippet online)

And the CSS content is:

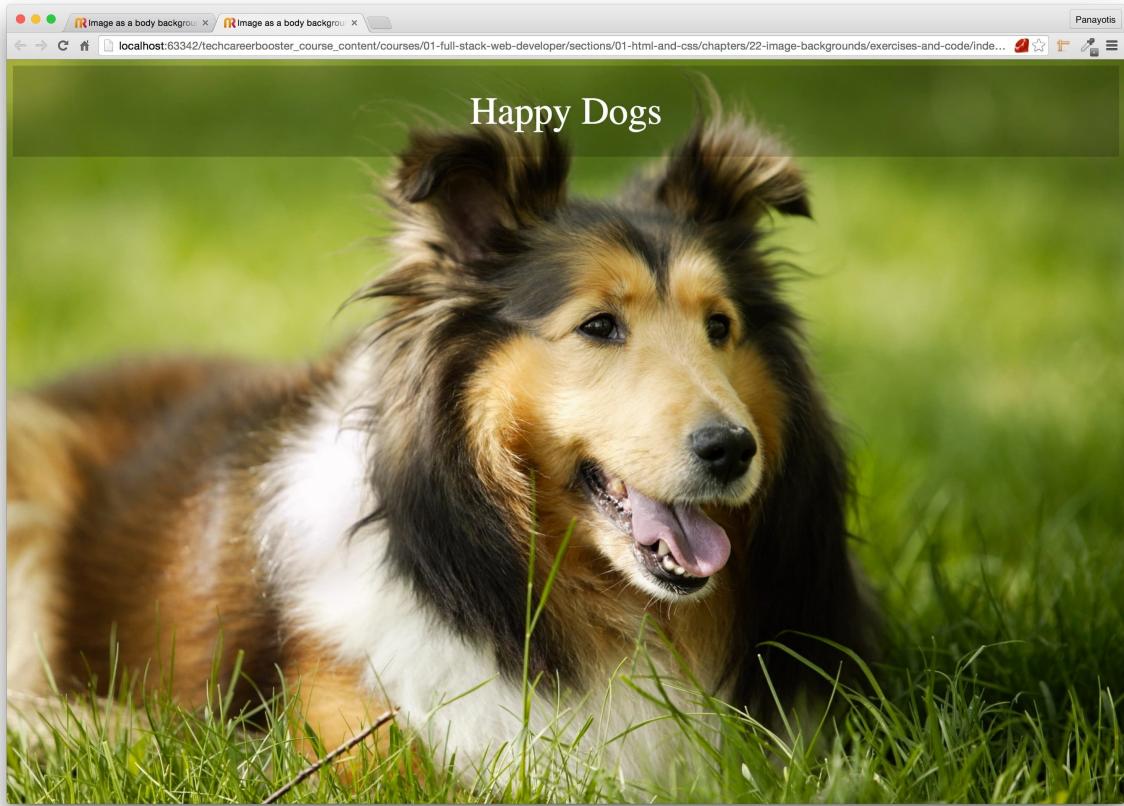
```
1 html {  
2     height: 100%;  
3     background-color: White;  
4 }  
5 body {  
6     min-height: 100%;  
7     background-image: url("../images/nice-looking-dog.jpg");  
8     background-repeat: no-repeat;  
9     background-attachment: fixed;  
10    background-position: center;  
11    background-size: cover;  
12    margin: 0;  
13 }  
14  
15 #main-header {  
16     color: White;  
17     font-size: 50px;  
18     text-align: center;  
19     padding: 30px;  
20     background-color: rgba(0, 0, 0, 0.2);  
21 }
```

(the above code snippet online)

The important stuff on the CSS is that:

1. We have created an empty area around the text, using padding: 30px.
2. We have set the background color of the div to be black but with a lot of transparency (80%, i.e. 20% opacity - 0.2). This makes the text be easier to read and at the same time allows the image behind, to be visible. Creates the nice background blur effect behind the text and in between the background image.

Let's see the result:



Background Image Happy Dogs on Top - Blurred

Next step is to create the “Dog Training” container. Before we see the actual HTML, let’s make some notes here:

- This container does not start exactly below the “Happy Dogs” container. How can we achieve this gap? We can introduce a bottom margin to “Happy Dogs” container.
- The “Dog Training” container has its own background image. It is the file “dog-training.jpg” inside the images of the chapter.
- The “Dog Training” background image:
 - Does not repeat
 - Covers the size of its container
 - Will scroll as the page scrolls

Let’s see how we can introduce the “Dog Training” container with these properties. Here is the HTML:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Image as a body background</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9          <div id="main-header">Happy Dogs</div>
10
11         <div id="dog-training">
12             <h1>Dog Training</h1>
13             <p>Dog training is the application of behavior analysis which uses the envi\ 
14 ronmental events of antecedents and consequences to modify
15                 the behavior of a dog, either for it to assist in specific activities or \
16 undertake particular tasks, or for it to participate effectively
17                 in contemporary domestic life. While training dogs for specific roles dat\
18 es back to Roman times at least, the training of dogs to be
19                 compatible household pets developed with suburbanization in the 1950s.
20             </p>
21
22             <p>
23                 A dog learns from every interaction it has with its environment.[1] This \
24 can be through classical conditioning, where
25                 it forms an association between two stimuli; non-associative learning, wh\ 
26 ere its behavior is modified through habituation
27                 or sensitisation; and operant conditioning, where it forms an association\
28 between an antecedent and its consequence.[2]
29             </p>
30
31             <p>
32                 There are a variety of established methods of animals training, each with\
33 its adherents and critics. Some of the better
34                 known dog training procedures include the Koehler method, clicker trainin\
35 g, dominance-based training, negative reinforcement
36                 and relationship-based training. The common characteristics of successful\
37 methods are knowing the animal's attributes and
38                 personality, accurate timing of reinforcement and/or punishment and consi\
39 stent communication.
40             <p>
41         </div>
42     </body>
43 </html>
```

(the above code snippet online)

As you can inspect in the HTML code above, we have only added the `div` with the necessary

HTML content. Nothing special here. Let's see, though, the CSS content:

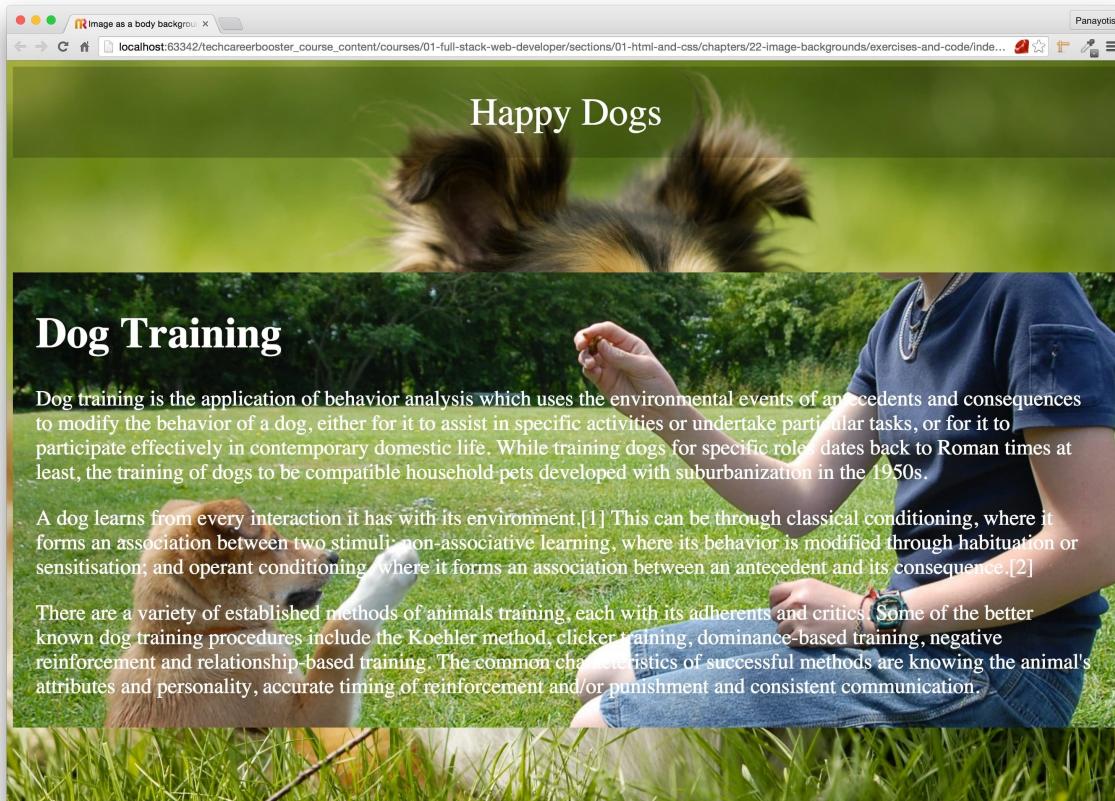
```
1 html {
2     height: 100%;
3     background-color: White;
4 }
5 body {
6     min-height: 100%;
7     background-image: url("../images/nice-looking-dog.jpg");
8     background-repeat: no-repeat;
9     background-attachment: fixed;
10    background-position: center;
11    background-size: cover;
12    margin: 0;
13 }
14
15 #main-header {
16     color: White;
17     font-size: 50px;
18     text-align: center;
19     padding: 30px;
20     margin-bottom: 150px;
21     background-color: rgba(0, 0, 0, 0.2);
22 }
23
24 #dog-training {
25     background-image: url("../images/dog-training.jpg");
26     background-repeat: no-repeat;
27     background-size: cover;
28     font-size: 28px;
29     padding: 10px 30px;
30     color: white;
31 }
```

(the above code snippet online)

The additions here are the following:

1. We have added the `margin-bottom: 150px;` for the “Happy Dogs” container. This will create the gap between the top “Happy Dogs” container and the bottom “Dog Training” container.
2. We have added rules for the “Dog Training” container.
 1. Rules for the background image.
 2. Rules for the font size.
 3. Rules for padding.
 4. Rules for text color.

Let's save all files and reload the page. Here is what we see:



Background Image Dog Training Ready

Nice! "Dog Training" container has been introduced and it is now as required. The next container, the "Dog Breeds" one, is a container that has similar properties to the ones of "Dog Training". The difference is only the background image used.

Let's introduce the HTML content first. Our HTML page now should have the following code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Image as a body background</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9          <div id="main-header">Happy Dogs</div>
10
11         <div id="dog-training">
12             <h1>Dog Training</h1>
13             <p>Dog training is the application of behavior analysis which uses the envi\ 
14 ronmental events of antecedents and consequences to modify
```

```
15      the behavior of a dog, either for it to assist in specific activities or \
16 undertake particular tasks, or for it to participate effectively
17      in contemporary domestic life. While training dogs for specific roles dat\
18 es back to Roman times at least, the training of dogs to be
19      compatible household pets developed with suburbanization in the 1950s.
20  </p>
21
22  <p>
23      A dog learns from every interaction it has with its environment.[1] This \
24 can be through classical conditioning, where
25      it forms an association between two stimuli; non-associative learning, wh\
26 ere its behavior is modified through habituation
27      or sensitisation; and operant conditioning, where it forms an association\
28 between an antecedent and its consequence.[2]
29  </p>
30
31  <p>
32      There are a variety of established methods of animals training, each with\
33 its adherents and critics. Some of the better
34      known dog training procedures include the Koehler method, clicker trainin\
35 g, dominance-based training, negative reinforcement
36      and relationship-based training. The common characteristics of successful\
37 methods are knowing the animal's attributes and
38      personality, accurate timing of reinforcement and/or punishment and consi\
39 stent communication.
40  <p>
41  </div>
42
43  <div id="dog-breeds">
44      <h1>Dog Breeds</h1>
45      <p>Dog breeds are dogs that have relatively uniform physical characteristic\
46 s developed under controlled conditions by humans,
47      with breeding animals selected for phenotypic traits such as size, coat c\
48 olor, structure, and behavior.[1] The Fédération Cynologique
49      Internationale recognizes over 400 pure dog breeds.
50  </p>
51
52  <p>
53      Other uses of the term breed when referring to dogs may include pure bree\
54 ds, cross-breeds, mixed breeds and natural breeds.
55  </p>
56
57  <p>
58      Archaeology has revealed dog remains of various sizes but there does not \
59 appear to have been distinctive breeds until 3,000-4,000 years ago
60      when greyhound-type dogs were depicted on pottery and paintings in Egypt \
```

```
61 and Western Asia. Mastiff-type dogs were kept for guarding and hunting,  
62 and short-legged dogs were also bred. By Roman times, most of the breed-t\\  
63 ypes known today were well-defined and their qualities and functions recorded.  
64 Dog breeds proliferated in Europe during the Middle Ages 700 years ago.[2\\  
65 ] Most modern dog breeds are the products of the controlled breeding practices  
66 of the Victorian era (1830-1900),[3][4] and the accurate documenting of p\\  
67 edigrees with the establishment of the English Kennel Club in 1873 in imitation  
68 of other stud book registries for cattle and horses.  
69 <p>  
70 </div>  
71 </body>  
72 </html>
```

(the above code snippet online)

The only new thing here is the new `div` for the “Dog Breeds” container. How can we style this? We can style it with similar properties like the ones for “Dog Training”, but referencing the image file `dog-breeds.jpg`. Here is the CSS fragment.

```
1 #dog-breeds {  
2     background-image: url("../images/dog-breeds.jpg");  
3     background-repeat: no-repeat;  
4     background-size: cover;  
5     font-size: 28px;  
6     padding: 10px 30px;  
7     color: white;  
8 }
```

(the above code snippet online)

Add the above to the CSS file, save and reload your page. You will see a page that is almost what we really wanted. The problems are actually 2:

1. The “Dog Breeds” container starts exactly below the “Dog Training” container. What we wanted was to have another gap in between the two containers.
2. The “Dog Breeds” container bottom aligns a little bit above the bottom of the page. What we wanted was to have another closing gap at the bottom, below the “Dog Breeds” container.

Let’s solve the 1st problem. We will add a bottom margin to the “Dog Training” container:

Hence, the CSS content can be specified as follows:

```
1 html {
2     height: 100%;
3     background-color: White;
4 }
5 body {
6     min-height: 100%;
7     background-image: url("../images/nice-looking-dog.jpg");
8     background-repeat: no-repeat;
9     background-attachment: fixed;
10    background-position: center;
11    background-size: cover;
12    margin: 0;
13 }
14
15 #main-header {
16     color: White;
17     font-size: 50px;
18     text-align: center;
19     padding: 30px;
20     background-color: rgba(0, 0, 0, 0.2);
21     margin-bottom: 150px;
22 }
23 #dog-training {
24     background-image: url("../images/dog-training.jpg");
25     background-repeat: no-repeat;
26     background-size: cover;
27     font-size: 28px;
28     padding: 10px 30px;
29     color: white;
30     margin-bottom: 150px;
31 }
32
33 #dog-breeds {
34     background-image: url("../images/dog-breeds.jpg");
35     background-repeat: no-repeat;
36     background-size: cover;
37     font-size: 28px;
38     padding: 10px 30px;
39     color: white;
40 }
```

(the above code snippet online)

If you save the above CSS file and reload your page, you will see that “Dog Breeds” does not start immediately after “Dog Training” container. There is a gap of 150px.

conditioning, where it forms an association between two stimuli; non-associative learning, where its behavior is modified through habituation or sensitisation; and operant conditioning, where it forms an association between an antecedent and its consequence.[2]

There are a variety of established methods of animals training, each with its adherents and critics. Some of the better known dog training procedures include the Koehler method, clicker training, dominance-based training, negative reinforcement and relationship-based training. The common characteristics of successful methods are knowing the animal's attributes and personality, accurate timing of reinforcement and/or punishment and consistent communication.

Dog Breeds

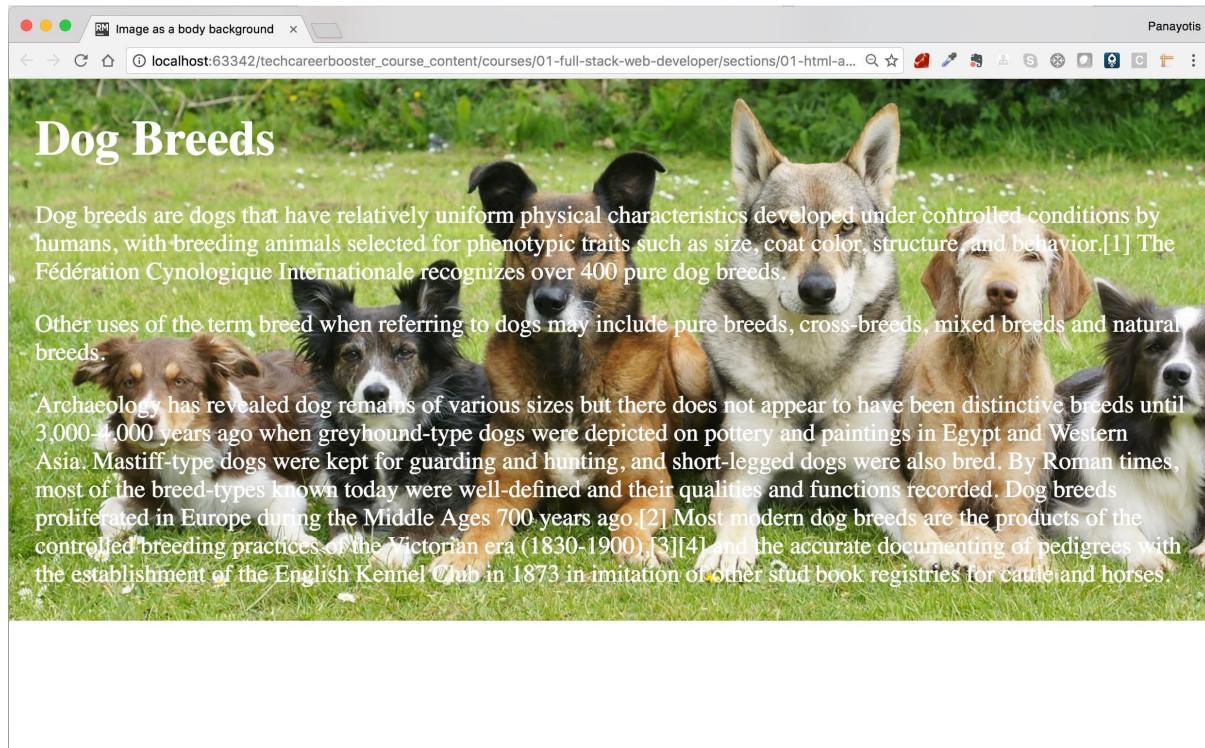
Dog breeds are dogs that have relatively uniform physical characteristics developed under controlled conditions by humans, with breeding animals selected for phenotypic traits such as size, coat color, structure, and behavior.[1] The Fédération Cynologique Internationale recognizes over

Gap 150px between Dog Training and Dog Breeds

However, we still have to solve 2nd problem. We want to have a gap between the bottom edge of the “Dog Breeds” container and the bottom of the page.

Bottom Margin Will Not Work

If you try to add a bottom margin to #dog-breeds container, e.g. `margin-bottom: 150px;` like we have for the #dog-training container, then you will see that the margin is added, but extends below the bottom edge of the body element.



Bottom Margin Extends Below the Bottom Edge of the Body element

Note: Why bottom margin is a problem? See later on.

A Blank Div Will Do

However, if we add a div with a blank space in it and height equal to the number of pixels we want the bottom gap to be, problem is solved. Here is the CSS:

```
1 html {  
2     height: 100%;  
3     background-color: White;  
4 }  
5 body {  
6     min-height: 100%;  
7     background-image: url("../images/nice-looking-dog.jpg");  
8     background-repeat: no-repeat;  
9     background-attachment: fixed;  
10    background-position: center;  
11    background-size: cover;  
12    margin: 0;  
13 }  
14  
15 #main-header {  
16     color: White;  
17     font-size: 50px;
```

```
18     text-align: center;
19     padding: 30px;
20     background-color: rgba(0, 0, 0, 0.2);
21     margin-bottom: 150px;
22 }
23 #dog-training {
24     background-image: url("../images/dog-training.jpg");
25     background-repeat: no-repeat;
26     background-size: cover;
27     font-size: 28px;
28     padding: 10px 30px;
29     color: white;
30     margin-bottom: 150px;
31 }
32
33 #dog-breeds {
34     background-image: url("../images/dog-breeds.jpg");
35     background-repeat: no-repeat;
36     background-size: cover;
37     font-size: 28px;
38     padding: 10px 30px;
39     color: white;
40 }
41
42 #bottom-gap {
43     height: 150px;
44 }
```

(the above code snippet online)

And here is the HTML:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Image as a body background</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8   <body>
9     <div id="main-header">Happy Dogs</div>
10
11    <div id="dog-training">
12      <h1>Dog Training</h1>
13      <p>Dog training is the application of behavior analysis which uses the envi\ 
14 ronmental events of antecedents and consequences to modify
15       the behavior of a dog, either for it to assist in specific activities or \
```

```
16 undertake particular tasks, or for it to participate effectively
17     in contemporary domestic life. While training dogs for specific roles dat\
18 es back to Roman times at least, the training of dogs to be
19     compatible household pets developed with suburbanization in the 1950s.
20     </p>
21
22     <p>
23         A dog learns from every interaction it has with its environment.[1] This \
24 can be through classical conditioning, where
25             it forms an association between two stimuli; non-associative learning, wh\
26 ere its behavior is modified through habituation
27             or sensitisation; and operant conditioning, where it forms an association\
28 between an antecedent and its consequence.[2]
29     </p>
30
31     <p>
32         There are a variety of established methods of animals training, each with\
33 its adherents and critics. Some of the better
34             known dog training procedures include the Koehler method, clicker trainin\
35 g, dominance-based training, negative reinforcement
36             and relationship-based training. The common characteristics of successful\
37 methods are knowing the animal's attributes and
38             personality, accurate timing of reinforcement and/or punishment and consi\
39 stent communication.
40     <p>
41     </div>
42
43     <div id="dog-breeds">
44         <h1>Dog Breeds</h1>
45         <p>Dog breeds are dogs that have relatively uniform physical characteristic\
46 s developed under controlled conditions by humans,
47             with breeding animals selected for phenotypic traits such as size, coat c\
48 olor, structure, and behavior.[1] The Fédération Cynologique
49             Internationale recognizes over 400 pure dog breeds.
50     </p>
51
52     <p>
53         Other uses of the term breed when referring to dogs may include pure bree\
54 ds, cross-breeds, mixed breeds and natural breeds.
55     </p>
56
57     <p>
58         Archaeology has revealed dog remains of various sizes but there does not \
59 appear to have been distinctive breeds until 3,000-4,000 years ago
60             when greyhound-type dogs were depicted on pottery and paintings in Egypt \
61 and Western Asia. Mastiff-type dogs were kept for guarding and hunting,
```

```
62      and short-legged dogs were also bred. By Roman times, most of the breed-t\
63 types known today were well-defined and their qualities and functions recorded.
64      Dog breeds proliferated in Europe during the Middle Ages 700 years ago.[2\
65 ] Most modern dog breeds are the products of the controlled breeding practices
66      of the Victorian era (1830-1900),[3][4] and the accurate documenting of p\
67 edigrees with the establishment of the English Kennel Club in 1873 in imitation
68      of other stud book registries for cattle and horses.
69      <p>
70      </div>
71
72      <div id="bottom-gap">
73      &nbsp;
74      </div>
75      </body>
76 </html>
```

(the above code snippet online)

If you save the above CSS file and reload your page, you will see that you finally have the result that you wanted from the beginning. The HTML page is ready.

Why Bottom Margin Didn't Work?

This is because margins of a contained `div` are not taken into account when calculating the height of the container block element (i.e. of the `body` in our case) unless there is a `div` next to them that would occupy part of the contained area of the container. In other words, on our case, the bottom margin of the last `div` is not take into account to calculate the necessary body height, and hence, scrolling at the bottom, we can see the white gap created (i.e. the background image is not used to draw that area).

Having said that, another approach to the requirement to have a bottom gap, below the “Dog Breeds” container would have been the following:

1. Add the bottom margin of 150px to the “Dog Breeds” container and
2. Add the `#bottom-gap` container with the blank inside, but without attaching CSS rule to that for its height:

Here is the CSS:

```
1 html {
2     height: 100%;
3     background-color: White;
4 }
5 body {
6     min-height: 100%;
7     background-image: url("../images/nice-looking-dog.jpg");
8     background-repeat: no-repeat;
9     background-attachment: fixed;
10    background-position: center;
11    background-size: cover;
12    margin: 0;
13 }
14
15 #main-header {
16     color: White;
17     font-size: 50px;
18     text-align: center;
19     padding: 30px;
20     background-color: rgba(0, 0, 0, 0.2);
21     margin-bottom: 150px;
22 }
23 #dog-training {
24     background-image: url("../images/dog-training.jpg");
25     background-repeat: no-repeat;
26     background-size: cover;
27     font-size: 28px;
28     padding: 10px 30px;
29     color: white;
30     margin-bottom: 150px;
31 }
32
33 #dog-breeds {
34     background-image: url("../images/dog-breeds.jpg");
35     background-repeat: no-repeat;
36     background-size: cover;
37     font-size: 28px;
38     padding: 10px 30px;
39     color: white;
40     margin-bottom: 150px;
41 }
```

(the above code snippet online)

The HTML remains the same like before.

Move Background to HTML instead

Another alternative, would have been to move the background image at the `html` level rather than the `body` level. Do you want to try that?

Important: CSS is an old technology that evolves over time, now being in CSS3. The same problem can be solved with different techniques. This chapter presents some of the techniques. It is 100% sure, that you will learn more the more you practice and study front-end development.

DRYing our Code

Did we finish? **No**. The fact that the result is the one that we want, does not mean that we have done good job. We need to take care of our code too. Our code needs to be clean and with as few repetitions as possible.

Important: You need to be DRY with your code. This is a general rule that, if followed, creates a code base that it is easy to maintain. DRY stands for Don't Repeat Yourself. It essentially advice you to avoid code repetitions as a code reuse method. Code should not be reused by copying/pasting techniques, but should be reused with the help of the tools that the language you write the code to provides you with. We will repeat this advice many times within the course and you will learn which tools are helping you to DRY your code.

Our CSS file contains a lot of repetition that we need to eliminate. Here is what we mean:

1. `dog-training` and `dog-breeds` containers have all their CSS properties in common, except from the background image. Hence, we need to find a way to specify the common properties only once.
2. `dog-training`, `dog-breeds` and `main-header` containers have all a bottom margin of 150px height. We need to find a way to specify this only once.

Here is the final DRYed CSS code:

```
1 html {  
2     height: 100%;  
3     background-color: White;  
4 }  
5 body {  
6     min-height: 100%;  
7     background-image: url("../images/nice-looking-dog.jpg");  
8     background-repeat: no-repeat;  
9     background-attachment: fixed;  
10    background-position: center;  
11    background-size: cover;
```

```
12     margin: 0;
13 }
14
15 .block-element {
16     margin-bottom: 150px;
17 }
18
19 #main-header {
20     color: White;
21     font-size: 50px;
22     text-align: center;
23     padding: 30px;
24     background-color: rgba(0, 0, 0, 0.2);
25 }
26
27 #dog-training {
28     background-image: url("../images/dog-training.jpg");
29 }
30
31 #dog-breeds {
32     background-image: url("../images/dog-breeds.jpg");
33 }
34
35 #dog-training, #dog-breeds {
36     background-repeat: no-repeat;
37     background-size: cover;
38     font-size: 28px;
39     padding: 10px 30px;
40     color: white;
41 }
```

(the above code snippet online)

As you can see above

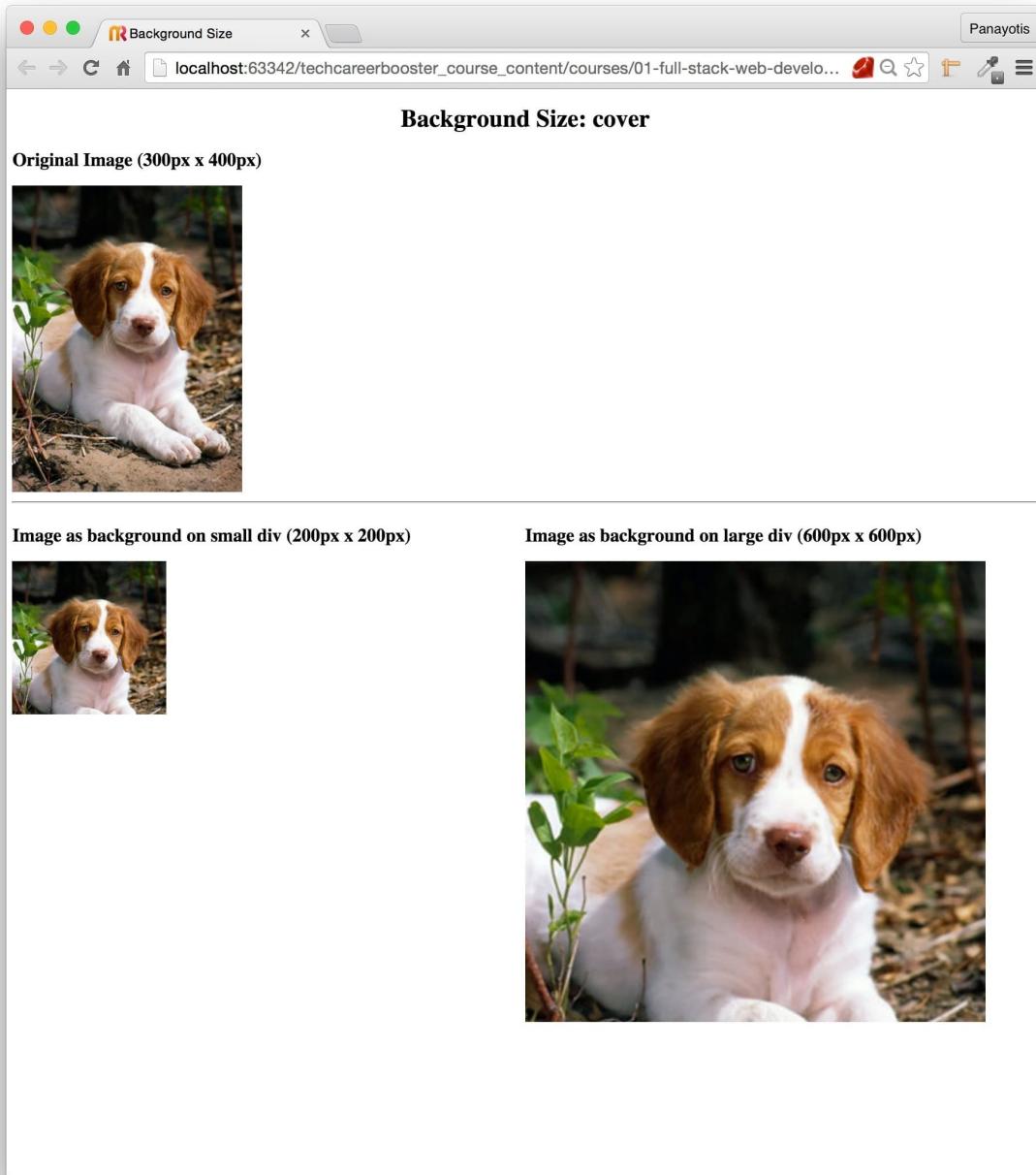
1. We specify one class with name `.block-element` that can be used for the 3 containers that require the same bottom margin.
2. We have specified only once the CSS properties for `dog-training` and `dog-breeds` that are common to both of them.

Don't forget to use the class `.block-element` inside your HTML content. It needs to be applied to all 3 containers.

Background Size

Here, we would like to draw your attention to the fact that the value `cover` for `background-size` crops the image.

Let's see an example. Look at the following page:



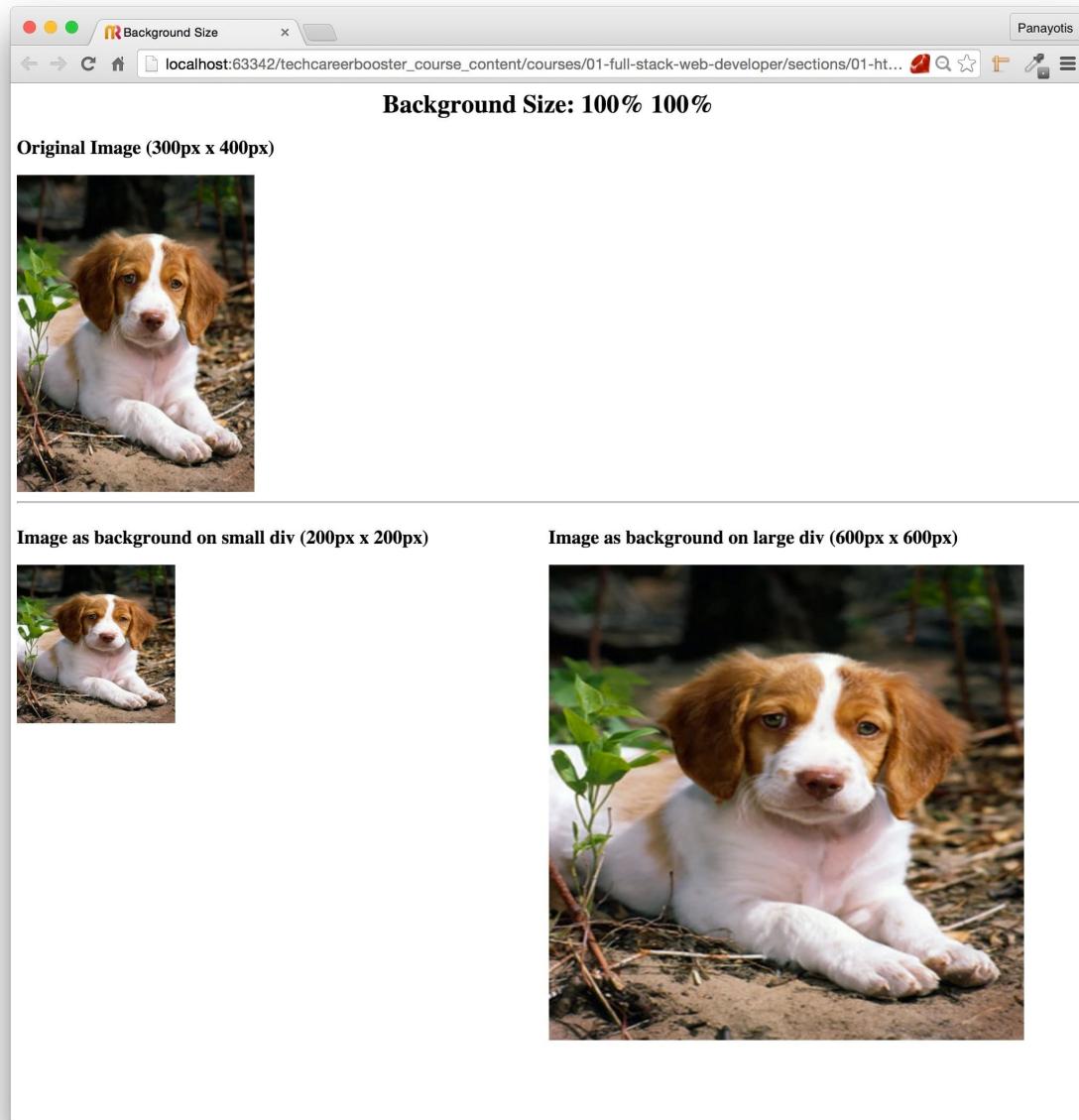
Page Demo for Background Image Size as Cover

The above page shows how a background image is applied with value `cover` for its `background-size` property. It is used as a background to the two bottom divs. Above, you see the original image. As you can see, the bottom images are cropped. The legs of the dog are not visible any more. This is because we have used the value `cover` for the `background-size` property.

If you want the div to contain the whole image, you will need to use the value `100% 100%`; for the `background-size` property. The first `100%` is for the width of the image and the second one is for the height. With these values, we are asking the browser to make sure that the dimensions

of the image drawn are equal to the div it is attached to.

This is the result of the previous page, but with `background-size: 100% 100%;` used (instead of `background-size: cover`).



Page Demo for Background Image Size with 100%

As you can see above, the images on the bottom divs are not cropped, if compared to the original image. However, they are stretched. This is because the containers are square (200 x 200 and 600 x 600) whereas the original image is a portrait image with width less than height.

Hence, if you decide to use `background-size: 100% 100%;` make sure that your divs have size which is analogous to the original image, so that the aspect ratio (width size / height size) is reserved. Otherwise, your images will not look good (unless this is what you want in the first place).

For the sake of completeness, we are giving you here the source code of the above two pages:

HTML for cover case:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Background Size</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h1>Background Size: cover</h1>
11         <h2>Original Image (300px x 400px)</h2>
12         <div>
13             
14         </div>
15
16         <hr/>
17
18         <div id="left-column" class="column">
19             <h2>Image as background on small div (200px x 200px)</h2>
20
21             <div id="small-div" class="image-background">
22                 </div>
23             </div>
24
25             <div id="right-column" class="column">
26                 <h2>Image as background on large div (600px x 600px)</h2>
27                 <div id="large-div" class="image-background">
28                     </div>
29             </div>
30
31         </body>
32     </html>
```

(the above code snippet online)

CSS for cover case:

```
1 body {
2     height: 1300px;
3 }
4
5 .column {
6     width: 50%;
7     float: left;
8 }
9
10 h1 {
11     text-align: center;
12 }
13
14 .image-background {
15     background-repeat: no-repeat;
16     background-position: top left;
17     background-size: cover;
18 }
19
20 #small-div {
21     width: 200px;
22     height: 200px;
23     background-image: url("../images/cute-medium-sized-dog.jpg");
24 }
25
26 #large-div {
27     width: 600px;
28     height: 600px;
29     background-image: url("../images/cute-medium-sized-dog.jpg");
30 }
```

(the above code snippet online)

HTML for 100% 100% case:

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8">
5         <title>Background 100% 100%</title>
6         <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7     </head>
8
9     <body>
10        <h1>Background Size: 100% 100%</h1>
11        <h2>Original Image (300px x 400px)</h2>
12        <div>
```

```
13     
14 </div>
15
16 <hr/>
17
18 <div id="left-column" class="column">
19     <h2>Image as background on small div (200px x 200px)</h2>
20
21     <div id="small-div" class="image-background">
22         </div>
23 </div>
24
25 <div id="right-column" class="column">
26     <h2>Image as background on large div (600px x 600px)</h2>
27     <div id="large-div" class="image-background">
28         </div>
29 </div>
30
31 </body>
32 </html>
```

(the above code snippet online)

CSS for 100% 100% case:

```
1 body {
2     height: 1300px;
3 }
4
5 .column {
6     width: 50%;
7     float: left;
8 }
9
10 h1 {
11     text-align: center;
12 }
13
14 .image-background {
15     background-repeat: no-repeat;
16     background-position: top left;
17     background-size: 100% 100%;
18 }
19
20 #small-div {
21     width: 200px;
22     height: 200px;
```

```
23     background-image: url("../images/cute-medium-sized-dog.jpg");  
24 }  
25  
26 #large-div {  
27     width: 600px;  
28     height: 600px;  
29     background-image: url("../images/cute-medium-sized-dog.jpg");  
30 }
```

(the above code snippet online)

Background Color

Having used all the background-x properties here, I would like to tell you that you can specify the CSS property `background-color` alongside the `background-image` property. This is done if we want the browser to fall back to background color if cannot load the background image.

background shorthand property

You will often see CSS code using the `background` property instead of the specific background-x properties that we have learned above. The `background` property is used as a shorthand, to specify the values for all the following background-x properties and in this order:

1. `background-color`
2. `background-image`
3. `background-repeat`
4. `background-attachment`
5. `background-position`

So, the following two CSS fragments are equivalent:

```
1 body {  
2     background-color: maroon;  
3     background-image: url("../images/architecture-body-background-image.jpg");  
4     background-repeat: no-repeat;  
5     background-attachment: fixed;  
6     background-position: center;  
7     background-size: cover;  
8 }
```

(the above code snippet online)

and

```
1 body {  
2     background: maroon url("../images/architecture-body-background-image.jpg") no\  
3 -repeat fixed center;  
4     background-size: cover;  
5 }
```

(the above code snippet online)

Note that `background-size` is not part of the `background` shorthand property and needs to be specified separately.

Note also that, when using the `background` shorthand, you do not have to specify values for all the properties, as long as the ones specified are in the correct order:

- Color
- Image
- Repeat
- Attachment
- Position

Try to memorize this acronym: **C I R A P**. It will help you remember the order of the `background-x` properties inside the `background` shorthand.

Also, when you specify the `background` properties separately, try to put them in the same order, like this:

```
1 body {  
2     background-color: maroon;  
3     background-image: url("../images/architecture-body-background-image.jpg");  
4     background-repeat: no-repeat;  
5     background-attachment: fixed;  
6     background-position: center;  
7     background-size: cover;  
8 }
```

(the above code snippet online)

and not like this:

```
1 body {  
2     background-color: maroon;  
3     background-position: center;  
4     background-repeat: no-repeat;  
5     background-attachment: fixed;  
6     background-image: url("../images/architecture-body-background-image.jpg");  
7     background-size: cover;  
8 }
```

(the above code snippet online)

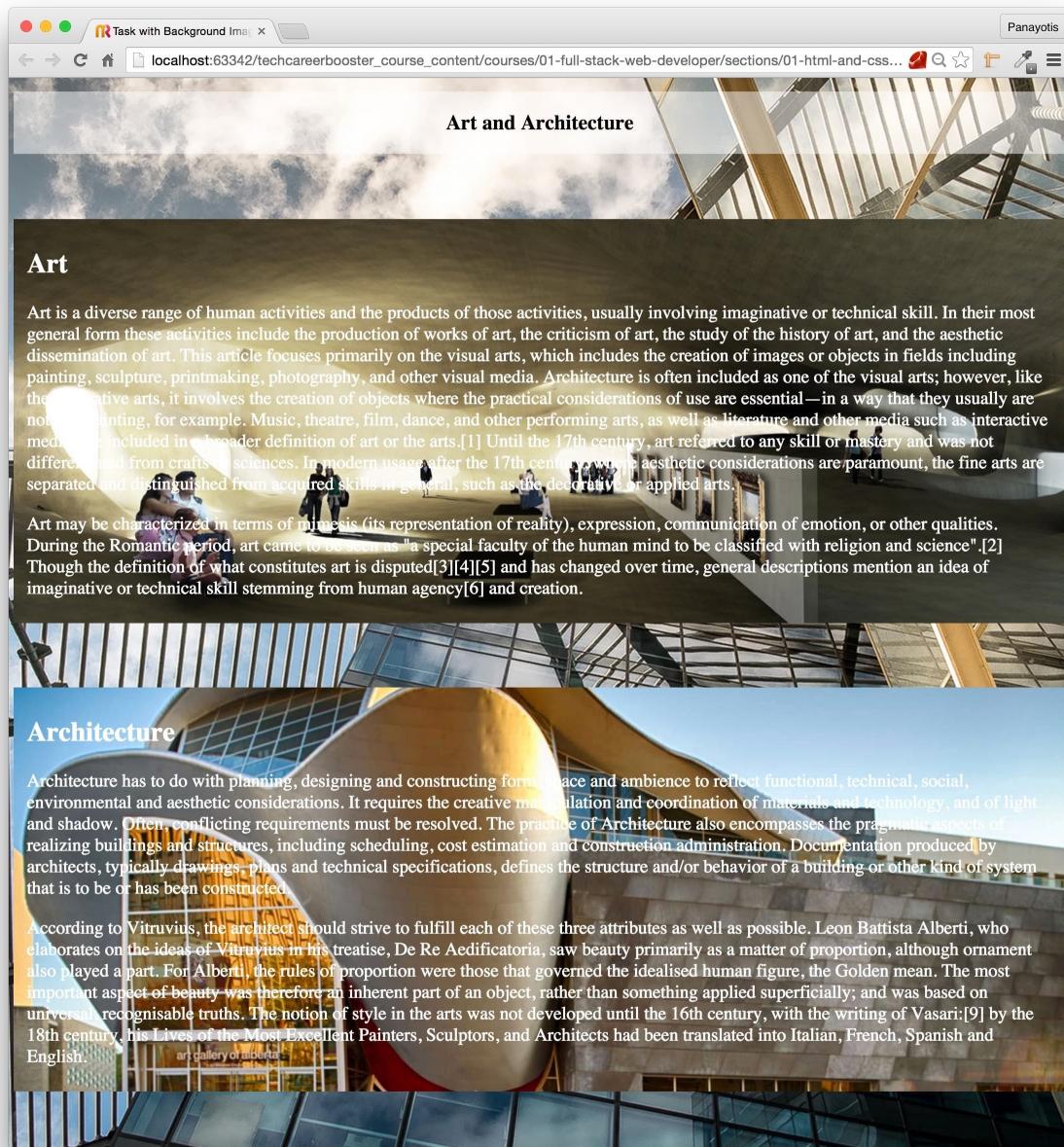
Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Upload your Happy Dogs site to your Netlify account.
2. Create a site like the following:



Task with Background Image

3. Here is a video showing that live:

Site for your task

4. Things that you need to be aware of:

1. The images that you will need are:
 1. task-architecture-body-background-image.jpg
 2. task-art-background-image.jpg
 3. task-architecture-background-image.jpg
2. The margin between the containers needs to be 100px.
3. The background color of the main header container is a transparent white color.
4. The main header has top and bottom padding equal to 30px and right and left padding equal to 50px.

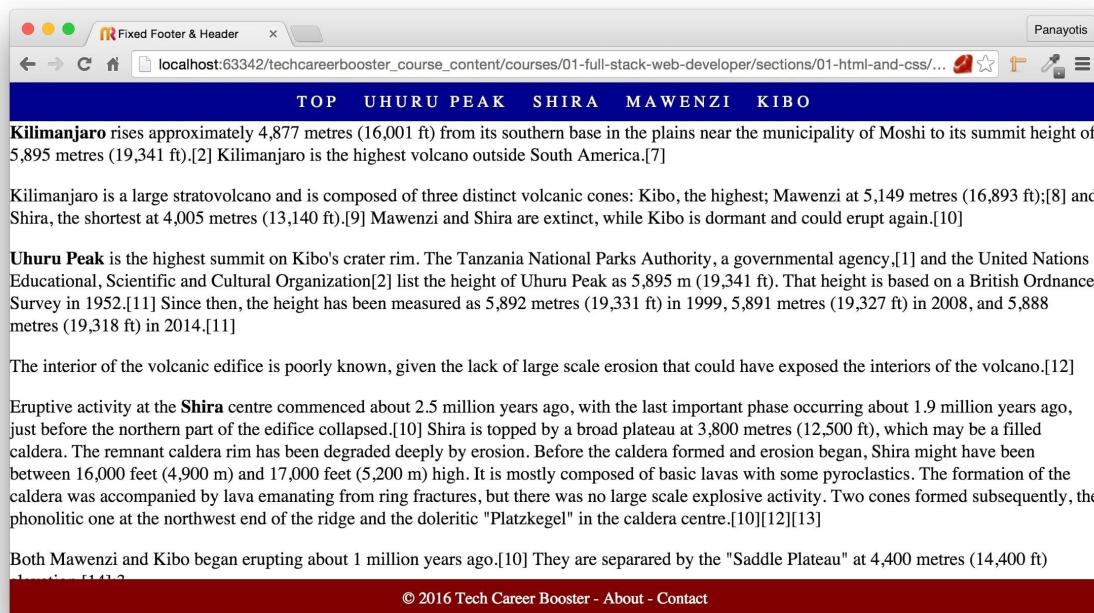
5. Font size for the paragraph text containers is 28px.
6. Paragraph text containers have top and bottom padding equal to 10px and right and left padding equal to 20px.
7. You need to be DRY. Do not repeat code if you can avoid that.
5. Upload your site to your Netlify account.

Chapter 23 - Element Positioning

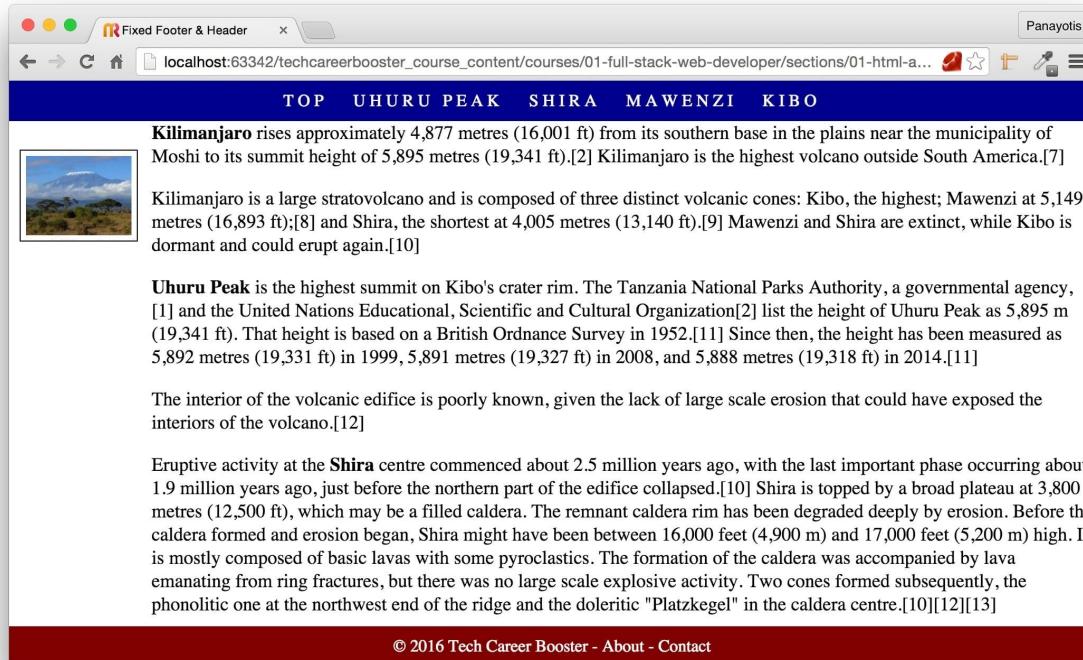
Summary

In this chapter we are going to learn how we can position our HTML elements on the canvas. The elements are usually positioned using the normal page flow. However, we may be in the need to position elements out of the normal flow, at specific positions within the browser window.

We build pages like these:

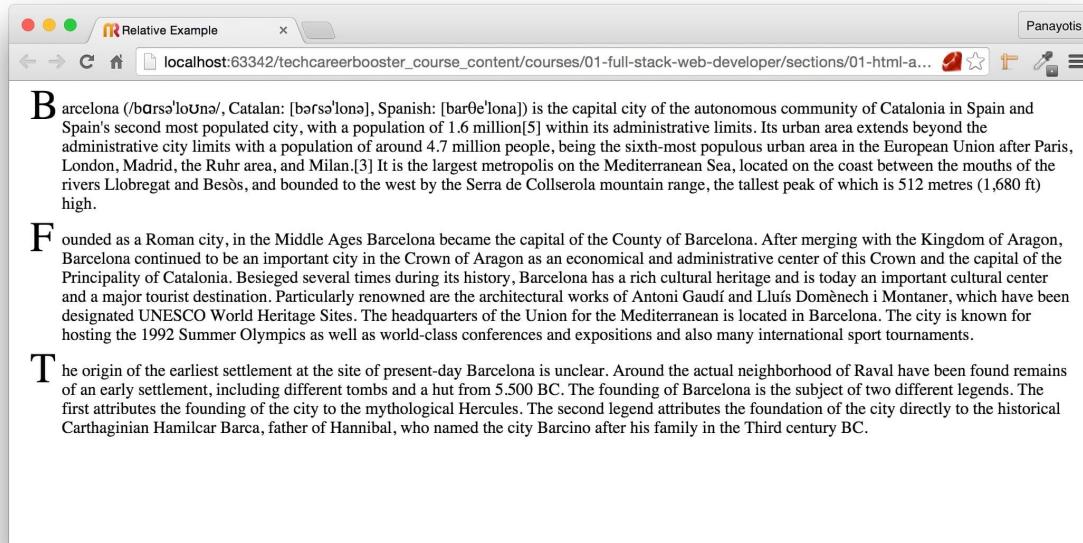


Page With Top and Bottom Nav Bar



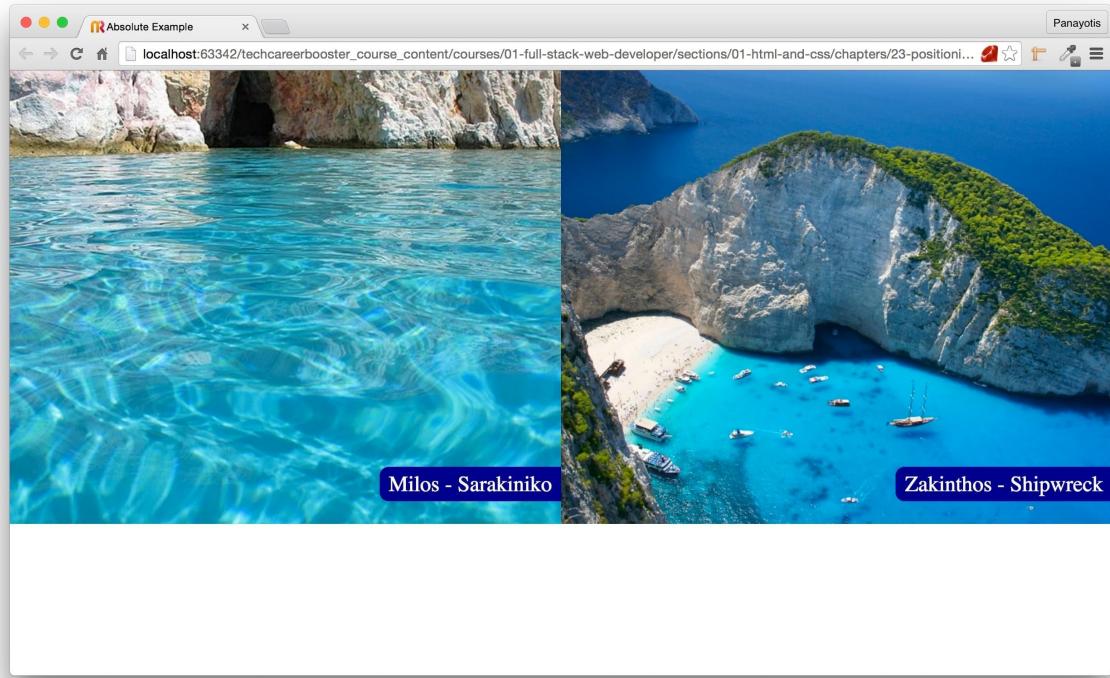
The screenshot shows a web browser window with a title bar "Fixed Footer & Header" and a status bar "Panayotis". The main content area has a blue header bar with the text "TOP UHURU PEAK SHIRA MAWENZI KIBO". Below the header, there is a small image of Mount Kilimanjaro on the left. The text discusses the mountain's height and three volcanic cones: Kibo, Mawenzi, and Shira. It also mentions the Tanzania National Parks Authority and the United Nations Educational, Scientific and Cultural Organization. A paragraph about Uhuru Peak follows, mentioning its height and the British Ordnance Survey. Another paragraph discusses the interior of the volcano and its eruptive history. At the bottom of the page is a red footer bar with the text "© 2016 Tech Career Booster - About - Contact".

Page With Fixed Image On The Left

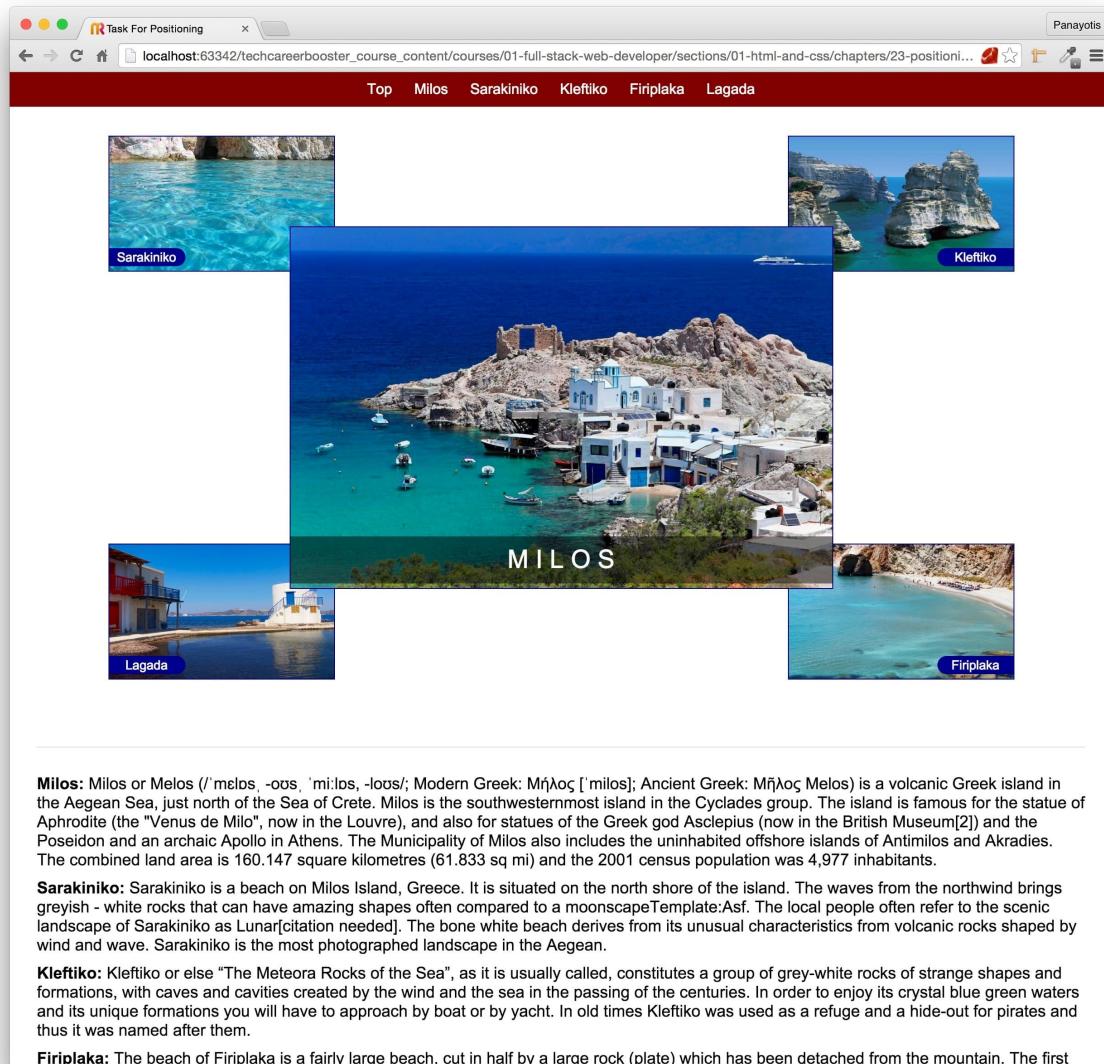


The screenshot shows a web browser window with a title bar "Relative Example" and a status bar "Panayotis". The main content area features a large image of the city of Barcelona on the left side. To the right of the image, the text describes Barcelona as the capital city of Catalonia in Spain and the second most populous city in Spain. It highlights its rich cultural heritage, including Antoni Gaudí and Lluís Domènech i Montaner, and its status as a major tourist destination. The text also mentions its designation as a UNESCO World Heritage Site. Another paragraph discusses the origin of the city, mentioning the mythical Hercules and the Carthaginian Hamilcar Barca.

Page with Paragraphs First Char Stand Out



Two Images With Tags



Milos: Milos or Melos (/ˈmɛlɒs, -oʊs, ˈmiːlɒs, -loʊs/; Modern Greek: Μήλος [ˈmílos]; Ancient Greek: Μῆλος Melos) is a volcanic Greek island in the Aegean Sea, just north of the Sea of Crete. Milos is the southwesternmost island in the Cyclades group. The island is famous for the statue of Aphrodite (the "Venus de Milo", now in the Louvre), and also for statues of the Greek god Asclepius (now in the British Museum[2]) and the Poseidon and an archaic Apollo in Athens. The Municipality of Milos also includes the uninhabited offshore islands of Antimilos and Akradies. The combined land area is 160.147 square kilometres (61.833 sq mi) and the 2001 census population was 4,977 inhabitants.

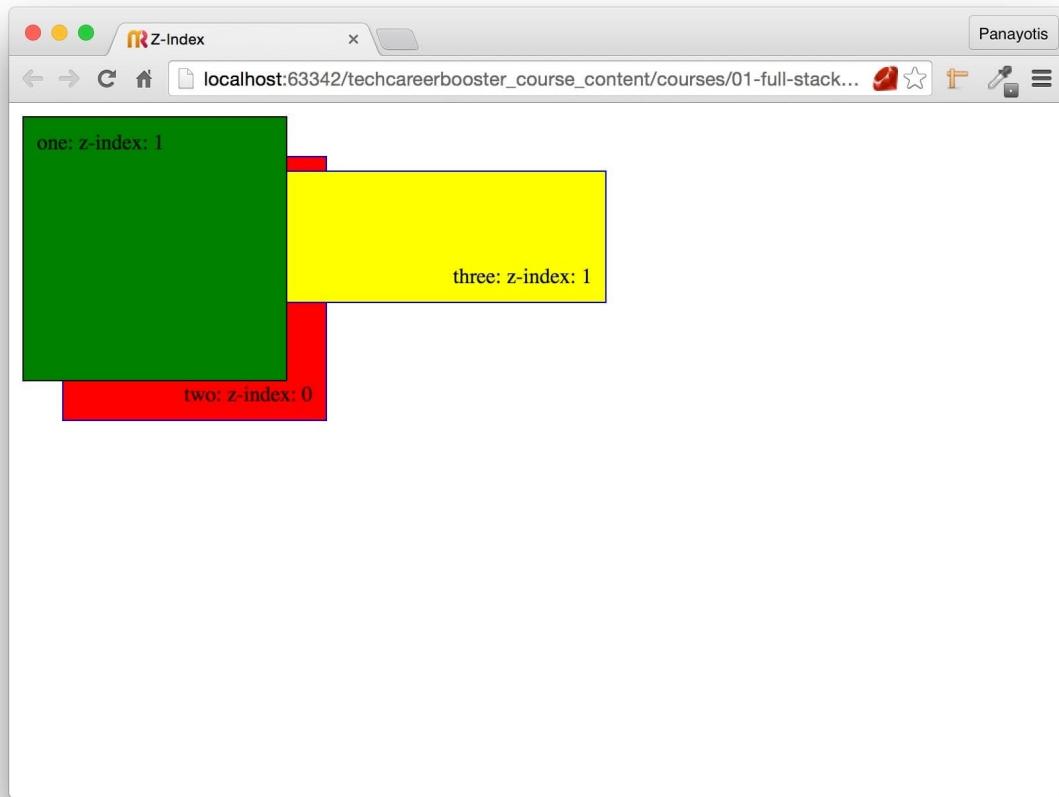
Sarakiniko: Sarakiniko is a beach on Milos Island, Greece. It is situated on the north shore of the island. The waves from the northwind brings greyish - white rocks that can have amazing shapes often compared to a moonscapeTemplate:Asf. The local people often refer to the scenic landscape of Sarakiniko as Lunar[citation needed]. The bone white beach derives from its unusual characteristics from volcanic rocks shaped by wind and wave. Sarakiniko is the most photographed landscape in the Aegean.

Kleftiko: Kleftiko or else "The Meteora Rocks of the Sea", as it is usually called, constitutes a group of grey-white rocks of strange shapes and formations, with caves and cavities created by the wind and the sea in the passing of the centuries. In order to enjoy its crystal blue green waters and its unique formations you will have to approach by boat or by yacht. In old times Kleftiko was used as a refuge and a hide-out for pirates and thus it was named after them.

Firiplaka: The beach of Firiplaka is a fairly large beach, cut in half by a large rock (plate) which has been detached from the mountain. The first

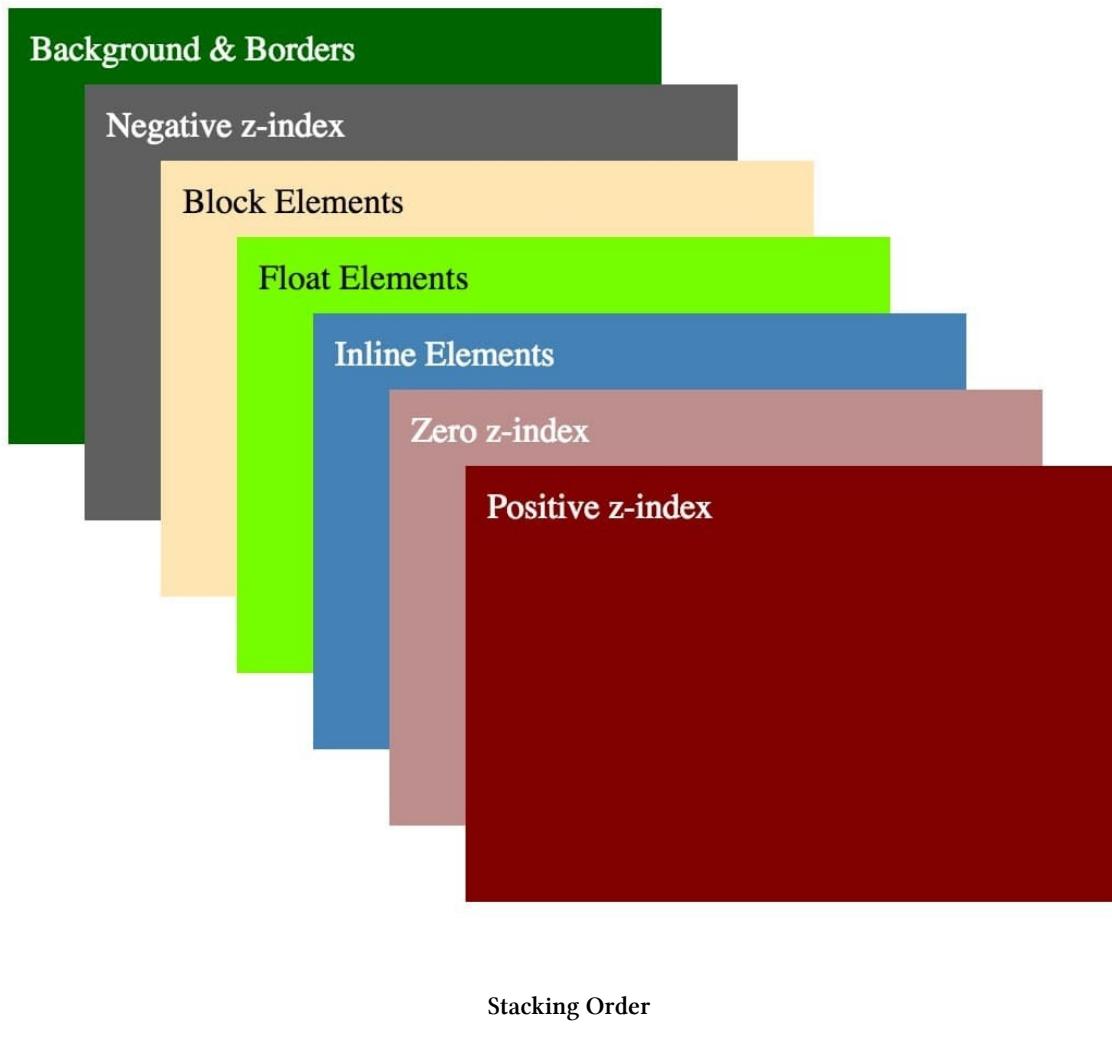
Positioning Example

We will also learn how to stack elements one on top of the other:



Stacked Elements

and about the stacking order used by the browser:



Learning Goals

1. Learn how inline elements are positioned on canvas.
2. Learn about positioning of block elements.
3. Learn about floating elements.
4. Learn about the clear property.
5. Learn about the positioned elements:
 1. fixed
 2. absolute
 3. relative
 4. static
6. Learn about the positioning properties:
 1. top
 2. bottom
 3. right
 4. left

7. Learn about the `z-index` property.
8. Learn about the stacking order.

Until now, we have learned how to place HTML elements on our page, but we didn't have too much freedom with regards to where exactly on the page we could place our elements.

Element Positioning

Inline Elements

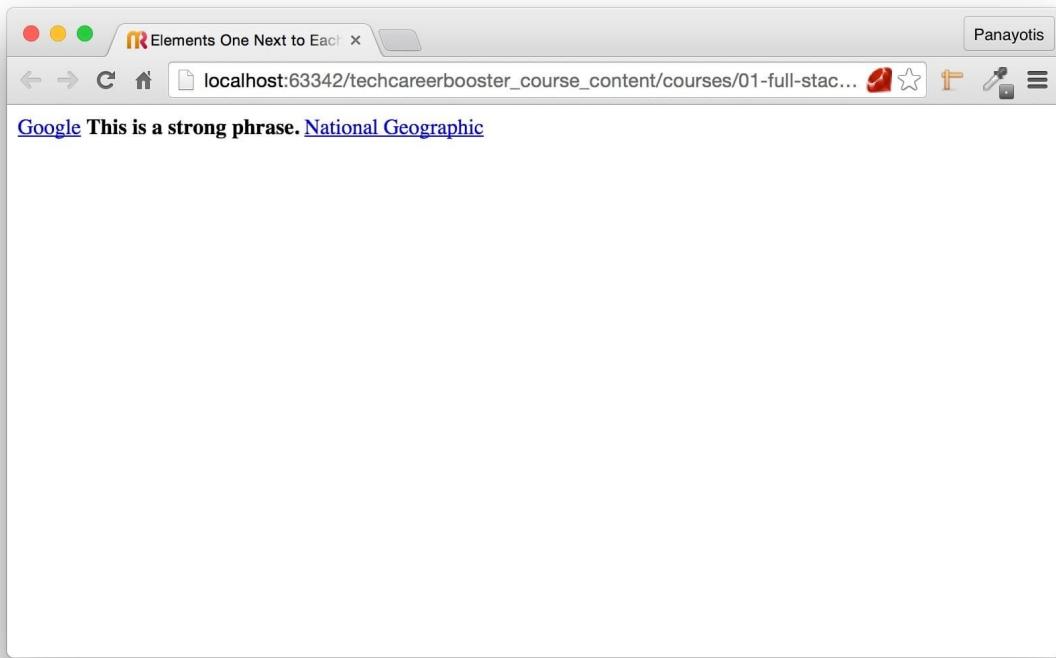
The elements are positioned one next to the other, when they are siblings, and one inside the other when they have containing-contained (parent-child) relationship.

Look at the following HTML code. Write it, save it to a file and load the page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Elements One Next to Each Other</title>
6      </head>
7
8      <body>
9
10         <a href="https://www.google.com">Google</a>
11         <strong>This is a strong phrase. </strong>
12         <a href="https://www.nationalgeographic.com">National Geographic</a>
13
14     </body>
15
16 </html>
```

(the above code snippet online)

If you load this page on your browser, you will see this:



Inline Elements One Next To The Other

As you can see, browser starts to draw elements, from top left, one by one, putting the 3 inline elements that this page includes, one next to the other. If there is enough space for all of them to be printed on the same line, they will. Otherwise, there is going to be a wrapping and ending part of the line will be printed below the starting part of it.

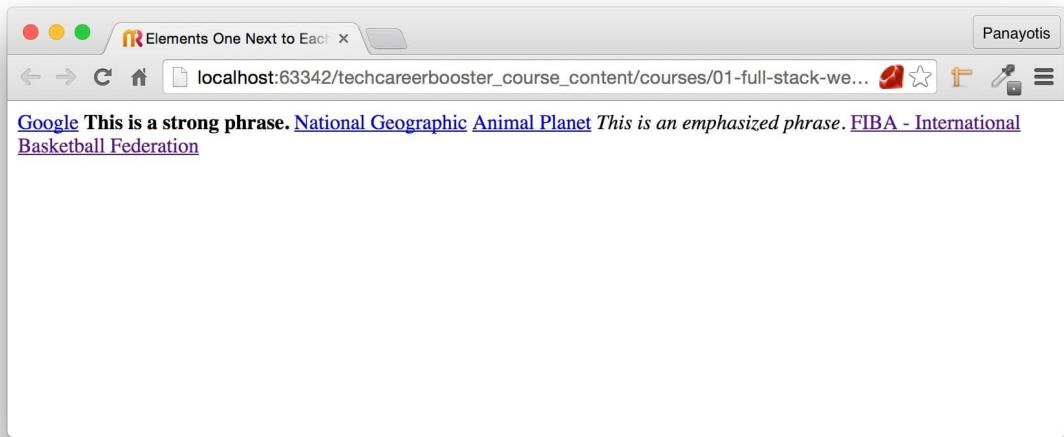
Try for example the following:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Elements One Next to Each Other - With Wrapping</title>
6   </head>
7
8   <body>
9
10    <a href="https://www.google.com">Google</a>
11    <strong>This is a strong phrase. </strong>
12    <a href="https://www.nationalgeographic.com">National Geographic</a>
13    <a href="https://www.animalplanet.com">Animal Planet</a>
14    <em>This is an emphasized phrase. </em>
15    <a href="http://www.fiba.com/">FIBA - International Basketball Federation</a>
16
17  </body>
```

```
18  
19  </html>
```

(the above code snippet online)

If you reload this on your browser, you will see this:



Text Wraps To Next Line if Not Enough Space

As you can see above (make sure your browser window is quite narrow), when there is not much space, there is a wrapping of text to the next line.

But still, the elements are drawn one after the other, exactly in the order they appear inside the HTML document.

Block Elements

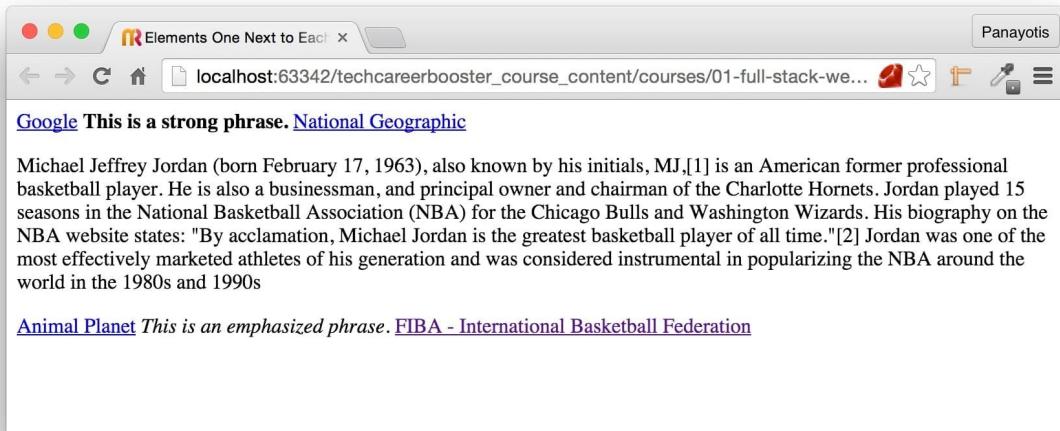
Block elements is one way we can start controlling, little-by-little, the position an HTML element is drawn on the page. Still, with limited capabilities, the block element is drawn on a completely new line, occupies the whole available width and any other HTML element before that or after that are drawn on different lines. So, a block element occupies the whole line without any other element left or right to it.

Let's put a block element in between the existing inline elements, on our last HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Elements One Next to Each Other - Block Elements</title>
6      </head>
7
8      <body>
9
10         <a href="https://www.google.com">Google</a>
11         <strong>This is a strong phrase. </strong>
12         <a href="https://www.nationalgeographic.com">National Geographic</a>
13         <p>
14             Michael Jeffrey Jordan (born February 17, 1963), also known by his initials\
15             , MJ,[1] is an American former professional basketball player.
16             He is also a businessman, and principal owner and chairman of the Charlotte\
17             Hornets. Jordan played 15 seasons in the National Basketball
18                 Association (NBA) for the Chicago Bulls and Washington Wizards. His biograp\
19             hy on the NBA website states: "By acclamation, Michael Jordan
20                 is the greatest basketball player of all time."<[2] Jordan was one of the mo\
21             st effectively marketed athletes of his generation and was
22                 considered instrumental in popularizing the NBA around the world in the 198\
23             0s and 1990s
24         </p>
25         <a href="https://www.animalplanet.com">Animal Planet</a>
26         <em>This is an emphasized phrase. </em>
27         <a href="http://www.fiba.com/">FIBA - International Basketball Federation</a>
28
29     </body>
30
31 </html>
```

(the above code snippet online)

If you save and load this page on your browser, you will see something like this:



Block Element on its own Line

We have added a `<p>` element, which is a block element and you can see how the block element stays “alone”, with a blank line before and after it to separate it from their siblings.

Float Property

Besides the block elements, another tool that we have learned to help us position elements on the canvas, the drawing area, is the `float` property. With `float: left;`, the element occupies the first top left position available. With `float: right;`, occupies the first top right position available. We have seen that, when we tried to put text around an image, like in the following HTML example:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Elements One Next to Each Other - Float Image</title>
6      <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7    </head>
8
9    <body>
10
11      <a href="https://www.google.com">Google</a>
12      <strong>This is a strong phrase. </strong>
13      <a href="https://www.nationalgeographic.com">National Geographic</a>
14      <p>
15         Michael Jeffrey Jordan (born February 17, 19\
17        63), also known by his initials, MJ,[1] is an American former professional basket\
18        ball player.

```

```

19      He is also a businessman, and principal owner and chairman of the Charlotte\
20      Hornets. Jordan played 15 seasons in the National Basketball
21          Association (NBA) for the Chicago Bulls and Washington Wizards. His biograp\
22      hy on the NBA website states: "By acclamation, Michael Jordan
23          is the greatest basketball player of all time."[2] Jordan was one of the mo\
24      st effectively marketed athletes of his generation and was
25          considered instrumental in popularizing the NBA around the world in the 198\
26      0s and 1990s
27      </p>
28      <a href="https://www.animalplanet.com">Animal Planet</a>
29      <em>This is an emphasized phrase. </em>
30      <a href="http://www.fiba.com/">FIBA - International Basketball Federation</a>
31
32      </body>
33
34  </html>

```

(the above code snippet online)

with CSS:

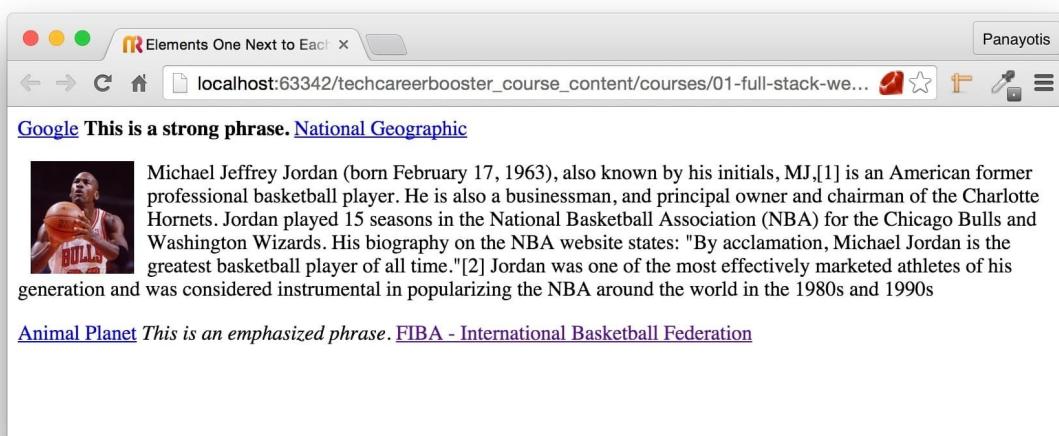
```

1 img {
2     float: left;
3     padding: 0 10px;
4     width: 80px;
5 }

```

(the above code snippet online)

If you save the above files and reload your page, you will see something like that:



Float Used to Wrap Text Around Image

Clear Property

We have also learned about the `clear` property. Elements next to a floating element will float around it. If we want to stop that, then we use the `clear` property. So, with `clear:left;` we prevent any floating element exist to the left of the element this property applies to.

See the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Float and Clear Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="left-column" class="column">
12             Left Column
13         </div>
14
15         <div id="right-column" class="column">
16             Right Column
17         </div>
18
19         <div id="footer">
20             Footer
21         </div>
22
23     </body>
24
25 </html>
```

(the above code snippet online)

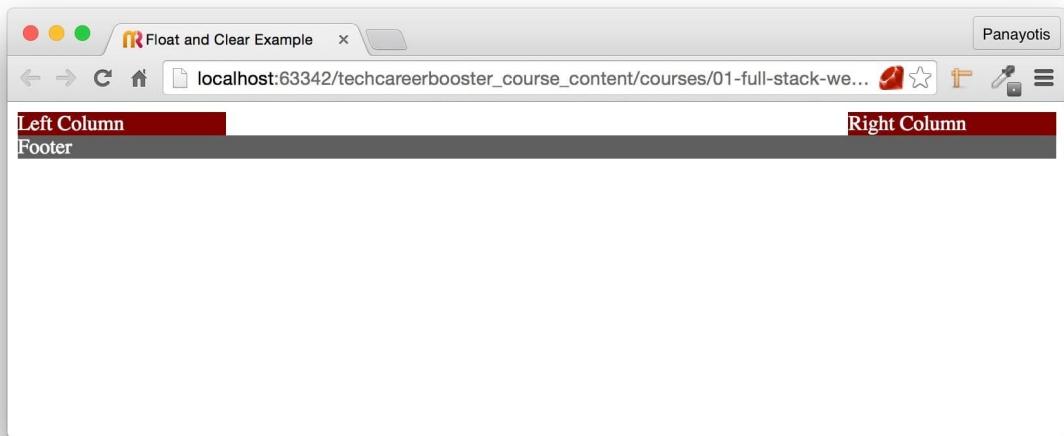
with CSS:

```
1  .column {
2      width: 20%;
3      background-color: Maroon;
4      color: white;
5  }
6
7  #left-column {
8      float: left;
9  }
```

```
10  
11 #right-column {  
12     float: right;  
13 }  
14  
15 #footer {  
16     background-color: #5F5F5F;  
17     color: White;  
18     clear: left;  
19 }
```

(the above code snippet online)

If you save those files and reload your page you will see this:



Clear Property Used On Footer

As you can see from the CSS, the left column floats left, the right column floats right and footer is on new line because it has `clear: left;` which prevents any floating element left to itself. Try to remove the `clear: left;` property rule and reload your page. What you will see? Note that, `clear` can take the value `right` and the value `both`, which refers to both sides of an element.

And that was the summary of the tools that we have learned until now, about influencing the position of elements on our page.

1. block vs inline elements
2. float and clear properties

The position of the elements on the HTML canvas is affected also by the CSS property with name `position`. By default, the value of this property is `static`. Static position means that the element will be positioned next to the previous one, as we described above.

Positioning Relative to Browser Window

Sometimes, we want to be able to position an element to a position relative to the browser window. For example, let's suppose that we want to create a bottom fixed footer like this:



Footer Positioned At Bottom

When you resize this page, the footer always stays at the bottom. Watch the following video to understand what we mean:

Footer is fixed to the bottom

Let's see how we can achieve this. First we write the HTML content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Bottom Fixed Footer</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="footer">
12             &copy; 2016 Tech Career Booster - <a href="#">About</a> - <a href="#">Contact</a>
13         </div>
14
15     </body>
16
17
18 </html>
```

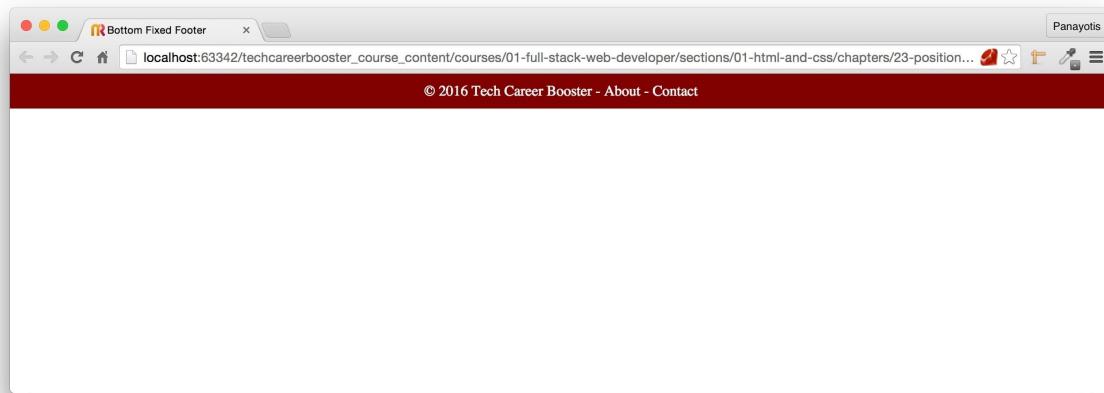
(the above code snippet online)

And let's start with a simple CSS file (`stylesheets/main.css`):

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     margin: 0;
7 }
8
9 #footer {
10    text-align: center;
11    background-color: Maroon;
12    color: White;
13    padding: 10px 0;
14 }
15
16 a {
17    text-decoration: none;
18    color: inherit;
19 }
```

(the above code snippet online)

If you save these files and load your page, you will see this:



Our Footer Appears On top

As you can see, our footer appears on top of the browser window. This is expected, as we said earlier. The elements are drawn from top left to right, in the order they appear inside the HTML content.

In order to ask browser to draw the element on specific position relative to the browser window, you need to use the `position` property with value `fixed`.

Let's do that. In CSS file, in `#footer` selector, we add `position: fixed;`.

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     margin: 0;
7 }
8
9 #footer {
10    text-align: center;
11    background-color: Maroon;
12    color: White;
13    padding: 10px 0;
14    position: fixed;
15 }
16
17 a {
18    text-decoration: none;
19    color: inherit;
20 }
```

(the above code snippet online)

If we save the file and reload the page, we will see this:



Footer Div With Position Fixed

Hmmmmmm. Something strange has happened when we set the `position: fixed;` property to our div. For some reason, it does not occupy the whole available width. But it occupies only the width necessary to host its content. Before we deal with the width of our footer, we will first deal with its actual position.

bottom property

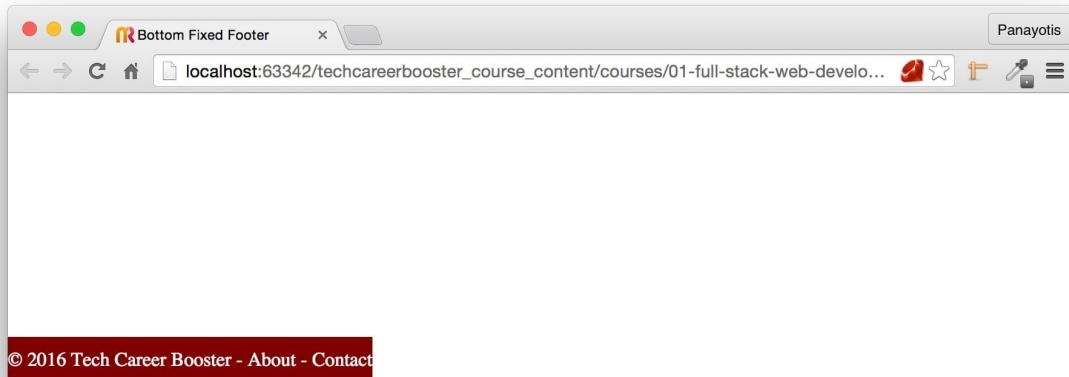
Now it is drawn on the top left of the browser window. What we want is to be drawn at the bottom of the browser window. We will use the `bottom` property. Add the property `bottom` with value `0` for the footer element.

Our CSS becomes:

```
1 * {
2     box-sizing: border-box;
3 }
4
5 body {
6     margin: 0;
7 }
8
9 #footer {
10    text-align: center;
11    background-color: Maroon;
12    color: White;
13    padding: 10px 0;
14    position: fixed;
15    bottom: 0;
16 }
17
18 a {
19    text-decoration: none;
20    color: inherit;
21 }
```

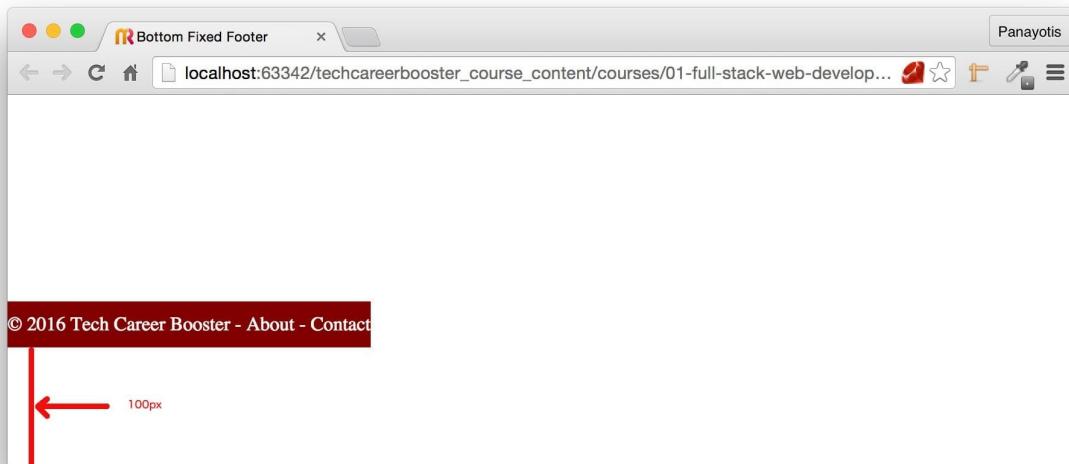
(the above code snippet online)

And if you reload the page, you will see this:



Setting Bottom To 0

Nice. The `bottom` property worked. It specifies the bottom margin distance from the bottom side of the browser window. For example, if you set that to `100px`, and reload your page, you will see that:



Setting Bottom To 100px

As you can see, there is a distance of the bottom margin of the footer box equal to `100px` from the bottom side of the browser window.

Now, let's revert this value back to `0`, since we want our footer to be at the very bottom of the browser window.

What we have to fix now, is the width of the element. The problem with the width started when we specified the `position` property to `fixed`. This is because positioned elements do not occupy the whole width. They occupy the width necessary to host their content. Unless we specify the width explicitly. On our particular case, we want the element to occupy the whole width of the element it is using to position itself relative to. Hence, if we specify the width to be

100%, then we make sure that the footer element will have the width equal to the width of the browser window.

Let's do that. Have your CSS rules be as follows:

```
1 * {
2     box-sizing: border-box;
3 }
4 body {
5     margin: 0;
6 }
7
8 #footer {
9     text-align: center;
10    background-color: Maroon;
11    color: White;
12    padding: 10px 0;
13    position: fixed;
14    bottom: 0;
15    width: 100%;
16 }
17
18 a {
19     text-decoration: none;
20     color: inherit;
21 }
```

(the above code snippet online)

Basically, we have only added one more rule for the footer element. The one that sets its width to 100% value. If you save this file and reload your page, you will see this:



Footer Positioned At Bottom

which is exactly what we wanted to achieve. If you resize your browser, you will see that the footer stays always at the bottom.

Besides this property, i.e. footer staying always at the bottom, you will see also that the footer is not covered by other content of the HTML page, but it stays on the foreground. Let's try to amend our HTML page content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Bottom Fixed Footer</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="footer">
12             &copy; 2016 Tech Career Booster - <a href="#">About</a> - <a href="#">Conta\
13 ct</a>
14         </div>
15
16
17         <div id="main-content">
18             <p>Kilimanjaro rises approximately 4,877 metres (16,001 ft) from its southe\
19 rn base in the plains near the municipality of Moshi to its summit height of 5,89\
20 5 metres
21             (19,341 ft).[2] Kilimanjaro is the highest volcano outside South America.\\
22 [7]
23             </p>
24
25             <p>
26                 Kilimanjaro is a large stratovolcano and is composed of three distinct volc\
27 anic cones: Kibo, the highest; Mawenzi at 5,149 metres (16,893 ft);[8] and Shira,
28                 the shortest at 4,005 metres (13,140 ft).[9] Mawenzi and Shira are extinct,\\
29 while Kibo is dormant and could erupt again.[10]
30             </p>
31
32             <p>
33                 Uhuru Peak is the highest summit on Kibo's crater rim. The Tanzania Nationa\
34 l Parks Authority, a governmental agency,[1] and the United Nations Educational,
35                 Scientific and Cultural Organization[2] list the height of Uhuru Peak as 5,\\
36 895 m (19,341 ft). That height is based on a British Ordnance Survey in 1952.[11]\\
37 Since
38                 then, the height has been measured as 5,892 metres (19,331 ft) in 1999, 5,8\\
39 91 metres (19,327 ft) in 2008, and 5,888 metres (19,318 ft) in 2014.[11]
40             </p>
41
42             <p>
```

43 The interior of the volcanic edifice is poorly known, given the lack of large scale erosion that could have exposed the interiors of the volcano.[12]

44 </p>

45

46 <p>

47 Eruptive activity at the Shira centre commenced about 2.5 million years ago, with the last important phase occurring about 1.9 million years ago, just before

48 the northern part of the edifice collapsed.[10] Shira is topped by a broad plateau at 3,800 metres (12,500 ft), which may be a filled caldera. The remnant caldera

49 rim has been degraded deeply by erosion. Before the caldera formed and erosion began, Shira might have been between 16,000 feet (4,900 m) and 17,000 feet (5,200 m)

50 high. It is mostly composed of basic lavas with some pyroclastics. The formation of the caldera was accompanied by lava emanating from ring fractures, but there was

51 no large scale explosive activity. Two cones formed subsequently, the phonolitic one at the northwest end of the ridge and the doleritic "Platzkegel" in the caldera centre.[10][12][13]

52 </p>

53

54 <p>

55 Both Mawenzi and Kibo began erupting about 1 million years ago.[10] They are separated by the "Saddle Plateau" at 4,400 metres (14,400 ft) elevation.[14]:3

56 </p>

57

58 <p>

59 The youngest dated rocks at Mawenzi are about 448,000 years old.[10] Mawenzi forms a horseshoe shaped ridge with pinnacles and ridges opening to the northeast which

60 has a tower like shape resulting from deep erosion and a mafic dyke swarm. Several large cirques cut into the ring, the largest of these sits on top of the Great

61 Barranco gorge. Also notable are the Ost and West Barrancos on the northeastern side of the mountain. Most of the eastern side of the mountain has been removed by

62 erosion. Mawenzi has a subsidiary peak named Neumann Tower (4,425 metres (14,518 ft)).[10][12][13]

63 </p>

64

65 <p>

66 Kibo is the largest cone and is more than 15 miles (24 km) wide at the "Saddle Plateau" altitude. The last activity here has been dated to between 150,000 and 200,000 years ago and created the current Kibo summit crater. Kibo still has gas-emitting fumaroles in the crater.[10][12][13] Kibo is capped by an

```

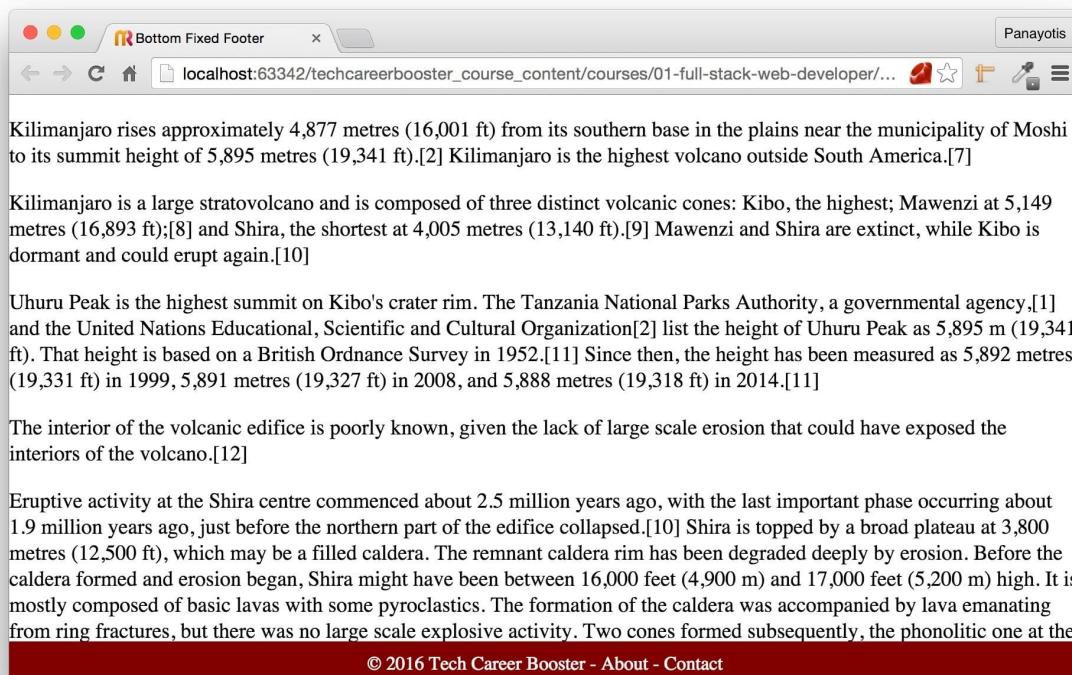
89      almost symmetrical cone with escarpments rising 180 metres (590 ft) to 200 \
90  metres (660 ft) on the south side. These escarpments define a 2.5-kilometre-wide \
91  (1.6 mi)
92      caldera[15] caused by the collapse of the summit. Within this caldera is th\
93  e Inner Cone and within the crater of the Inner Cone is the Reusch Crater, which
94      the Tanganyika government in 1954 named after Gustav Otto Richard Reusch up\
95  on his climbing the mountain for the 25th time (out of 65 attempts during his lif\
96  etime).[16][17]
97      The Ash Pit, 350 metres (1,150 ft) deep, lies within the Reusch Crater.[18]\ \
98  About 100,000 years ago, part of Kibo's crater rim collapsed, creating the area \
99  known as
100     the Western Breach and the Great Barranco.[19]
101     </p>
102     </div>
103
104     </body>
105
106 </html>

```

(the above code snippet online)

And the CSS file with a rule for `main-content` to have `font-size` equal to `18px`.

If you save the above and reload your page, you will see this:



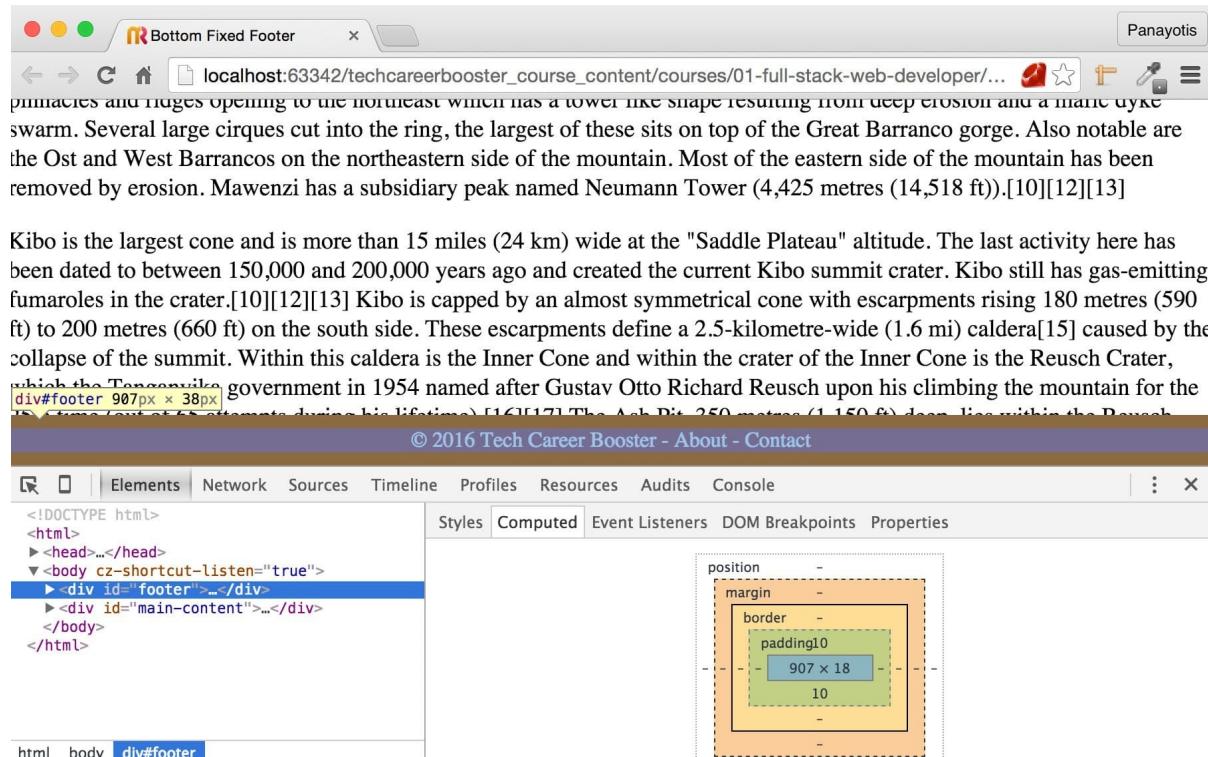
Footer Positioned At Bottom - Long Text in Body

As you can see above, the footer stays at the bottom, even if it is defined in the HTML content first. Also, you will encounter the footer covers the last part of the main text. Try to scroll and see whether you are able to view the last part of the text. You will not be able to view it. Watch this video here that demonstrates that:

How footer covers last part of main text

What you can do in order to make sure that the main content is not covered by the footer? You need to make sure that there is enough margin at the bottom of your body element, which is the container for anything you put there. Enough margin means enough for the content to finish before the footer starts.

First, you need to measure the height of the footer. Use the developer tools to do that.



Measure Footer Height With Developer Tools

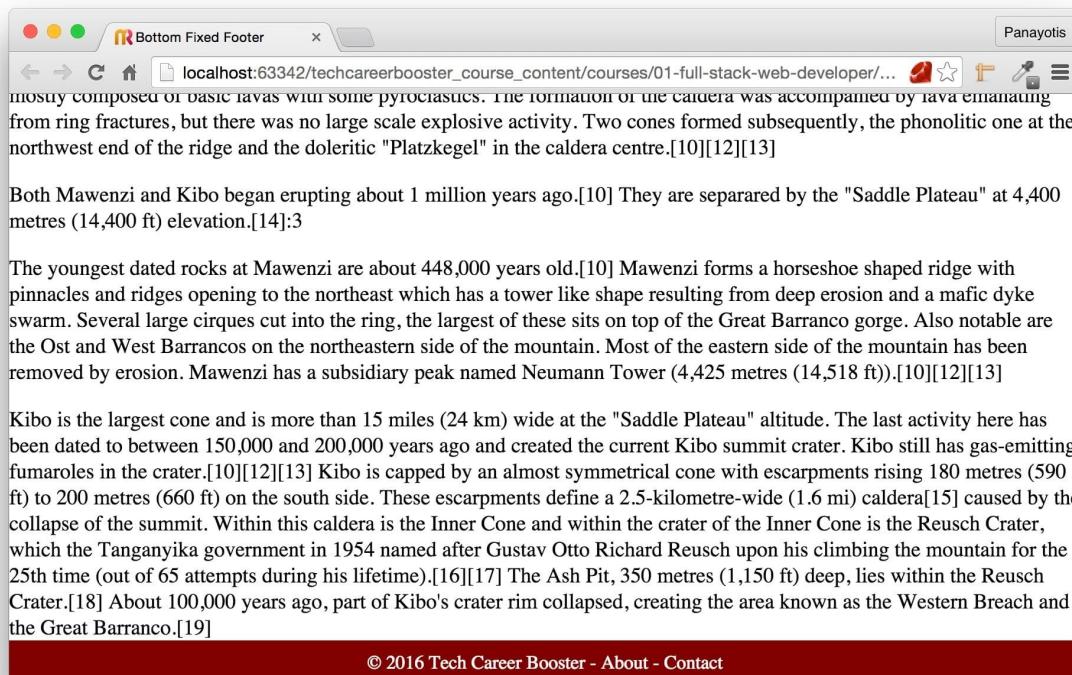
As you can see, the footer height is 38px. (10 for padding top, 10 for padding bottom and 18 for the actual content height). So, if you pick up a bottom margin for the body equal to that number, then you are good to see the whole content of your page. It will not be covered by the footer.

Let's add the CSS rule:

```
1 * {
2     box-sizing: border-box;
3 }
4 body {
5     margin: 0 0 38px;
6 }
7
8 #footer {
9     text-align: center;
10    background-color: Maroon;
11    color: White;
12    padding: 10px 0;
13    position: fixed;
14    bottom: 0;
15    width: 100%;
16 }
17
18 a {
19     text-decoration: none;
20     color: inherit;
21 }
22
23 #main-content {
24     font-size: 18px;
25 }
```

(the above code snippet online)

As you can see above, we have changed the `margin` value for `body` so that the margin bottom has the value `38px`. Let's save that and reload the page.



With Bottom Margin On Body

If you reload your page and scroll at the bottom, you will, now, be able to see the last part of the content, which was previously hidden by the footer. Now, the footer does not hide it anymore, because the `body` element, which contains the main content, now has a `margin-bottom` equal to the value of the height of the footer.

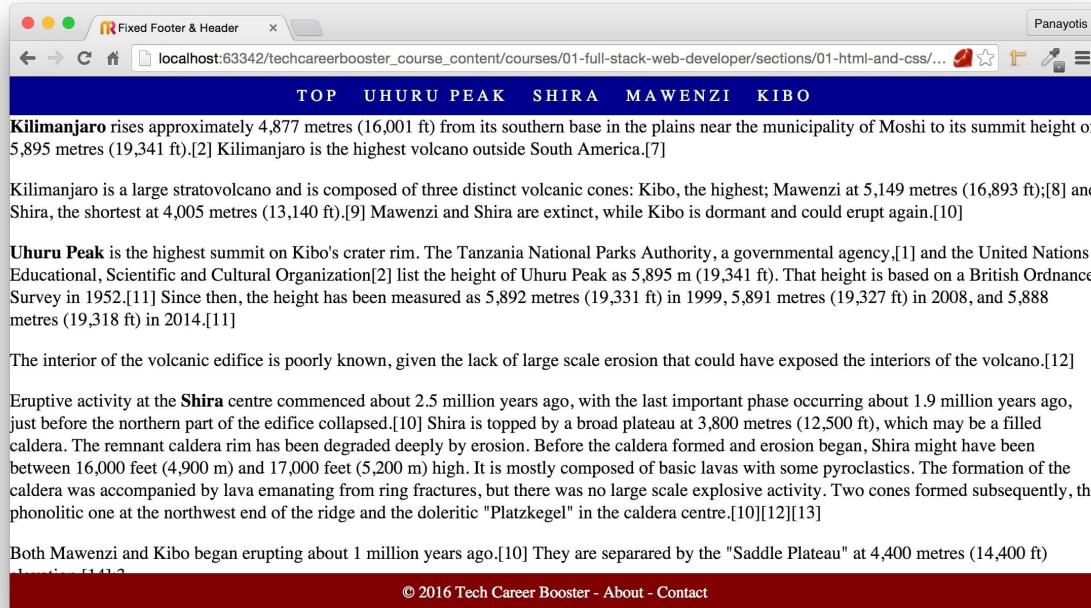
`top, left, right`

Except from the `bottom` property, we can use the `top`, `left` and `right` properties to specify the exact position of the element we want to position on our page.

`top`

The `top` is used to set the position of an element by specifying the position of its top margin relative to the top margin of the browser window.

Let's enhance our last HTML page in order to include a top navigation bar. This is the sample page that we want to build:



Page With Top and Bottom Nav Bar

This page has a top and bottom navigation bar. When the user scrolls to visit the bottom of the content, both bars remain on their position. Also, the start and bottom of text are not hidden by the nav bars respectively. There is also one more property: when we click on the top nav bar links the text scrolls automatically at that position.

Watch this video in order to understand how this page works:

[Page with top and bottom nav bar](#)

Let's start with the HTML having the top navigation bar content:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Fixed Footer & Header</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div id="header">
11             <a href="#start">Top</a>
12             <a href="#uhuru-peak">Uhuru Peak</a>
13             <a href="#shira">Shira</a>
14             <a href="#mawenzi">Mawenzi</a>
15             <a href="#kibo">Kibo</a>
16         </div>

```

```
17
18     <div id="footer">
19         © 2016 Tech Career Booster - <a href="#">About</a> - <a href="#">Conta\
20 ct</a>
21     </div>
22
23
24     <div id="main-content">
25         <p id="start" class="bookmark"><strong>Kilimanjaro</strong> rises approxima\
26 tely 4,877 metres (16,001 ft) from its southern base in the plains near the munic\
27 ipality of Moshi to its summit height of 5,895 metres
28         (19,341 ft). [2] Kilimanjaro is the highest volcano outside South America.\\
29 [7]
30     </p>
31
32     <p>
33         Kilimanjaro is a large stratovolcano and is composed of three distinct volc\
34 anic cones: Kibo, the highest; Mawenzi at 5,149 metres (16,893 ft); [8] and Shira,
35         the shortest at 4,005 metres (13,140 ft). [9] Mawenzi and Shira are extinct,\\
36 while Kibo is dormant and could erupt again. [10]
37     </p>
38
39     <p id="uhuru-peak" class="bookmark">
40         <strong>Uhuru Peak</strong> is the highest summit on Kibo's crater rim. The\
41 Tanzania National Parks Authority, a governmental agency, [1] and the United Nati\
42 ons Educational,
43         Scientific and Cultural Organization [2] list the height of Uhuru Peak as 5,\\
44 895 m (19,341 ft). That height is based on a British Ordnance Survey in 1952. [11]\\
45 Since
46         then, the height has been measured as 5,892 metres (19,331 ft) in 1999, 5,8\
47 91 metres (19,327 ft) in 2008, and 5,888 metres (19,318 ft) in 2014. [11]
48     </p>
49
50     <p>
51         The interior of the volcanic edifice is poorly known, given the lack of lar\
52 ge scale erosion that could have exposed the interiors of the volcano. [12]
53     </p>
54
55     <p id="shira" class="bookmark">
56         Eruptive activity at the <strong>Shira</strong> centre commenced about 2.5 \
57 million years ago, with the last important phase occurring about 1.9 million year\
58 s ago, just before
59         the northern part of the edifice collapsed. [10] Shira is topped by a broad \
60 plateau at 3,800 metres (12,500 ft), which may be a filled caldera. The remnant c\
61 aldera
62         rim has been degraded deeply by erosion. Before the caldera formed and eros\
```

```
63 ion began, Shira might have been between 16,000 feet (4,900 m) and 17,000 feet (5\
64 ,200 m)
65     high. It is mostly composed of basic lavas with some pyroclastics. The form\
66 ation of the caldera was accompanied by lava emanating from ring fractures, but t\
67 here was
68     no large scale explosive activity. Two cones formed subsequently, the phono\
69 litic one at the northwest end of the ridge and the doleritic "Platzkegel" in the\
70 caldera centre.[10][12][13]
71     </p>
72
73     <p>
74     Both Mawenzi and Kibo began erupting about 1 million years ago.[10] They ar\
75 e separated by the "Saddle Plateau" at 4,400 metres (14,400 ft) elevation.[14]:3
76     </p>
77
78     <p id="mawenzi" class="bookmark">
79     The youngest dated rocks at <strong>Mawenzi</strong> are about 448,000 year\
80 s old.[10] Mawenzi forms a horseshoe shaped ridge with pinnacles and ridges openi\
81 ng to the northeast which
82     has a tower like shape resulting from deep erosion and a mafic dyke swarm. \
83 Several large cirques cut into the ring, the largest of these sits on top of the \
84 Great
85     Barranco gorge. Also notable are the Ost and West Barrancos on the northeas\
86 tern side of the mountain. Most of the eastern side of the mountain has been remo\
87 ved by
88     erosion. Mawenzi has a subsidiary peak named Neumann Tower (4,425 metres (1\
89 4,518 ft)).[10][12][13]
90     </p>
91
92     <p id="kibo" class="bookmark">
93     <strong>Kibo</strong> is the largest cone and is more than 15 miles (24 km)\\
94 wide at the "Saddle Plateau" altitude. The last activity here has been dated to \
95 between 150,000
96     and 200,000 years ago and created the current Kibo summit crater. Kibo stil\
97 l has gas-emitting fumaroles in the crater.[10][12][13] Kibo is capped by an
98     almost symmetrical cone with escarpments rising 180 metres (590 ft) to 200 \
99 metres (660 ft) on the south side. These escarpments define a 2.5-kilometre-wide \
100 (1.6 mi)
101     caldera[15] caused by the collapse of the summit. Within this caldera is th\
102 e Inner Cone and within the crater of the Inner Cone is the Reusch Crater, which
103     the Tanganyika government in 1954 named after Gustav Otto Richard Reusch up\
104 on his climbing the mountain for the 25th time (out of 65 attempts during his lif\
105 etime).[16][17]
106     The Ash Pit, 350 metres (1,150 ft) deep, lies within the Reusch Crater.[18]\\
107 About 100,000 years ago, part of Kibo's crater rim collapsed, creating the area \
108 known as
```

```
109     the Western Breach and the Great Barranco.[19]  
110     </p>  
111     </div>  
112  
113     </body>  
114  
115 </html>
```

(the above code snippet online)

The differences, if compared to the previous HTML page content, are:

1. We have the div with id `header` that has the 4 links that function as page navigation links.
2. Each link on the top navigation bar is a link to a bookmark, position inside the same page itself.
3. Inside the `main-content` div, there are the corresponding menu link targets. The targets are `p` elements with the correct `id` value (`start`, `uhuru-peak`, `shira`, `mawenzi`, `kibo`).
4. Besides the `id` attribute, the `p` elements that are targets of the links, they have a class `bookmark` that will be used to style them. See later on about this.

Update your CSS file to be as follows:

```
1 * {  
2     box-sizing: border-box;  
3 }  
4  
5 body {  
6     margin: 38px 0 38px;  
7 }  
8  
9 #footer {  
10    text-align: center;  
11    background-color: Maroon;  
12    color: White;  
13    padding: 10px 0;  
14    position: fixed;  
15    bottom: 0;  
16    width: 100%;  
17 }  
18  
19 a {  
20    text-decoration: none;  
21    color: inherit;  
22 }  
23  
24 #main-content {
```

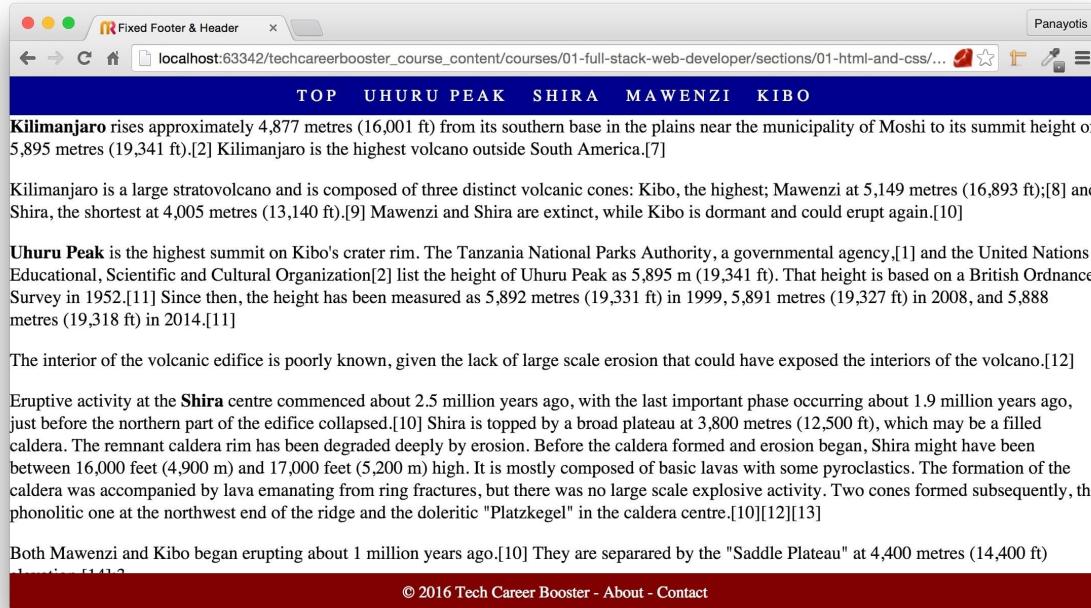
```
25     font-size: 18px;
26 }
27
28 #header {
29     background-color: darkblue;
30     color: white;
31     padding: 10px 0;
32     text-align: center;
33     position: fixed;
34     top: 0;
35     width: 100%;
36 }
37
38 #header a {
39     padding: 10px 10px;
40     text-transform: uppercase;
41     letter-spacing: 4px;
42 }
43
44 #header a:hover {
45     background-color: white;
46     color: darkblue;
47 }
```

(the above code snippet online)

Things to note here are the following:

1. We have added a top margin to body element like we have for bottom. It is 38px height and it will serve the same purpose like the one bottom does. Will allow the start part of the text to be visible and not hidden by the top navigation bar.
2. The #header CSS selector applies rules for the top navigation bar. They are almost the same like the ones for the #footer except from the colors that we are using. Note that it is important here to specify the top property with value 0. If it is not specified, then it has the default value auto, and in that case, the browser automatically calculates the position of the element. The automatic position is not always the best and depends on other parts of the HTML content. Hence, if you omit the top: 0; setting, you will see that the top nav bar is not placed at the top.
3. We have some styling for the header anchor links. See the text-transform property, which is used to make all characters of the links uppercase. Also, see the letter-spacing property which is used to allow for same space in between the characters of the words of the links.

If you save the above files and load your page, you will see this:



Page With Top and Bottom Nav Bar

which is what we want as a result. There is only one problem, that you will encounter if you click on the links on the top navigation bar. The page scrolls to the target positions (`p`) but, the start of each `p`, is actually hidden behind the top navigation bar.

In order to solve that problem, we need to do the following:

1. Each `p` that is a target of a top navigation link, needs its content appear a little bit below (38px, which is the height of the top nav bar) from the top edge of the `p` block. This can be done if we set `padding-top` equal to 38px. Then...
2. The actual `p` top edge needs to be brought down to recover for the extra space that previous condition (`padding-top`) creates. This is done with a negative `margin-top` equal to the value of `padding-top`, i.e. -38px.

Let's add, first, the `padding-top` with value 38px and see what will happen. We will set this rule with the selector `.bookmark`, since our target `p` have this class attached.

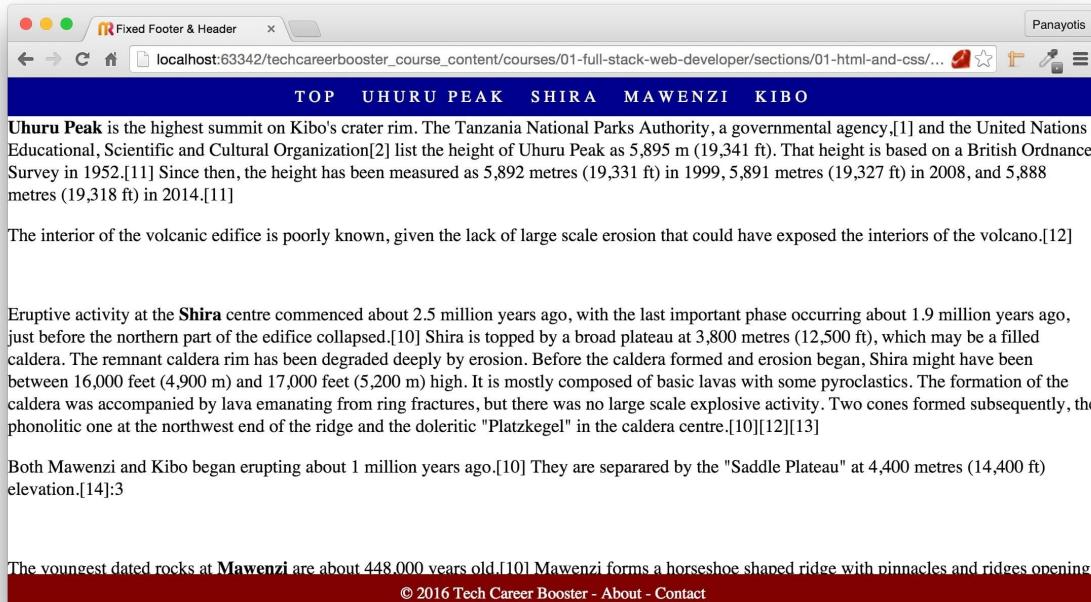
Set your CSS to the following:

```
1 * {
2     box-sizing: border-box;
3 }
4 body {
5     margin: 38px 0 38px;
6 }
7
8 #footer {
9     text-align: center;
10    background-color: Maroon;
11    color: White;
12    padding: 10px 0;
13    position: fixed;
14    bottom: 0;
15    width: 100%;
16 }
17
18 a {
19     text-decoration: none;
20     color: inherit;
21 }
22
23 #main-content {
24     font-size: 18px;
25 }
26
27 #header {
28     background-color: darkblue;
29     color: white;
30     padding: 10px 0;
31     text-align: center;
32     position: fixed;
33     top: 0;
34     width: 100%;
35 }
36
37 #header a {
38     padding: 10px 10px;
39     text-transform: uppercase;
40     letter-spacing: 4px;
41 }
42
43 #header a:hover {
44     background-color: white;
45     color: darkblue;
46 }
```

```
47  
48 .bookmark {  
49   padding-top: 38px;  
50 }
```

(the above code snippet online)

Save the file and reload the page. Then try to navigate to the 2nd anchor, for example:



Links Scroll To The Correct Position

As you can see, now, anchors on top navigation bar scroll to the correct position. And the start of the corresponding target paragraph is not hidden by the top navigation bar. The padding-top: 38px; has solved the problem. However, it has created another one. Can you see that, now, not all paragraphs have the same blank space in between?

Uhuru Peak is the highest summit on Kibo's crater rim. The Tanzania National Parks Authority, a governmental agency,[1] and the United Nations Educational, Scientific and Cultural Organization[2] list the height of Uhuru Peak as 5,895 m (19,341 ft). That height is based on a British Ordnance Survey in 1952.[11] Since then, the height has been measured as 5,892 metres (19,331 ft) in 1999, 5,891 metres (19,327 ft) in 2008, and 5,888 metres (19,318 ft) in 2014.[11]

Shira is poorly known, given the lack of large scale erosion that could have exposed the interiors of the volcano.[12]

Eruptive activity at the **Shira** centre commenced about 2.5 million years ago, with the last important phase occurring about 1.9 million years ago, just before the northern part of the edifice collapsed.[10] Shira is topped by a broad plateau at 3,800 metres (12,500 ft), which may be a filled caldera. The remnant caldera rim has been degraded deeply by erosion. Before the caldera formed and erosion began, Shira might have been between 16,000 feet (4,900 m) and 17,000 feet (5,200 m) high. It is mostly composed of basic lavas with some pyroclastics. The formation of the caldera was accompanied by lava emanating from ring fractures, but there was no large scale explosive activity. Two cones formed subsequently, the

© 2016 Tech Career Booster - About - Contact

Elements Network Sources Timeline Profiles Resources Audits Console

Styles Computed Event Listeners DOM Breakpoints Properties

Filter

element.style { }

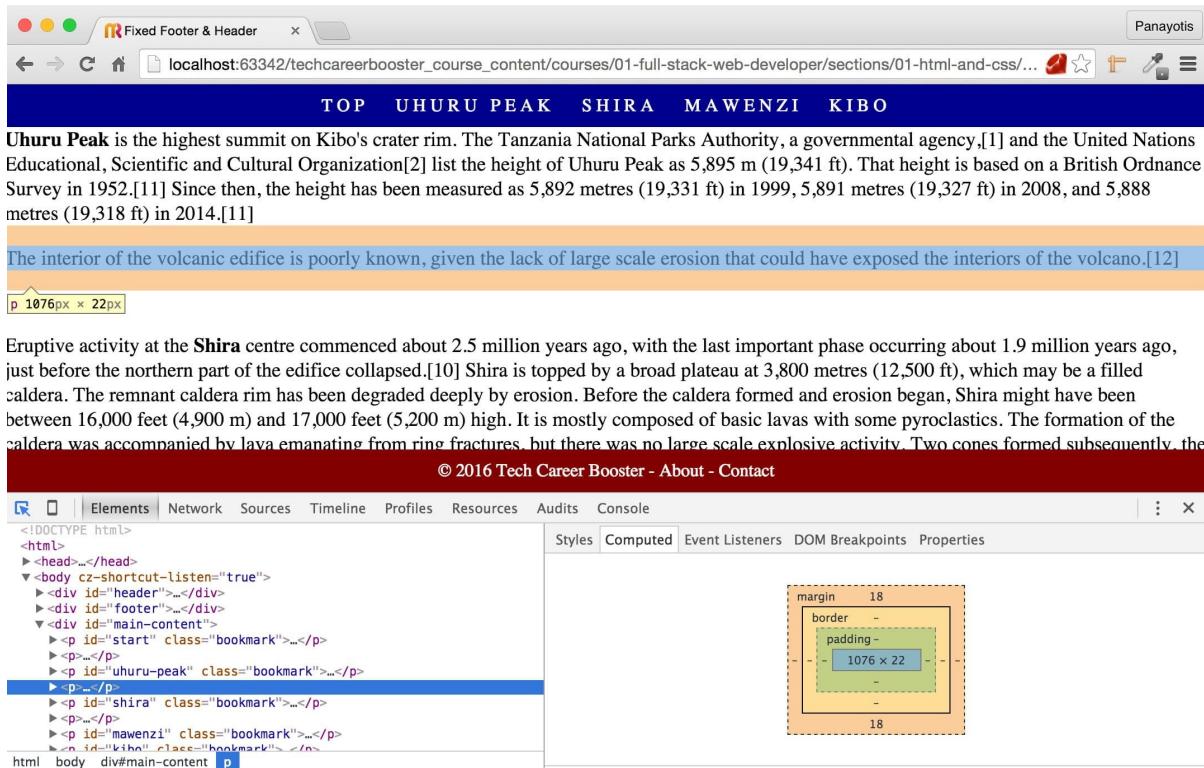
.bookmark { padding-top: 38px; } bottom-fixed-footer.css:48

* { box-sizing: border-box; } bottom-fixed-footer.css:1

p { display: block; -webkit-margin-before: 1em; } user agent stylesheet

Big Blank Space Before Bookmark Paragraphs

The fact that we have added `padding-top: 38px;` to some of the paragraphs (the `p` with class `bookmark`), has created this discrepancy. This is the space that we want to decrease back to its original value. Equal to the space we have to the other `p` elements in the documents.



The Normal Space Between Paragraphs

We want all the `p` elements, bookmark and not, to have equal space in between. We will do that by setting the `margin-top` of the bookmark `p` to a negative value, `-38px`.

Hence, the CSS now has as follows:

```

1  * {
2      box-sizing: border-box;
3  }
4  body {
5      margin: 38px 0 38px;
6  }
7
8  #footer {
9      text-align: center;
10     background-color: Maroon;
11     color: White;
12     padding: 10px 0;
13     position: fixed;
14     bottom: 0;
15     width: 100%;
16 }
17
18 a {
19     text-decoration: none;
20     color: inherit;

```

```
21 }
22
23 #main-content {
24     font-size: 18px;
25 }
26
27 #header {
28     background-color: darkblue;
29     color: white;
30     padding: 10px 0;
31     text-align: center;
32     position: fixed;
33     top: 0;
34     width: 100%;
35 }
36
37 #header a {
38     padding: 10px 10px;
39     text-transform: uppercase;
40     letter-spacing: 4px;
41 }
42
43 #header a:hover {
44     background-color: white;
45     color: darkblue;
46 }
47
48 .bookmark {
49     padding-top: 38px;
50     margin-top: -38px;
51 }
```

(the above code snippet online)

If you save the above CSS and reload your page, you will see that the space between the paragraphs has been recovered:

Kilimanjaro rises approximately 4,877 metres (16,001 ft) from its southern base in the plains near the municipality of Moshi to its summit height of 5,895 metres (19,341 ft).^[2] Kilimanjaro is the highest volcano outside South America.^[7]

Kilimanjaro is a large stratovolcano and is composed of three distinct volcanic cones: Kibo, the highest; Mawenzi at 5,149 metres (16,893 ft);^[8] and Shira, the shortest at 4,005 metres (13,140 ft).^[9] Mawenzi and Shira are extinct, while Kibo is dormant and could erupt again.^[10]

Uhuru Peak is the highest summit on Kibo's crater rim. The Tanzania National Parks Authority, a governmental agency,^[1] and the United Nations Educational, Scientific and Cultural Organization^[2] list the height of Uhuru Peak as 5,895 m (19,341 ft). That height is based on a British Ordnance Survey in 1952.^[11] Since then, the height has been measured as 5,892 metres (19,331 ft) in 1999, 5,891 metres (19,327 ft) in 2008, and 5,888 metres (19,318 ft) in 2014.^[11]

The interior of the volcanic edifice is poorly known, given the lack of large scale erosion that could have exposed the interiors of the volcano.^[12]

Eruptive activity at the **Shira** centre commenced about 2.5 million years ago, with the last important phase occurring about 1.9 million years ago, just before the northern part of the edifice collapsed.^[10] Shira is topped by a broad plateau at 3,800 metres (12,500 ft), which may be a filled caldera. The remnant caldera rim has been degraded deeply by erosion. Before the caldera formed and erosion began, Shira might have been between 16,000 feet (4,900 m) and 17,000 feet (5,200 m) high. It is mostly composed of basic lavas with some pyroclastics. The formation of the caldera was accompanied by lava emanating from ring fractures, but there was no large scale explosive activity. Two cones formed subsequently, the phonolitic one at the northwest end of the ridge and the doleritic "Platzkegel" in the caldera centre.^{[10][12][13]}

Both Mawenzi and Kibo began erupting about 1 million years ago.^[10] They are separated by the "Saddle Plateau" at 4,400 metres (14,400 ft) elevation.^{[14]:3}

The youngest dated rocks at **Mawenzi** are about 448,000 years old.^[10] Mawenzi forms a horseshoe shaped ridge with pinnacles and ridges opening to the northeast which has a tower like shape resulting from deep erosion and a mafic dyke swarm. Several large cirques cut into the ring, the largest of these sits on top of the Great Barranco gorge. Also notable are the Ost and West Barrancos on the northeastern side of the mountain. Most of the

© 2016 Tech Career Booster - About - Contact

Space Between Paragraphs Has Been Recovered

left

We have seen how we can use `bottom` and `top` to position elements relative to the browser window bottom and top sides. Similarly, we can use the property `left` to position an element relative to the left side of the browser window.

Let's suppose that we want to create the following page:

Kilimanjaro rises approximately 4,877 metres (16,001 ft) from its southern base in the plains near the municipality of Moshi to its summit height of 5,895 metres (19,341 ft).^[2] Kilimanjaro is the highest volcano outside South America.^[7]

Kilimanjaro is a large stratovolcano and is composed of three distinct volcanic cones: Kibo, the highest; Mawenzi at 5,149 metres (16,893 ft);^[8] and Shira, the shortest at 4,005 metres (13,140 ft).^[9] Mawenzi and Shira are extinct, while Kibo is dormant and could erupt again.^[10]

Uhuru Peak is the highest summit on Kibo's crater rim. The Tanzania National Parks Authority, a governmental agency, [1] and the United Nations Educational, Scientific and Cultural Organization^[2] list the height of Uhuru Peak as 5,895 m (19,341 ft). That height is based on a British Ordnance Survey in 1952.^[11] Since then, the height has been measured as 5,892 metres (19,331 ft) in 1999, 5,891 metres (19,327 ft) in 2008, and 5,888 metres (19,318 ft) in 2014.^[11]

The interior of the volcanic edifice is poorly known, given the lack of large scale erosion that could have exposed the interiors of the volcano.^[12]

Eruptive activity at the **Shira** centre commenced about 2.5 million years ago, with the last important phase occurring about 1.9 million years ago, just before the northern part of the edifice collapsed.^[10] Shira is topped by a broad plateau at 3,800 metres (12,500 ft), which may be a filled caldera. The remnant caldera rim has been degraded deeply by erosion. Before the caldera formed and erosion began, Shira might have been between 16,000 feet (4,900 m) and 17,000 feet (5,200 m) high. It is mostly composed of basic lavas with some pyroclastics. The formation of the caldera was accompanied by lava emanating from ring fractures, but there was no large scale explosive activity. Two cones formed subsequently, the phonolitic one at the northwest end of the ridge and the doleritic "Platzkegel" in the caldera centre.^{[10][12][13]}

© 2016 Tech Career Booster - About - Contact

Page With Fixed Image On The Left

As you can see above, there is an image to the left of the text. If you scroll or resize the browser window, the image stays on the same position. Watch this video to understand what I mean:

A left positioned image, fixed when scrolling

This page is not much different to the previous one that we have created. Starting from the HTML, we only have to include the part of the image. Here is the full new HTML content:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Fixed Footer & Header</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div id="header">
11             <a href="#start">Top</a>
12             <a href="#uhuru-peak">Uhuru Peak</a>
13             <a href="#shira">Shira</a>
14             <a href="#mawenzi">Mawenzi</a>
15             <a href="#kibo">Kibo</a>
16         </div>
17

```

```
18     <div id="footer">
19         &copy; 2016 Tech Career Booster - <a href="#">About</a> - <a href="#">Conta\
20 ct</a>
21     </div>
22
23
24     <div id="main-content">
25         <p id="start" class="bookmark"><strong>Kilimanjaro</strong> rises approxima\
26 tely 4,877 metres (16,001 ft) from its southern base in the plains near the munic\
27 ipality of Moshi to its summit height of 5,895 metres
28             (19,341 ft).[2] Kilimanjaro is the highest volcano outside South America.\\
29 [7]
30         </p>
31
32         <p>
33             Kilimanjaro is a large stratovolcano and is composed of three distinct volc\
34 anic cones: Kibo, the highest; Mawenzi at 5,149 metres (16,893 ft);[8] and Shira,
35             the shortest at 4,005 metres (13,140 ft).[9] Mawenzi and Shira are extinct,\\
36 while Kibo is dormant and could erupt again.[10]
37         </p>
38
39         <p id="uhuru-peak" class="bookmark">
40             <strong>Uhuru Peak</strong> is the highest summit on Kibo's crater rim. The\
41 Tanzania National Parks Authority, a governmental agency,[1] and the United Nati\
42 ons Educational,
43             Scientific and Cultural Organization[2] list the height of Uhuru Peak as 5,\\
44 895 m (19,341 ft). That height is based on a British Ordnance Survey in 1952.[11]\\
45 Since
46             then, the height has been measured as 5,892 metres (19,331 ft) in 1999, 5,8\
47 91 metres (19,327 ft) in 2008, and 5,888 metres (19,318 ft) in 2014.[11]
48         </p>
49
50         <p>
51             The interior of the volcanic edifice is poorly known, given the lack of lar\
52 ge scale erosion that could have exposed the interiors of the volcano.[12]
53         </p>
54
55         <p id="shira" class="bookmark">
56             Eruptive activity at the <strong>Shira</strong> centre commenced about 2.5 \
57 million years ago, with the last important phase occurring about 1.9 million year\
58 s ago, just before
59             the northern part of the edifice collapsed.[10] Shira is topped by a broad \
60 plateau at 3,800 metres (12,500 ft), which may be a filled caldera. The remnant c\
61 aldera
62             rim has been degraded deeply by erosion. Before the caldera formed and eros\
63 ion began, Shira might have been between 16,000 feet (4,900 m) and 17,000 feet (5\
```

64 ,200 m)
65 high. It is mostly composed of basic lavas with some pyroclastics. The form\
66 ation of the caldera was accompanied by lava emanating from ring fractures, but t\
67 here was
68 no large scale explosive activity. Two cones formed subsequently, the phono\
69 litic one at the northwest end of the ridge and the doleritic "Platzkegel" in the\
70 caldera centre.[10][12][13]
71 </p>
72
73 <p>
74 Both Mawenzi and Kibo began erupting about 1 million years ago.[10] They ar\
75 e separated by the "Saddle Plateau" at 4,400 metres (14,400 ft) elevation.[14]:3
76 </p>
77
78 <p id="mawenzi" class="bookmark">
79 The youngest dated rocks at Mawenzi are about 448,000 year\
80 s old.[10] Mawenzi forms a horseshoe shaped ridge with pinnacles and ridges openi\
81 ng to the northeast which
82 has a tower like shape resulting from deep erosion and a mafic dyke swarm. \
83 Several large cirques cut into the ring, the largest of these sits on top of the \
84 Great
85 Barranco gorge. Also notable are the Ost and West Barrancos on the northeas\
86 tern side of the mountain. Most of the eastern side of the mountain has been remo\
87 ved by
88 erosion. Mawenzi has a subsidiary peak named Neumann Tower (4,425 metres (1\
89 4,518 ft)).[10][12][13]
90 </p>
91
92 <p id="kibo" class="bookmark">
93 Kibo is the largest cone and is more than 15 miles (24 km)\
94 wide at the "Saddle Plateau" altitude. The last activity here has been dated to \
95 between 150,000
96 and 200,000 years ago and created the current Kibo summit crater. Kibo stil\
97 l has gas-emitting fumaroles in the crater.[10][12][13] Kibo is capped by an
98 almost symmetrical cone with escarpments rising 180 metres (590 ft) to 200 \
99 metres (660 ft) on the south side. These escarpments define a 2.5-kilometre-wide \
100 (1.6 mi)
101 caldera[15] caused by the collapse of the summit. Within this caldera is th\
102 e Inner Cone and within the crater of the Inner Cone is the Reusch Crater, which
103 the Tanganyika government in 1954 named after Gustav Otto Richard Reusch up\
104 on his climbing the mountain for the 25th time (out of 65 attempts during his lif\
105 etime).[16][17]
106 The Ash Pit, 350 metres (1,150 ft) deep, lies within the Reusch Crater.[18]\
107 About 100,000 years ago, part of Kibo's crater rim collapsed, creating the area \
108 known as
109 the Western Breach and the Great Barranco.[19]

```
110      </p>
111  </div>
112
113  <div id="k-image">
114    
115  </div>
116</body>
117
118</html>
```

(the above code snippet online)

We have only added one more div at the bottom of the HTML content, before the closing body tag. It is the div that holds a reference to an image. The image is the file images/kilimanjaro.jpg that exists inside the images folder of this chapter.

In order to position the image div on the left side of the content, we had to bring the content a little bit to the right. Here is the final CSS file:

```
1  * {
2    box-sizing: border-box;
3  }
4  body {
5    margin: 38px 0 38px;
6  }
7
8 #footer {
9   text-align: center;
10  background-color: Maroon;
11  color: White;
12  padding: 10px 0;
13  position: fixed;
14  bottom: 0;
15  width: 100%;
16 }
17
18 a {
19   text-decoration: none;
20   color: inherit;
21 }
22
23 #main-content {
24   font-size: 18px;
25   margin-left: 135px;
26 }
27
28 #header {
```

```

29     background-color: darkblue;
30     color: white;
31     padding: 10px 0;
32     text-align: center;
33     position: fixed;
34     top: 0;
35     width: 100%;
36 }
37
38 #header a {
39     padding: 10px 10px;
40     text-transform: uppercase;
41     letter-spacing: 4px;
42 }
43
44 #header a:hover {
45     background-color: white;
46     color: darkblue;
47 }
48
49 .bookmark {
50     padding-top: 38px;
51     margin-top: -38px;
52 }
53
54 #k-image {
55     position: fixed;
56     top: 65px;
57     left: 10px;
58     padding: 5px 5px;
59     border: 1px solid black;
60     line-height: 0;
61 }
```

(the above code snippet online)

We have done the following enhancements with regards to the previous one, in order to achieve the final desired result:

1. we have set a left margin for the div that contains the main content (`margin-left: 135px;`)
2. we have set the rules for the div that contains the image.
 1. `position: fixed;` so that we can fix the position of the div relative to browser window.
 2. `top: 65px;` to make sure it is drawn a little bit below the top edge of the window.
 3. `left: 10px;` to make sure that it is drawn a little bit next to the left side of the window, but not exactly at the left side.
 4. `line-height: 0;` to make sure that the image does not have extra space below that. (See the chapter “CSS Box Model”, on the line height part).

Nice! Save and load your page and you will see how we have managed to create the page that we had initially planned.

right

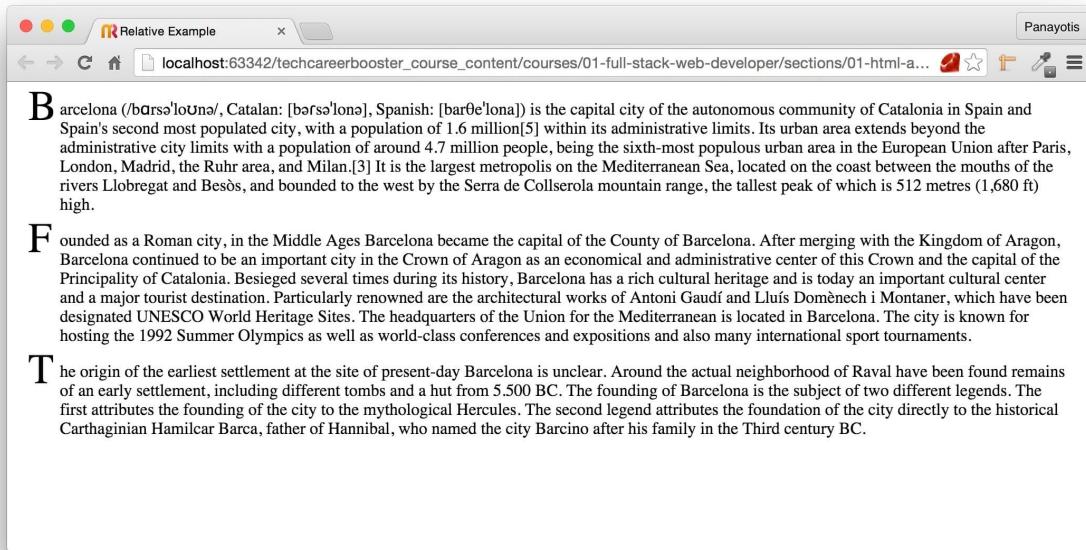
I assume that you now understand what the `right` property does. It positions the element relative to the right edge of the browser window.

Important: Statically positioned elements, i.e. those that have `position` equal to `static` or they do not have any `position` value, hence using the default one which is `static`, are not affected by the positional properties `top`, `bottom`, `left` or `right`.

Positioning Element Relative To Their Normal Position

Some times, we want to position an element on specific position, using `top`, `bottom`, `left` and `right` properties, but relative to their normal position and not relative to the browser window. In order to do that, we need to set the `position` property to have the value `relative`.

Let's suppose that we want to implement the following HTML page:



Page with Paragraphs First Char Stand Out

As you can see above:

1. The page has a series of paragraphs with the same style all of them.
2. The content of the page has some free space on the left and on the right.
3. The first character of each paragraph stands out to the left. The left side of the content of the page has enough room for the first characters of each paragraph to be drawn inside it.

Since, the content has paragraphs, let's start with them. We will create, first, an HTML page that has the paragraphs with the text:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Relative Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <p>
12             Barcelona (/bərəsəloñə/, Catalan: [bəsəlonə], Spanish: [baɾθelona]) is \
13             the capital city of the autonomous community of Catalonia in Spain and Spain's se\
14             cond most
15                 populated city, with a population of 1.6 million[5] within its administrati\
16             ve limits. Its urban area extends beyond the administrative city limits with a po\
17             pulation
18                 of around 4.7 million people, being the sixth-most populous urban area in t\
19             he European Union after Paris, London, Madrid, the Ruhr area, and Milan.[3] It is\
20             the largest
21                 metropolis on the Mediterranean Sea, located on the coast between the mouth\
22             s of the rivers Llobregat and Besòs, and bounded to the west by the Serra de Coll\
23             serola mountain
24                 range, the tallest peak of which is 512 metres (1,680 ft) high.
25         </p>
26
27         <p>
28             Founded as a Roman city, in the Middle Ages Barcelona became the capital of\
29             the County of Barcelona. After merging with the Kingdom of Aragon, Barcelona con\
30             tinued to be an
31                 important city in the Crown of Aragon as an economical and administrative c\
32             enter of this Crown and the capital of the Principality of Catalonia. Besieged se\
33             veral times during
34                 its history, Barcelona has a rich cultural heritage and is today an importa\
35             nt cultural center and a major tourist destination. Particularly renowned are the\
36             architectural works
37                 of Antoni Gaudí and Lluís Domènech i Montaner, which have been designated U\
38             NESCO World Heritage Sites. The headquarters of the Union for the Mediterranean i\
39             s located in
40                 Barcelona. The city is known for hosting the 1992 Summer Olympics as well a\
41             s world-class conferences and expositions and also many international sport tourna\
42             ments.
43         </p>
44
45         <p>
46             The origin of the earliest settlement at the site of present-day Barcelona \
```

```
47 is unclear. Around the actual neighborhood of Raval have been found remains of an\
48 early
49     settlement, including different tombs and a hut from 5.500 BC. The founding\
50 of Barcelona is the subject of two different legends. The first attributes the
51     founding of the city to the mythological Hercules. The second legend attrib\
52 utes the foundation of the city directly to the historical Carthaginian Hamilcar \
53 Barca,
54     father of Hannibal, who named the city Barcino after his family in the Thir\
55 d century BC.
56     </p>
57 </body>
58
59 </html>
```

(the above code snippet online)

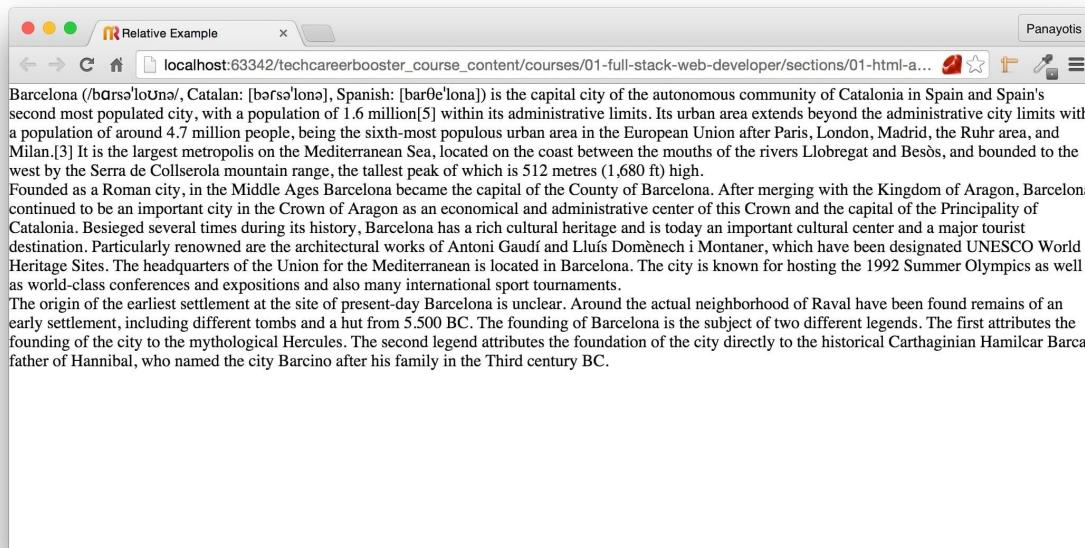
We start with a very simple CSS:

```
1 * {
2     box-sizing: border-box;
3     margin: 0;
4     padding: 0;
5 }
```

(the above code snippet online)

which basically starts with some sensitive defaults. We will enhance the rules further down the implementation.

If you save the above files and reload your page, you will see this:



Page With Paragraphs - No Particular Styling

As you can see, the paragraphs are one next to each other and there is no space left and right to the page content.

My approach here would be first to style the first character of each paragraph. This is because I want to see how the rest of the content will be affected. So, how can I style part of my text? We have learned that we can do that using the `span` element.

Here is the new HTML content:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Relative Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <p>
12             <span class="starting-paragraph-character">B</span>arcelona (/bərsəloñə/, \
13             Catalan: [bəsəlonə], Spanish: [barθełona]) is the capital city of the autonom\
14             ous community of Catalonia in Spain and Spain's second most
15                 populated city, with a population of 1.6 million[5] within its administrati\
16                 ve limits. Its urban area extends beyond the administrative city limits with a po\
17                 pulation
18                     of around 4.7 million people, being the sixth-most populous urban area in t\
19                     he European Union after Paris, London, Madrid, the Ruhr area, and Milan.[3] It is\

```

```
20 the largest
21     metropolis on the Mediterranean Sea, located on the coast between the mouth\
22 s of the rivers Llobregat and Besòs, and bounded to the west by the Serra de Coll\
23 serola mountain
24     range, the tallest peak of which is 512 metres (1,680 ft) high.
25 </p>
26
27 <p>
28     <span class="starting-paragraph-character">F</span>ounded as a Roman city, \
29 in the Middle Ages Barcelona became the capital of the County of Barcelona. After\
30 merging with the Kingdom of Aragon, Barcelona continued to be an
31     important city in the Crown of Aragon as an economical and administrative c\
32 enter of this Crown and the capital of the Principality of Catalonia. Besieged se\
33 veral times during
34     its history, Barcelona has a rich cultural heritage and is today an importa\
35 nt cultural center and a major tourist destination. Particularly renowned are the\
36 architectural works
37     of Antoni Gaudí and Lluís Domènech i Montaner, which have been designated U\
38 NESCO World Heritage Sites. The headquarters of the Union for the Mediterranean i\
39 s located in
40     Barcelona. The city is known for hosting the 1992 Summer Olympics as well a\
41 s world-class conferences and expositions and also many international sport tourn\
42 aments.
43 </p>
44
45 <p>
46     <span class="starting-paragraph-character">T</span>he origin of the earlies\
47 t settlement at the site of present-day Barcelona is unclear. Around the actual n\
48 eighborhood of Raval have been found remains of an early
49     settlement, including different tombs and a hut from 5.500 BC. The founding\
50 of Barcelona is the subject of two different legends. The first attributes the
51     founding of the city to the mythological Hercules. The second legend attrib\
52 utes the foundation of the city directly to the historical Carthaginian Hamilcar \
53 Barca,
54     father of Hannibal, who named the city Barcino after his family in the Thir\
55 d century BC.
56 </p>
57 </body>
58
59 </html>
```

(the above code snippet online)

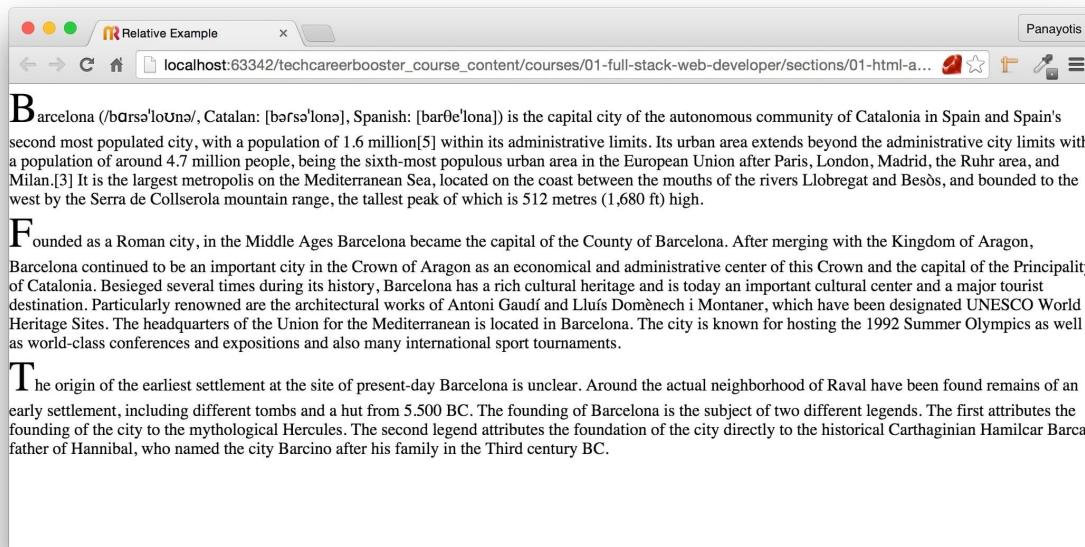
As you can see above, I have wrapped the first character of each paragraph with a span and I have given that span a class that I can later on use to style them all together. If you save the above content and reload your page, you will not see any particular change. Because the span does not affect the HTML layout or style by itself.

Let's add a CSS rule that will make those characters stand out. Let's try with setting the font size to 40px.

```
1 * {
2     box-sizing: border-box;
3     margin: 0;
4     padding: 0;
5 }
6
7 .starting-paragraph-character {
8     font-size: 40px;
9 }
```

(the above code snippet online)

If you save this and reload your page, you will see this:



Increased Font Size for First Character

Things are getting better. The fact that we have increased the font size now makes the paragraphs stand out from each other.

Let's now create the left space and the right space. The left space will be used to position the first character of each paragraph. Well, that's easy. We just increase the left and right margin of the body element.

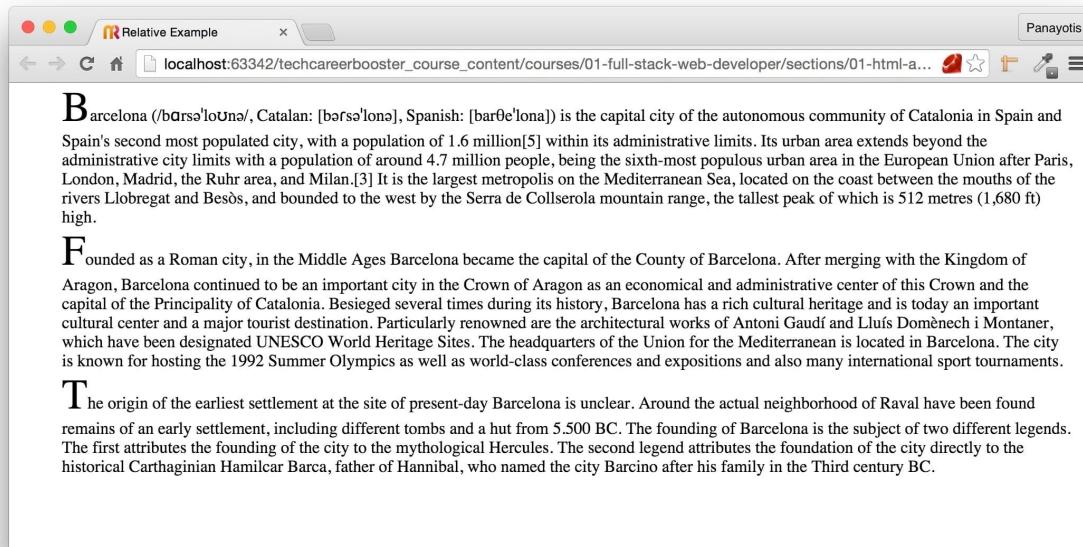
```

1 * {
2     box-sizing: border-box;
3     margin: 0;
4     padding: 0;
5 }
6
7 body {
8     margin: 0 20px 0 50px;
9 }
10
11 .starting-paragraph-character {
12     font-size: 40px;
13 }

```

(the above code snippet online)

As you can see above, we just gave a margin left equal to 50px and a margin right equal to 20px. The left margin is enough to hold the 40px size of the first character of each paragraph. Let's save this file and reload our page. We will see this:



Increased the Left Side to Have Enough Room for First Character

Now, we will move the first character of each paragraph to the left of its current position. This is where the `position` property with value `relative` will come handy. What we want to do is to position the character some distance to the left of its current position.

Moving to the left means a negative distance value. We will try `-30px` for the `left` property. Let's do that. Enhance your CSS as follows:

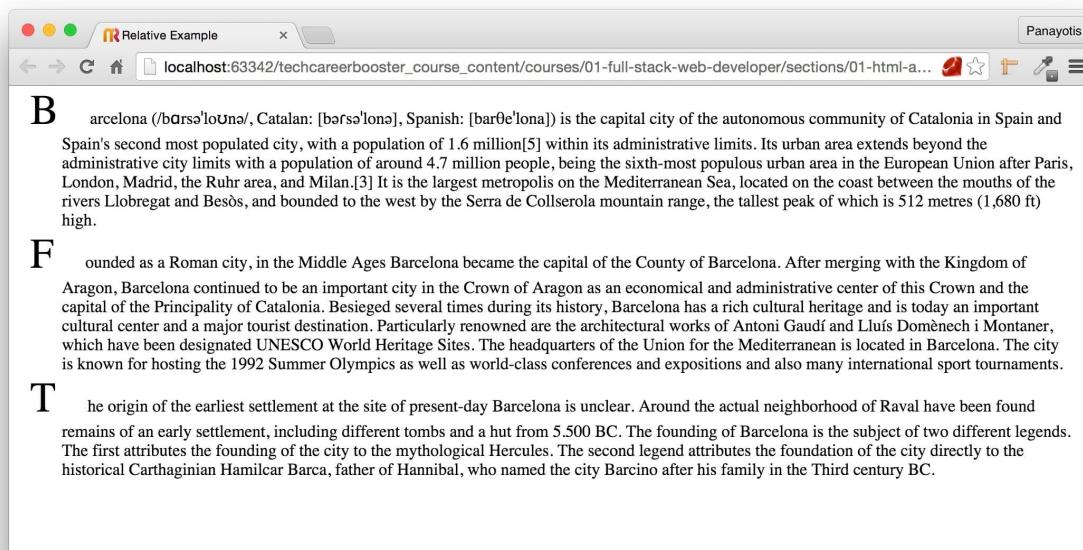
```

1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7  body {
8      margin: 0 20px 0 50px;
9  }
10
11 .starting-paragraph-character {
12     font-size: 40px;
13     position: relative;
14     left: -30px;
15 }

```

(the above code snippet online)

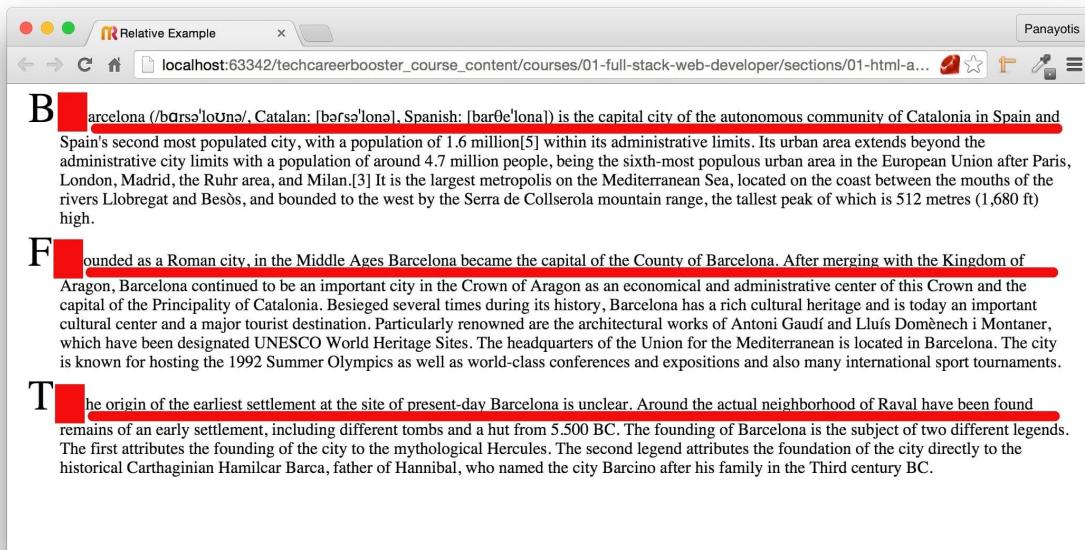
We have added the `position: relative;` and `left: -30px;` CSS rules. Let's save that and reload our page. The result will be:



Move First Character to The Left

It worked. The `left` position is relative to the actual position the character would have been. It does not refer to the left side of the browser window. And this is because of the `position: relative;`

However, the current implementation has some problems:



Original First Character Space Not Reused

As you can see in the above picture, the originally occupied first character space is not reused. Also, the distance between the first and second line of each paragraph is bigger than the distance between the rest of the lines.

One way we can remedy these two problems is to remove the first character from the `p` element and put it before that. Let's try that:

```

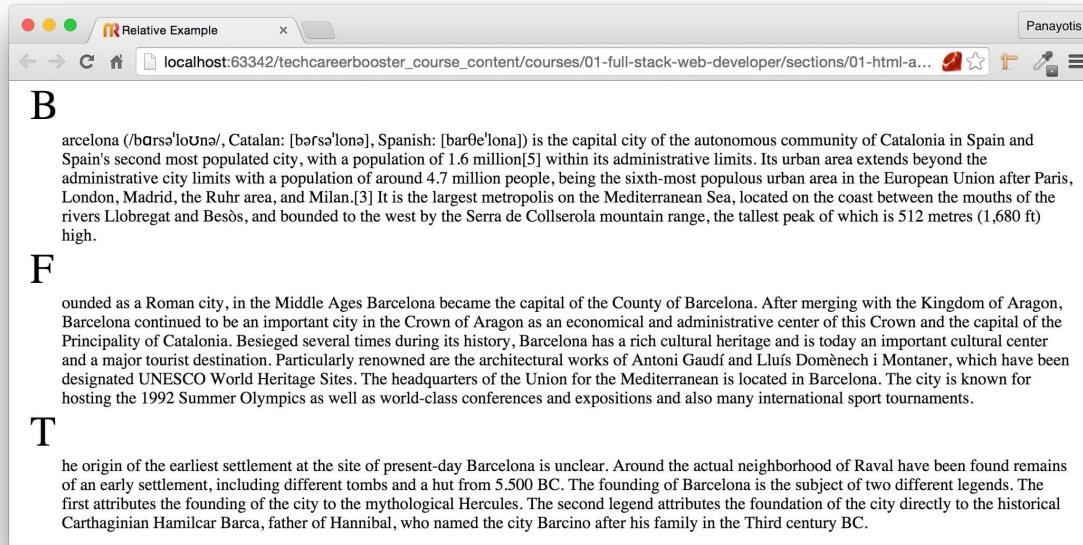
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Relative Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <span class="starting-paragraph-character">B</span>
12         <p>
13             arcelona (/bərˈsələnə/, Catalan: [bərsə'lona], Spanish: [barθeˈlona]) is t\
14             he capital city of the autonomous community of Catalonia in Spain and Spain's sec\
15             ond most
16                 populated city, with a population of 1.6 million[5] within its administrati\
17                 ve limits. Its urban area extends beyond the administrative city limits with a po\
18                 pulation
19                     of around 4.7 million people, being the sixth-most populous urban area in t\
20                     he European Union after Paris, London, Madrid, the Ruhr area, and Milan.[3] It is\
21                     the largest

```

```
22      metropolis on the Mediterranean Sea, located on the coast between the mouth\
23 s of the rivers Llobregat and Besòs, and bounded to the west by the Serra de Coll\
24 serola mountain
25      range, the tallest peak of which is 512 metres (1,680 ft) high.
26  </p>
27
28  <span class="starting-paragraph-character">F</span>
29  <p>
30      ounded as a Roman city, in the Middle Ages Barcelona became the capital of \
31 the County of Barcelona. After merging with the Kingdom of Aragon, Barcelona cont\
32 inued to be an
33      important city in the Crown of Aragon as an economical and administrative c\
34 enter of this Crown and the capital of the Principality of Catalonia. Besieged se\
35 veral times during
36      its history, Barcelona has a rich cultural heritage and is today an importa\
37 nt cultural center and a major tourist destination. Particularly renowned are the\
38 architectural works
39      of Antoni Gaudí and Lluís Domènech i Montaner, which have been designated U\
40 NESCO World Heritage Sites. The headquarters of the Union for the Mediterranean i\
41 s located in
42      Barcelona. The city is known for hosting the 1992 Summer Olympics as well a\
43 s world-class conferences and expositions and also many international sport tourn\
44 aments.
45  </p>
46
47  <span class="starting-paragraph-character">T</span>
48  <p>
49      he origin of the earliest settlement at the site of present-day Barcelona i\
50 s unclear. Around the actual neighborhood of Raval have been found remains of an \
51 early
52      settlement, including different tombs and a hut from 5.500 BC. The founding\
53 of Barcelona is the subject of two different legends. The first attributes the
54      founding of the city to the mythological Hercules. The second legend attrib\
55 utes the foundation of the city directly to the historical Carthaginian Hamilcar \
56 Barca,
57      father of Hannibal, who named the city Barcino after his family in the Thir\
58 d century BC.
59  </p>
60  </body>
61
62 </html>
```

(the above code snippet online)

Without doing any change on our CSS file, let's save the above HTML content and reload our page. We will see this:



Character Before Paragraph

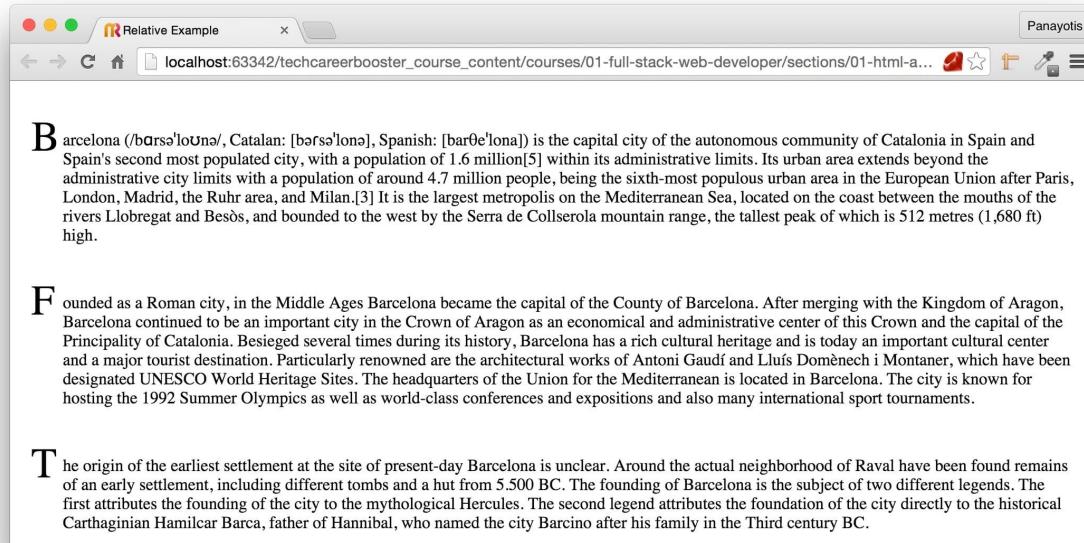
Ok. Good. The lines of the paragraph are now displayed ok. The problem is that the character has moved above the level of the paragraph. What we can do is to change its top position. Bring it down so that it stands on the level we want. Let's do that:

```

1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7  body {
8      margin: 0 20px 0 50px;
9  }
10
11 .starting-paragraph-character {
12     font-size: 40px;
13     position: relative;
14     left: -30px;
15     top: 30px;
16 }
```

(the above code snippet online)

We have added the 30px top property. This will bring the character down to the level we want. Let's save that and reload the page:



The screenshot shows a web browser window with the title "Relative Example". The address bar displays "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-developer/sections/01-html-a...". The content area contains three paragraphs of text:

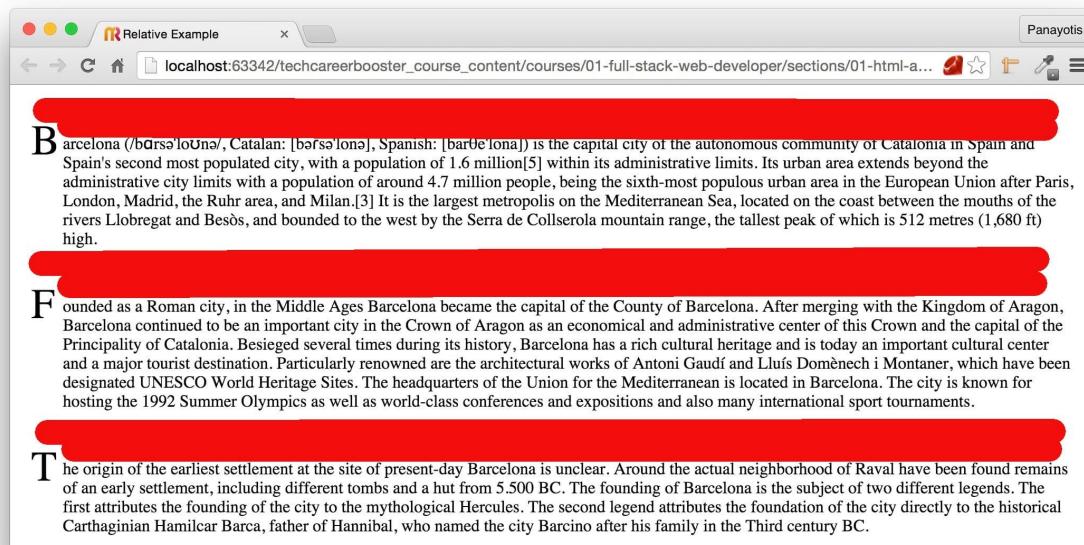
Barcelona (/bərˈsəlōnə/, Catalan: [bərˈsəlonə], Spanish: [barθeˈlona]) is the capital city of the autonomous community of Catalonia in Spain and Spain's second most populated city, with a population of 1.6 million[5] within its administrative limits. Its urban area extends beyond the administrative city limits with a population of around 4.7 million people, being the sixth-most populous urban area in the European Union after Paris, London, Madrid, the Ruhr area, and Milan.[3] It is the largest metropolis on the Mediterranean Sea, located on the coast between the mouths of the rivers Llobregat and Besòs, and bounded to the west by the Serra de Collserola mountain range, the tallest peak of which is 512 metres (1,680 ft) high.

Founded as a Roman city, in the Middle Ages Barcelona became the capital of the County of Barcelona. After merging with the Kingdom of Aragon, Barcelona continued to be an important city in the Crown of Aragon as an economical and administrative center of this Crown and the capital of the Principality of Catalonia. Besieged several times during its history, Barcelona has a rich cultural heritage and is today an important cultural center and a major tourist destination. Particularly renowned are the architectural works of Antoni Gaudí and Lluís Domènech i Montaner, which have been designated UNESCO World Heritage Sites. The headquarters of the Union for the Mediterranean is located in Barcelona. The city is known for hosting the 1992 Summer Olympics as well as world-class conferences and expositions and also many international sport tournaments.

The origin of the earliest settlement at the site of present-day Barcelona is unclear. Around the actual neighborhood of Raval have been found remains of an early settlement, including different tombs and a hut from 5.500 BC. The founding of Barcelona is the subject of two different legends. The first attributes the founding of the city to the mythological Hercules. The second legend attributes the foundation of the city directly to the historical Carthaginian Hamilcar Barca, father of Hannibal, who named the city Barcino after his family in the Third century BC.

Bring First Character Down

There is one more problem left to solve. There is a lot of white space above the paragraphs:



The screenshot shows a web browser window with the title "Relative Example". The address bar displays "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-developer/sections/01-html-a...". The content area contains three paragraphs of text, each preceded by a large amount of white space:

Barcelona (/bərˈsəlōnə/, Catalan: [bərˈsəlonə], Spanish: [barθeˈlona]) is the capital city of the autonomous community of Catalonia in Spain and Spain's second most populated city, with a population of 1.6 million[5] within its administrative limits. Its urban area extends beyond the administrative city limits with a population of around 4.7 million people, being the sixth-most populous urban area in the European Union after Paris, London, Madrid, the Ruhr area, and Milan.[3] It is the largest metropolis on the Mediterranean Sea, located on the coast between the mouths of the rivers Llobregat and Besòs, and bounded to the west by the Serra de Collserola mountain range, the tallest peak of which is 512 metres (1,680 ft) high.

Founded as a Roman city, in the Middle Ages Barcelona became the capital of the County of Barcelona. After merging with the Kingdom of Aragon, Barcelona continued to be an important city in the Crown of Aragon as an economical and administrative center of this Crown and the capital of the Principality of Catalonia. Besieged several times during its history, Barcelona has a rich cultural heritage and is today an important cultural center and a major tourist destination. Particularly renowned are the architectural works of Antoni Gaudí and Lluís Domènech i Montaner, which have been designated UNESCO World Heritage Sites. The headquarters of the Union for the Mediterranean is located in Barcelona. The city is known for hosting the 1992 Summer Olympics as well as world-class conferences and expositions and also many international sport tournaments.

The origin of the earliest settlement at the site of present-day Barcelona is unclear. Around the actual neighborhood of Raval have been found remains of an early settlement, including different tombs and a hut from 5.500 BC. The founding of Barcelona is the subject of two different legends. The first attributes the founding of the city to the mythological Hercules. The second legend attributes the foundation of the city directly to the historical Carthaginian Hamilcar Barca, father of Hannibal, who named the city Barcino after his family in the Third century BC.

A Lot Of White Space Above Paragraphs

This is because of the first character original position. There are some alternatives that we can use to solve that. Let's try one. We will wrap both the character and the paragraph into their own div so that we can manipulate them as a group:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Relative Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div class="paragraph-container">
12             <span class="starting-paragraph-character">B</span>
13             <p>
14                 arcelona (/bərələnə/, Catalan: [bərləlonə], Spanish: [barθełona]) is \
15                 the capital city of the autonomous community of Catalonia in Spain and Spain's s\
16                 econd most
17                 populated city, with a population of 1.6 million[5] within its administra\
18                 tive limits. Its urban area extends beyond the administrative city limits with a \
19                 population
20                 of around 4.7 million people, being the sixth-most populous urban area in\
21                 the European Union after Paris, London, Madrid, the Ruhr area, and Milan.[3] It \
22                 is the largest
23                 metropolis on the Mediterranean Sea, located on the coast between the mou\
24                 ths of the rivers Llobregat and Besòs, and bounded to the west by the Serra de Co\
25                 llserola mountain
26                 range, the tallest peak of which is 512 metres (1,680 ft) high.
27             </p>
28         </div>
29
30         <div class="paragraph-container">
31             <span class="starting-paragraph-character">F</span>
32             <p>
33                 ounded as a Roman city, in the Middle Ages Barcelona became the capital o\
34                 f the County of Barcelona. After merging with the Kingdom of Aragon, Barcelona co\
35                 ntinued to be an
36                 important city in the Crown of Aragon as an economical and administrative\
37                 center of this Crown and the capital of the Principality of Catalonia. Besieged \
38                 several times during
39                 its history, Barcelona has a rich cultural heritage and is today an impor\
40                 tant cultural center and a major tourist destination. Particularly renowned are t\
41                 he architectural works
42                 of Antoni Gaudí and Lluís Domènech i Montaner, which have been designated\
43                 UNESCO World Heritage Sites. The headquarters of the Union for the Mediterranean\
44                 is located in
45                 Barcelona. The city is known for hosting the 1992 Summer Olympics as well\
46                 as world-class conferences and expositions and also many international sport tou\
```

```
47 rnaments.  
48     </p>  
49 </div>  
50  
51     <div class="paragraph-container">  
52         <span class="starting-paragraph-character">T</span>  
53         <p>  
54             he origin of the earliest settlement at the site of present-day Barcelona\\  
55             is unclear. Around the actual neighborhood of Raval have been found remains of a\\  
56             n early  
57                 settlement, including different tombs and a hut from 5.500 BC. The foundi\\  
58             ng of Barcelona is the subject of two different legends. The first attributes the  
59                 founding of the city to the mythological Hercules. The second legend attr\\  
60             ibutes the foundation of the city directly to the historical Carthaginian Hamilca\\  
61             r Barca,  
62                 father of Hannibal, who named the city Barcino after his family in the Th\\  
63             ird century BC.  
64         </p>  
65     </div>  
66 </body>  
67  
68 </html>
```

(the above code snippet online)

As you can see above, we have grouped characters and their corresponding paragraphs into groups, using a div. Each group now has the class paragraph-container which we can use to style. Actually, we will remove some of the top margin by applying a negative value. Here is our CSS:

```
1 * {  
2     box-sizing: border-box;  
3     margin: 0;  
4     padding: 0;  
5 }  
6  
7 body {  
8     margin: 0 20px 10px 50px;  
9 }  
10  
11 .paragraph-container {  
12     margin-top: -30px;  
13 }  
14  
15 .starting-paragraph-character {  
16     position: relative;  
17     font-size: 40px;
```

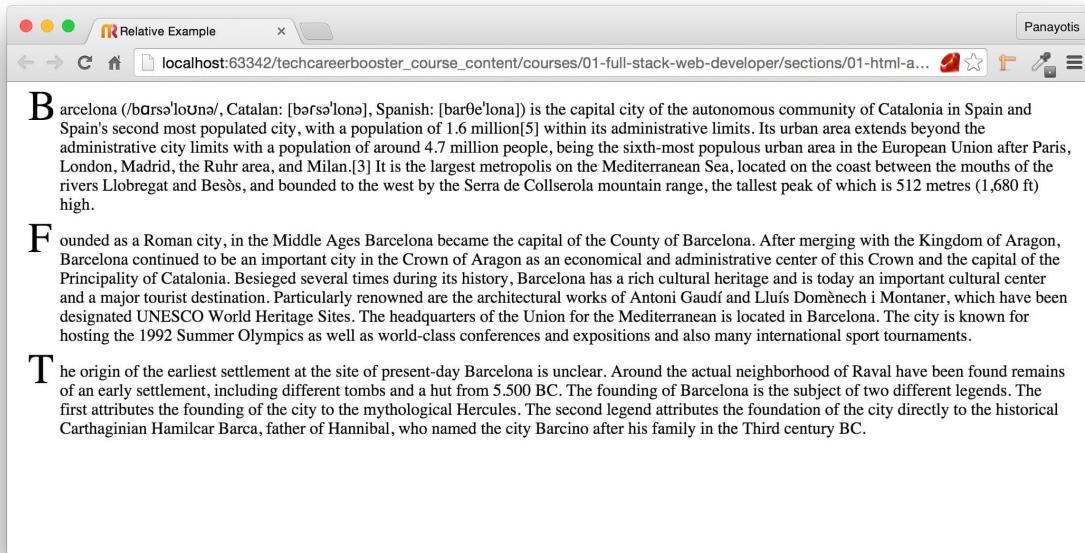
```

18     left: -30px;
19     top: 30px;
20 }

```

(the above code snippet online)

As you can see we have added the `margin-top: -30px;` for each group with class `.paragraph-container`. If we save the CSS and reload our page, we will see this:



Final Result With Negative Margin On Groups

Well Done! We have managed to bring the HTML page to match our original planned page.

Positioning Elements Relative To Other Element

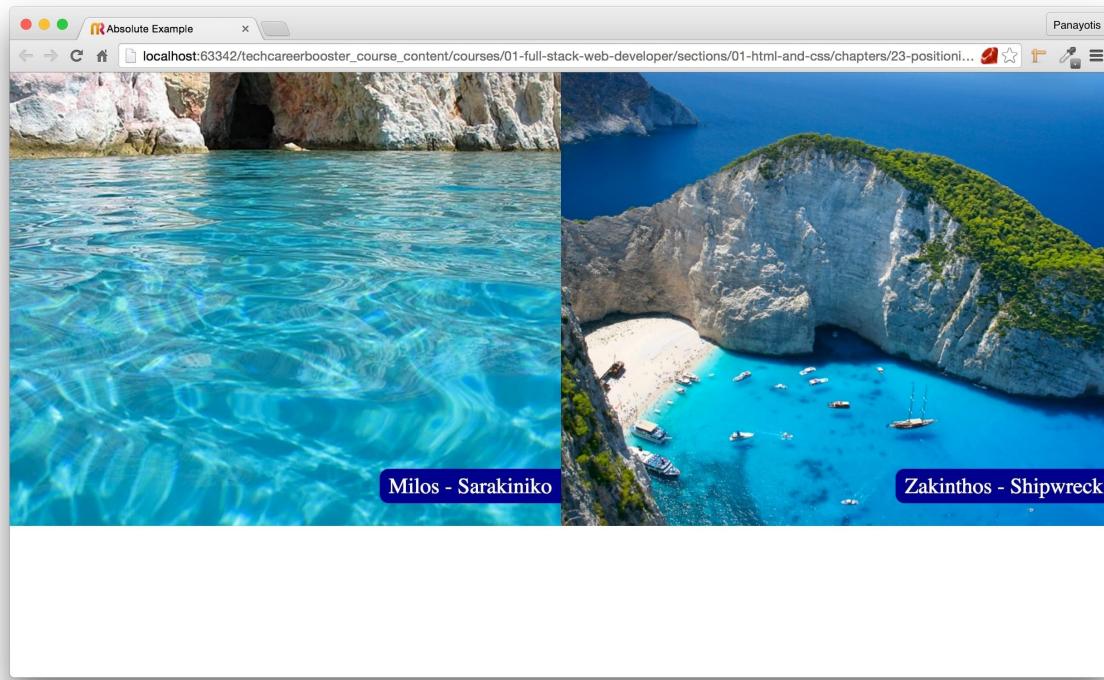
We have learned how to use `top`, `bottom`, `left` and `right` in order to

- position elements relative to the browser window (`position` with value `fixed`)
- position elements relative to their normal position (`position` with value `relative`)

We also know that the default value of `position` is `static`, case in which the `top`, `bottom`, `left` and `right` are not taken into account.

There is one more case that will be useful to us: `position` with value `absolute`. This is used when we want to position an element using the positional properties (`top`, `bottom`, `left` and `right`) but relative to the parent/ancestor element box edges. So, when the positioned element always stays in specific position with regards to another (parent/ancestor) element.

Let's try an example. Assume that we want to implement the following page:



Two Images With Tags

As you can see, there is one tag with the place name at the bottom right of each picture. And, you can understand that this tag, we want to always be displayed at the same position, bottom-right, relative to the parent/ancestor image container.

Let's start with the main HTML content that has the containers for the images.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Absolute Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="left-column" class="column">
12
13             </div>
14
15         <div id="right-column" class="column">
16
17             </div>
```

```
20    </body>
21
22 </html>
```

(the above code snippet online)

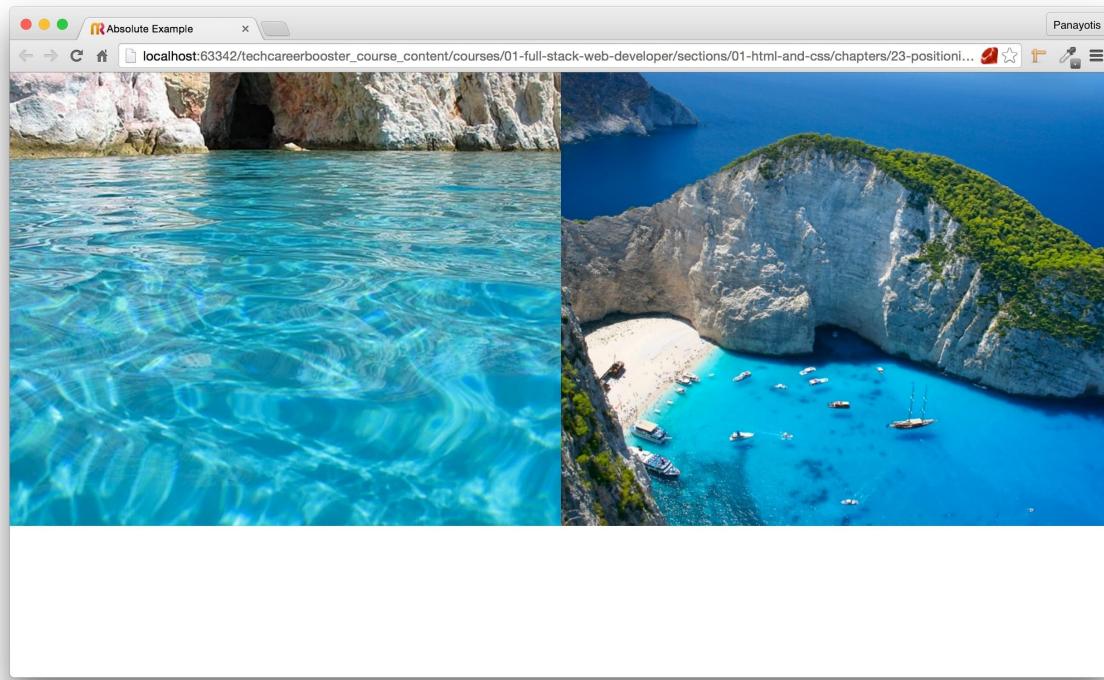
This is very simple. It has one div for the left column and one div for the right column. We only have to apply some CSS rules, including the background image:

```
1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7 .column {
8     width: 50%;
9     height: 500px;
10    background-repeat: no-repeat;
11    background-attachment: scroll;
12    background-position: center;
13    background-size: cover;
14 }
15
16 #left-column {
17     float: left;
18     background-image: url("../images/sarakiniko-water.jpg");
19 }
20
21 #right-column {
22     float: right;
23     background-image: url("../images/zakinthos-ship-wreck.jpg");
24 }
```

(the above code snippet online)

Everything this CSS file includes, we have already learned about.

Save those files and load your page into your browser. The result that you will see is the following:



Page With Images But No Tags Yet

Now, let's put the tags, but without actually caring about their position. We will only care about their color styling. Here is the HTML with the tags:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Absolute Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="left-column" class="column">
12             <span class="tag">
13                 Milos - Sarakiniko
14             </span>
15         </div>
16
17         <div id="right-column" class="column">
18             <span class="tag">
19                 Zakynthos - Shipwreck
20             </span>
21         </div>
```

```
22
23     </body>
24
25 </html>
```

(the above code snippet online)

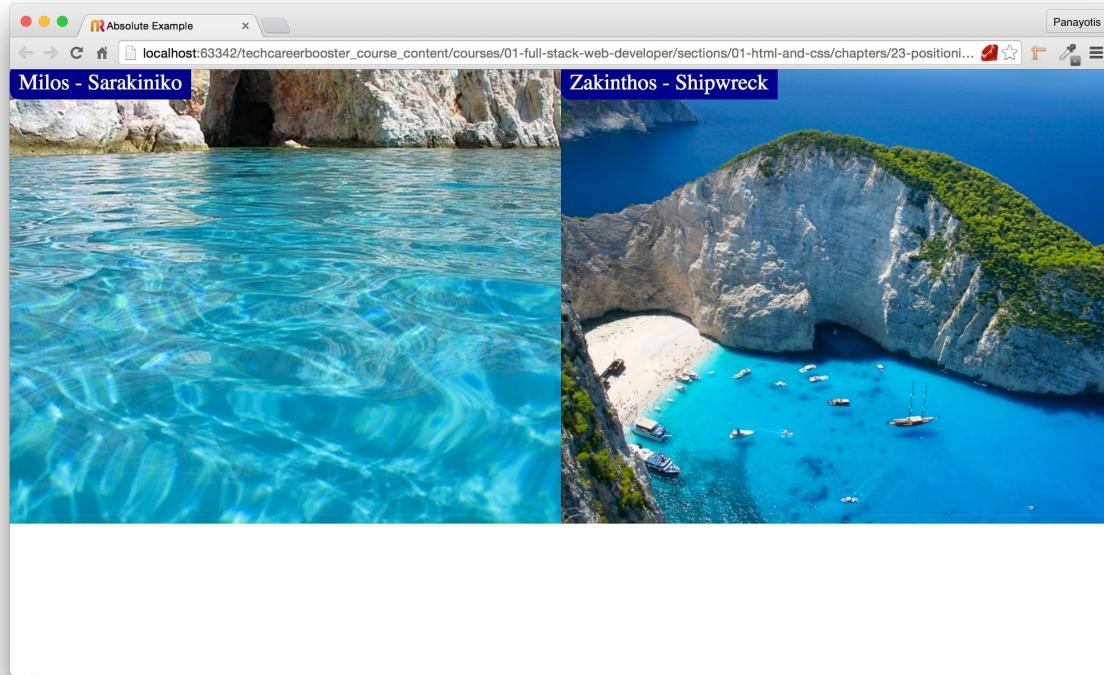
As you can see, we have added the tags as simple `span` elements with a class `tag` that will help us style them.

Let's do the basic styling. Amend the CSS file so that it looks like this:

```
1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7 .column {
8     width: 50%;
9     height: 500px;
10    background-repeat: no-repeat;
11    background-attachment: scroll;
12    background-position: center;
13    background-size: cover;
14 }
15
16 #left-column {
17     float: left;
18     background-image: url("../images/sarakiniko-water.jpg");
19 }
20
21 #right-column {
22     float: right;
23     background-image: url("../images/zakinthos-ship-wreck.jpg");
24 }
25
26 .tag {
27     background-color: darkblue;
28     color: white;
29     font-size: 24px;
30     padding: 5px 10px;
31     border-radius: 10px 0 0 10px;
32 }
```

(the above code snippet online)

If you save and reload your page, you will see that tags are correctly styled, as we want them, but they are not positioned at the required position. They are positioned at their normal position. At top left corner of their container:



Page With Tags On Normal Position

So, the last task is to position the tags at their correct position. Which is the correct position?

- Their right edge needs to be on the right edge of the parent image container.
- Their bottom edge needs to be some distance above the bottom edge of their parent image container.

In order to achieve that:

Since their position depends on the edges of their parent containers, parent container needs to have `position` property with value `relative`, and their own `position` property needs to have the value `absolute`.

Having said that, here is the final CSS content:

```
1 * {
2     box-sizing: border-box;
3     margin: 0;
4     padding: 0;
5 }
6
7 .column {
8     position: relative;
9     width: 50%;
10    height: 500px;
11    background-repeat: no-repeat;
12    background-attachment: scroll;
13    background-position: center;
14    background-size: cover;
15 }
16
17 #left-column {
18     float: left;
19     background-image: url("../images/sarakiniko-water.jpg");
20 }
21
22 #right-column {
23     float: right;
24     background-image: url("../images/zakinthos-ship-wreck.jpg");
25 }
26
27 .tag {
28     position: absolute;
29     right: 0;
30     bottom: 25px;
31     background-color: darkblue;
32     color: white;
33     font-size: 24px;
34     padding: 5px 10px;
35     border-radius: 10px 0 0 10px;
36 }
```

(the above code snippet online)

If you save this file and reload your page, you will see that you will finally have the desired result. If you try to resize your browser window, you will also see that the tags stay fixed on their positions, relative to the image container.

Important Note:

- (1) If you want to use `position: absolute;` in order to position an element relative to its parent element, then the parent element needs to have `position` with one of the

values: `fixed`, `relative`, `absolute`. In other words, it cannot have the value `static` (or `not` value, which corresponds to `static`), i.e. it has to be non-statically positioned.

(2) When you use `position: absolute`; the element is not, simply, positioned relative to its parent, but, in fact, it is positioned relative to its first/nearest non-statically positioned parent. So, if you have the following HTML fragment:

```

1  <div id="parent-1">
2    <div id="child-1-parent-2">
3      <div id="child-2">
4        </div>
5      </div>
6    </div>
```

(the above code snippet online)

you can use `position: absolute`; on `child-1-parent-2` as long as `parent-1` is non-statically positioned. Also, you can use `position: absolute`; for element `child-2` as long as `child-1-parent-2` is non-statically positioned. If `child-1-parent-2` is statically positioned, you can still use `position: absolute`; for element `child-2` as long as `parent-1` is non-statically positioned. In other words, when you have an absolute positioned element, its position is relative to the edges of its first non-statically positioned parent element. > > (3) `body`, by default, has `static` position value. > > (4) If an absolute position element does not have a non-static positioned parent, then its position is relative to `body`. Even if `body`, by default is not a positioned element.

Positioned Elements

We have learned how to position elements on the page, out of the normal page flow, using the `top`, `bottom`, `left` and `right` properties.

- With `position: fixed`; we position elements relative to the browser window edges.
- With `position: relative`; we position elements relative to their normal position.
- With `position: absolute`; we position elements relative to their first non-statically positioned parent edges.

Elements that are using one of `fixed`, `relative` or `absolute` position values, they can be referred to as *positioned elements*.

z-index

Besides positioning elements on the plane (horizontally with `left` and `right`: x-axis, and / or vertically with `top` and `bottom`: y-axis), there is one more property that can be used to position an element on the z-axis, i.e. its stack order. `z-index` is the property that is used to specify the stack order of elements. Elements with lower `z-index` value are moved to the background. Elements with higher `z-index` value are moved to the foreground. Note that `z-index` is only applicable to positioned elements.

Let's start with the following case:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Z-Index</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="one">
12             one
13         </div>
14
15         <div id="two">
16             two
17         </div>
18
19
20     </body>
21
22 </html>
```

(the above code snippet online)

with the following CSS file:

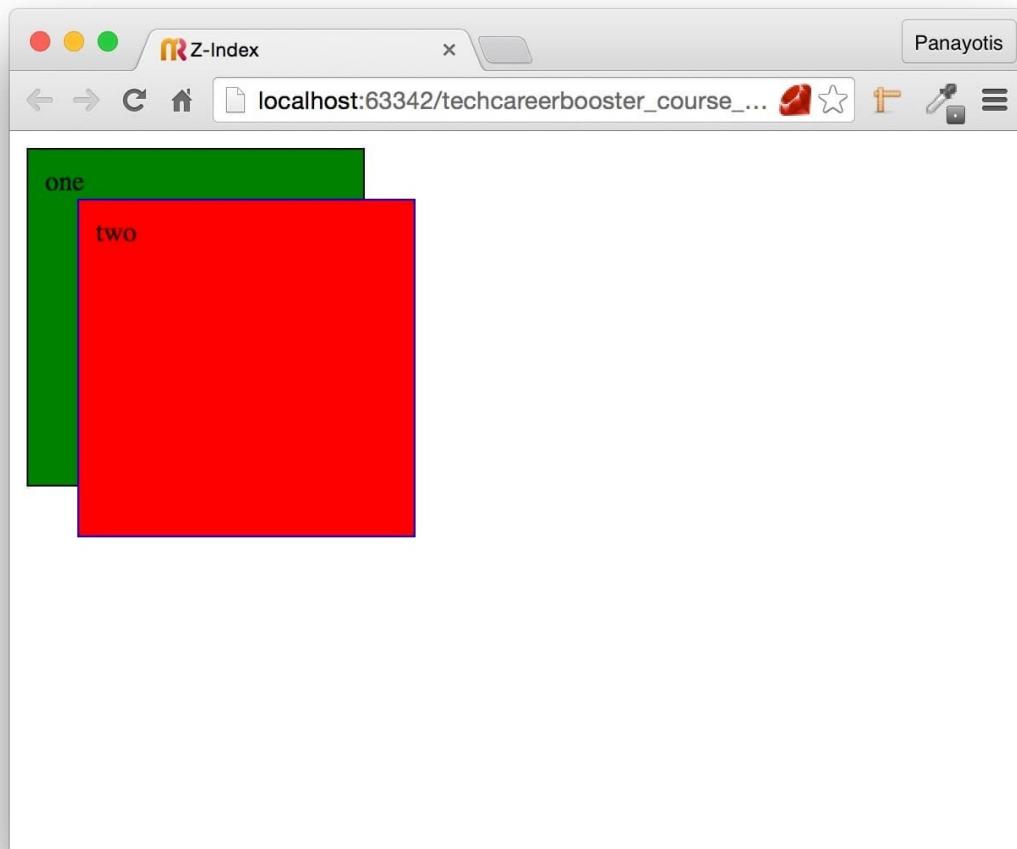
```
1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7 #one {
8     position: absolute;
9     top: 10px;
10    left: 10px;
11    width: 200px;
12    height: 200px;
13    background-color: green;
14    border: 1px solid black;
15    padding: 10px 10px;
16 }
17
18 #two {
19     position: absolute;
20     top: 40px;
```

```
21     left: 40px;  
22     width: 200px;  
23     height: 200px;  
24     background-color: red;  
25     border: 1px solid blue;  
26     padding: 10px 10px;  
27 }
```

(the above code snippet online)

We have absolute positioned elements but we are not using any `z-index` property yet. These are going to be put on the page in the order we define them in the HTML document. Note that these divs are siblings and not one inside the other.

If you save and load this page on your browser, it will be displayed as follows:



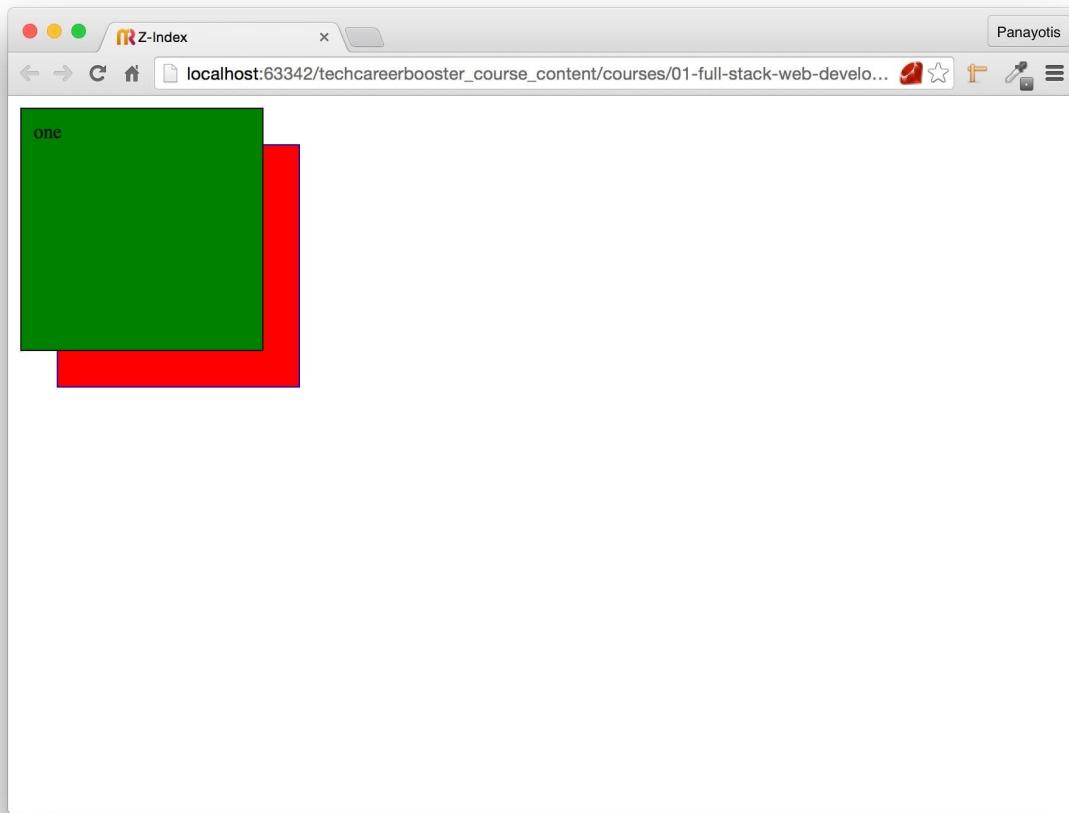
Simple Absolute Elements With No Z-Index

How can we make the div two appear behind div one. We only have to assign a positive `z-index` value to div one. Try to assign `z-index: 1;`: The new CSS file becomes:

```
1 * {
2     box-sizing: border-box;
3     margin: 0;
4     padding: 0;
5 }
6
7 #one {
8     position: absolute;
9     top: 10px;
10    left: 10px;
11    width: 200px;
12    height: 200px;
13    background-color: green;
14    border: 1px solid black;
15    padding: 10px 10px;
16    z-index: 1;
17 }
18
19 #two {
20     position: absolute;
21     top: 40px;
22     left: 40px;
23     width: 200px;
24     height: 200px;
25     background-color: red;
26     border: 1px solid blue;
27     padding: 10px 10px;
28 }
```

(the above code snippet online)

If you save and load your page again, you will see this:



Z-Index 1 On One Div Brings It To Foreground

Note that we could have achieved the same effect if we had set the `z-index` to a negative value for the two div. Try that if you want: Remove the positive value `z-index` from div one and set `z-index` to `-1` for the div two. You will have the same result. Won't you?

Ooooh. Now that we have one being on the foreground and two being on the background, we cannot see the text inside the two div. But I guess that you already know how to fix that. Let's change our HTML content as follows:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Z-Index</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="one">
12             one
```

```
13     </div>
14
15     <div id="two">
16         <span>two</span>
17     </div>
18
19
20 </body>
21
22 </html>
```

(the above code snippet online)

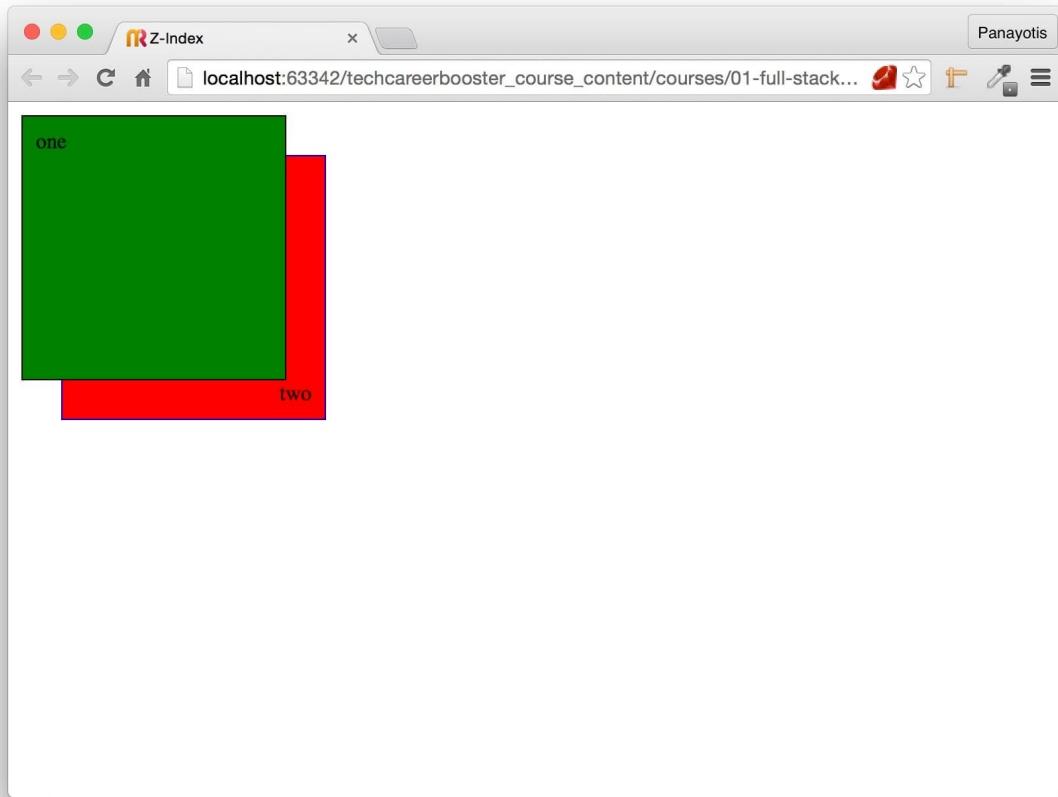
We have wrapped the word two inside a span so that we can set its position. Let's do that in the CSS:

```
1 * {
2     box-sizing: border-box;
3     margin: 0;
4     padding: 0;
5 }
6
7 #one {
8     position: absolute;
9     top: 10px;
10    left: 10px;
11    width: 200px;
12    height: 200px;
13    background-color: green;
14    border: 1px solid black;
15    padding: 10px 10px;
16    z-index: 1;
17 }
18
19 #two {
20     position: absolute;
21     top: 40px;
22     left: 40px;
23     width: 200px;
24     height: 200px;
25     background-color: red;
26     border: 1px solid blue;
27     padding: 10px 10px;
28 }
29
30 #two span {
31     position: absolute;
```

```
32     bottom: 10px;  
33     right: 10px;  
34 }
```

(the above code snippet online)

Do you see the rules for `#two span`. They position the text to the bottom right corner of the div. Save and reload the page. You should now see the word `two` appearing at the bottom right corner.



Word ‘two’ At the Bottom Right Corner

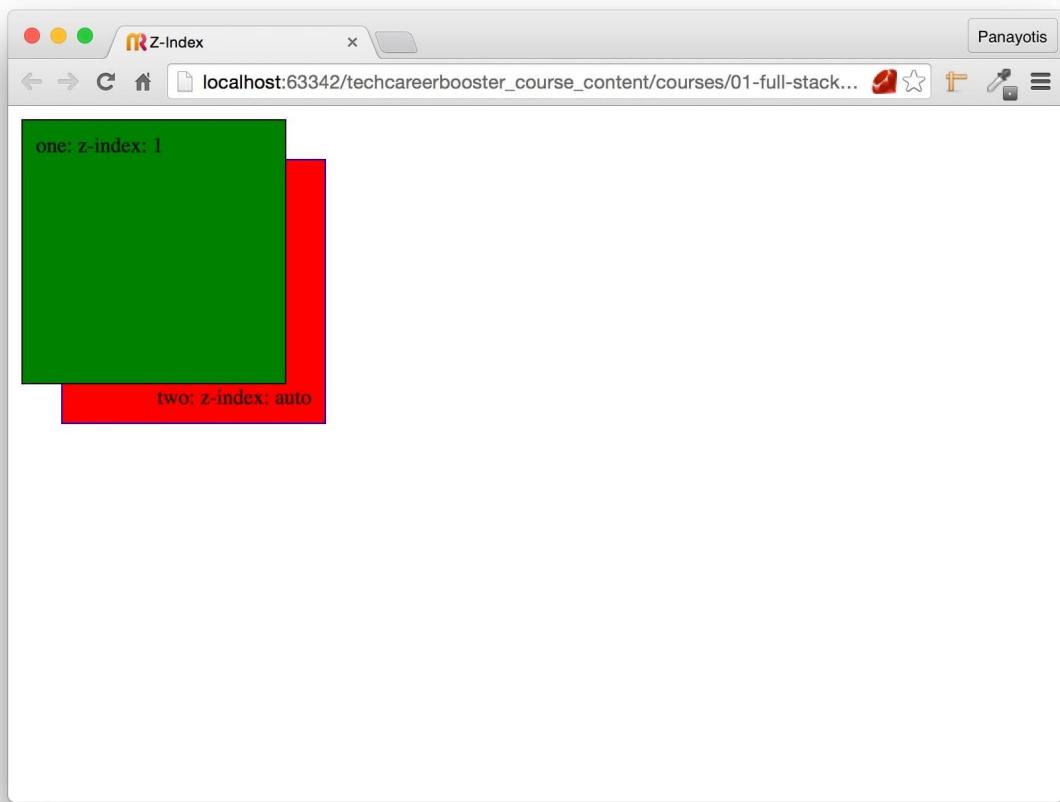
We will actually change the text of the divs to also display their z-index. Let's do that:

```
1  <!DOCTYPE html>  
2  <html>  
3      <head>  
4          <meta charset="utf-8">  
5          <title>Z-Index</title>  
6          <link rel="stylesheet" href="stylesheets/z-index.css" type="text/css">  
7      </head>  
8  
9      <body>
```

```
11   <div id="one">
12     one: z-index: 1
13   </div>
14
15   <div id="two">
16     <span>two: z-index: auto</span>
17   </div>
18
19 </body>
20
21 </html>
```

(the above code snippet online)

which gives this:



Text In Containers Displaying z-index Value

Let's add one more div, two-one as child of the two div. Give that an absolute position too, without z-index.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Z-Index</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="one">
12             one: z-index: 1
13         </div>
14
15         <div id="two">
16             <span>two: z-index: auto</span>
17             <div id="two-one">
18                 <span>three: z-index: auto</span>
19             </div>
20         </div>
21
22     </body>
23
24 </html>
```

(the above code snippet online)

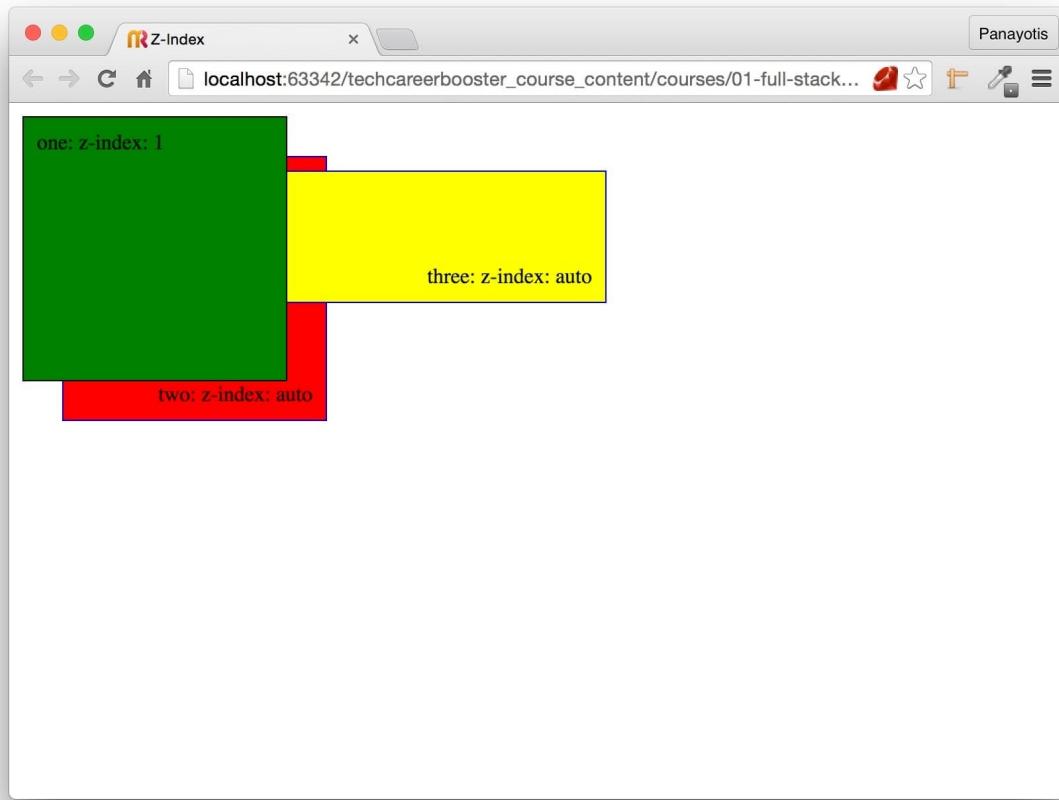
with the next CSS:

```
1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7  #one {
8      position: absolute;
9      top: 10px;
10     left: 10px;
11     width: 200px;
12     height: 200px;
13     background-color: green;
14     border: 1px solid black;
15     padding: 10px 10px;
16     z-index: 1;
17 }
18
```

```
19 #two {  
20     position: absolute;  
21     top: 40px;  
22     left: 40px;  
23     width: 200px;  
24     height: 200px;  
25     background-color: red;  
26     border: 1px solid blue;  
27     padding: 10px 10px;  
28 }  
29  
30 #two span {  
31     position: absolute;  
32     bottom: 10px;  
33     right: 10px;  
34 }  
35  
36 #two-one {  
37     position: absolute;  
38     background-color: yellow;  
39     border: 1px solid blue;  
40     width: 400px;  
41     height: 100px;  
42 }
```

(the above code snippet online)

If you save and reload the page, you will see this:



Child Div With No Z-Index

As you can see above, the two-one div, which is child of two is drawn after the two and behind the one div. That is expected. The two-one does not specify any `z-index` value. So, it will be drawn behind the one which does. It is on the same stack level with two because that one does not specify any `z-index` value too.

How, can we bring two-one on the foreground? Just set a `z-index` positive value, greater than or equal to the `z-index` of the one div.

```
1  * {  
2      box-sizing: border-box;  
3      margin: 0;  
4      padding: 0;  
5  }  
6  
7  #one {  
8      position: absolute;  
9      top: 10px;  
10     left: 10px;  
11     width: 200px;  
12     height: 200px;  
13     background-color: green;
```

```
14     border: 1px solid black;
15     padding: 10px 10px;
16     z-index: 1;
17 }
18
19 #two {
20     position: absolute;
21     top: 40px;
22     left: 40px;
23     width: 200px;
24     height: 200px;
25     background-color: red;
26     border: 1px solid blue;
27     padding: 10px 10px;
28 }
29
30 #two span {
31     position: absolute;
32     bottom: 10px;
33     right: 10px;
34 }
35
36 #two-one {
37     position: absolute;
38     background-color: yellow;
39     border: 1px solid blue;
40     width: 400px;
41     height: 100px;
42     z-index: 1;
43 }
```

(the above code snippet online)

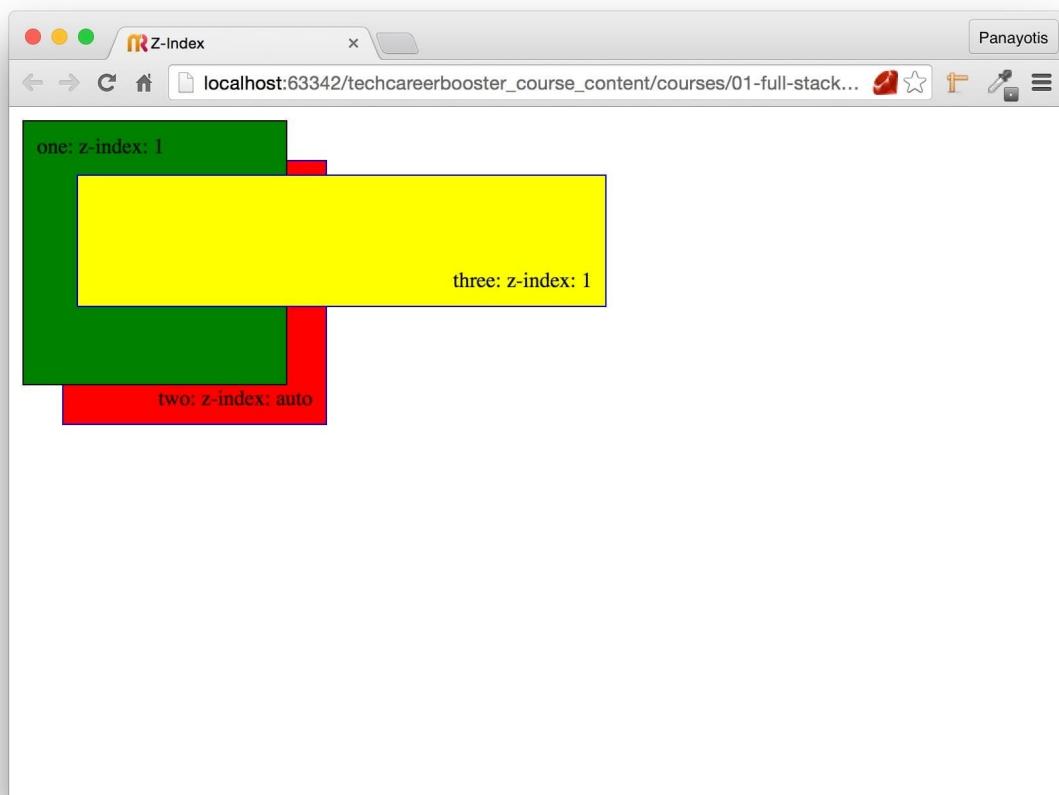
On the previous CSS, we have added the `z-index` property with value `1` for the `two-one` div. Also, change the HTML content so that it has the correct text for the `two-one` div:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Z-Index</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10
11     <div id="one">
```

```
12      one: z-index: 1
13  </div>
14
15  <div id="two">
16    <span>two: z-index: auto</span>
17    <div id="two-one">
18      <span>three: z-index: 1</span>
19    </div>
20  </div>
21
22 </body>
23
24 </html>
```

(the above code snippet online)

If you save both the above files and reload, you will see this:



Child Div Now Has Positive Z-Index That Brings That to the Foreground

That was pretty straightforward. However, with `z-index` there is a small detail that you need to be aware of. Whenever you set the `z-index` of an element to a value different from `auto` then you create a new *stack context* (or `z-index` context). Inside that new stack context, are going to

be stacked both the particular element plus all of its child elements.

Let's set the `z-index` value of the `div` two to value `0`. Here is the new HTML (to depict the new value for it):

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Z-Index</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="one">
12             one: z-index: 1
13         </div>
14
15         <div id="two">
16             <span>two: z-index: 0</span>
17             <div id="two-one">
18                 <span>three: z-index: 1</span>
19             </div>
20         </div>
21
22     </body>
23
24 </html>
```

(the above code snippet online)

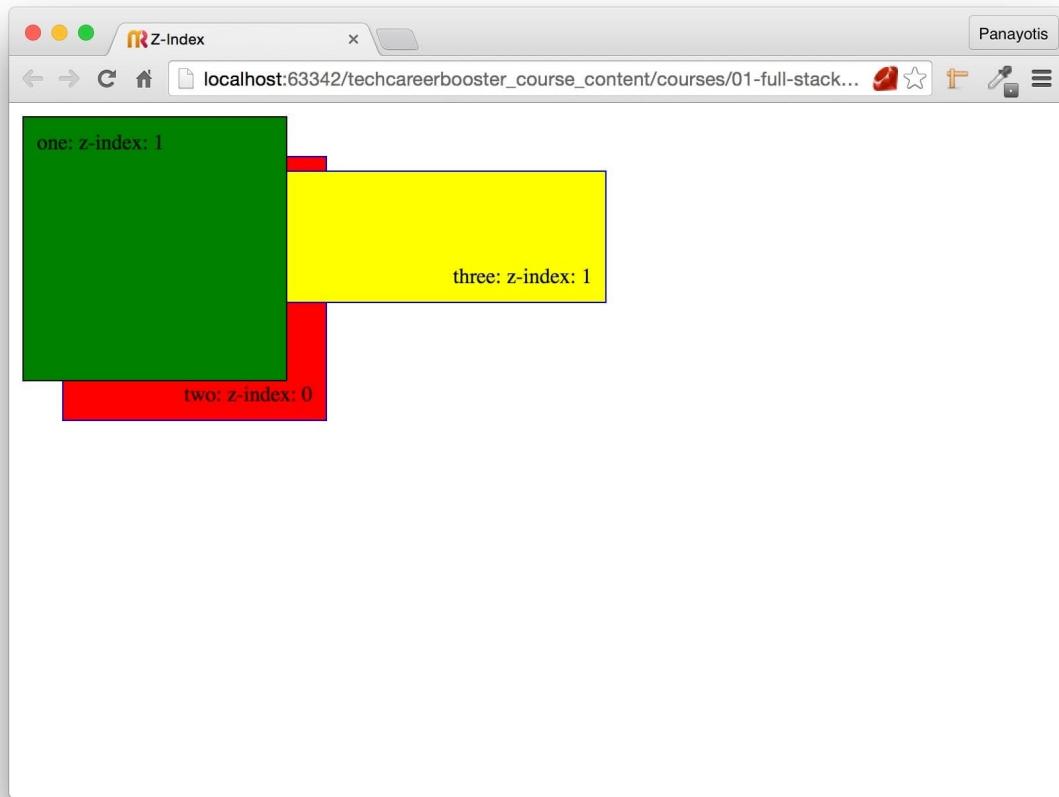
And the CSS file:

```
1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7  #one {
8      position: absolute;
9      top: 10px;
10     left: 10px;
11     width: 200px;
12     height: 200px;
13     background-color: green;
```

```
14     border: 1px solid black;
15     padding: 10px 10px;
16     z-index: 1;
17 }
18
19 #two {
20     position: absolute;
21     top: 40px;
22     left: 40px;
23     width: 200px;
24     height: 200px;
25     background-color: red;
26     border: 1px solid blue;
27     padding: 10px 10px;
28     z-index: 0;
29 }
30
31 #two span {
32     position: absolute;
33     bottom: 10px;
34     right: 10px;
35 }
36
37 #two-one {
38     position: absolute;
39     background-color: yellow;
40     border: 1px solid blue;
41     width: 400px;
42     height: 100px;
43     z-index: 1;
44 }
```

(the above code snippet online)

If you save these files and reload the page, you will see this:



New Stack Context on div 'two'

As you can see, the div two-one is now drawn behind div one. This is because now div two and div two-one form a new and isolated stack context. You can understand that even more, if you change the `z-index` of the two-one div, from 1, to 2. Let's do that:

The HTML:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Z-Index</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="one">
12             one: z-index: 1
13         </div>
14
15         <div id="two">
```

```
16      <span>two: z-index: 0</span>
17      <div id="two-one">
18          <span>three: z-index: 2</span>
19      </div>
20  </div>
21
22  </body>
23
24 </html>
```

(the above code snippet online)

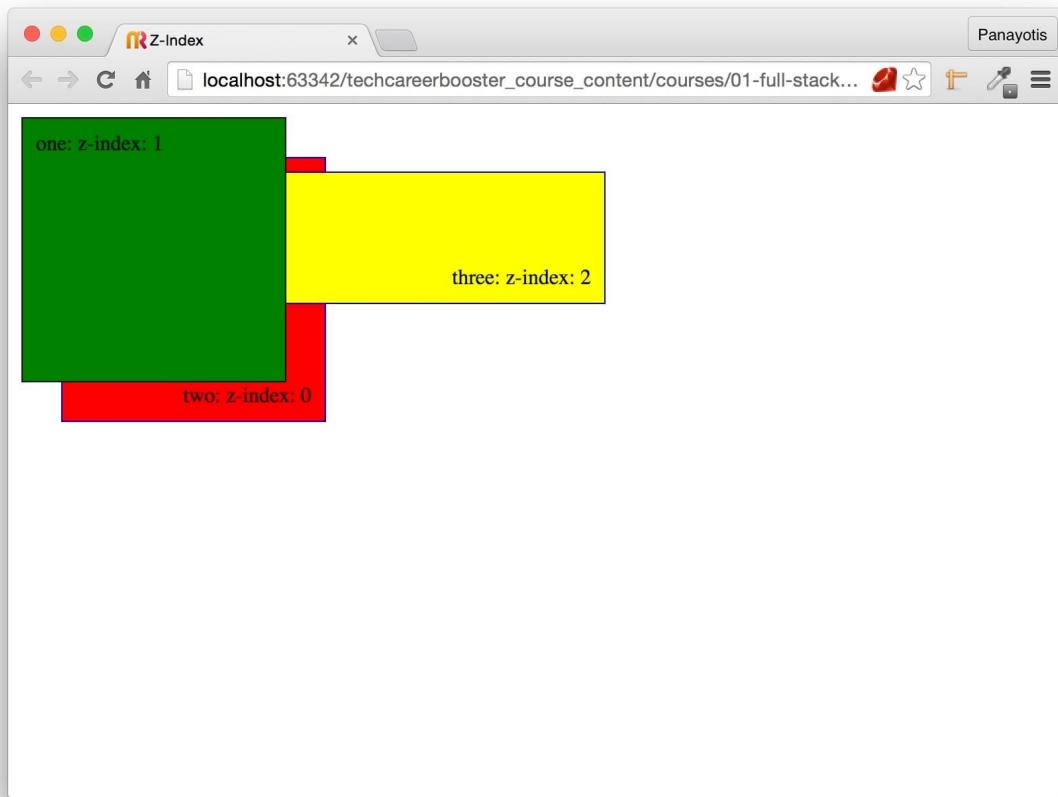
and the CSS:

```
1  * {
2      box-sizing: border-box;
3      margin: 0;
4      padding: 0;
5  }
6
7 #one {
8     position: absolute;
9     top: 10px;
10    left: 10px;
11    width: 200px;
12    height: 200px;
13    background-color: green;
14    border: 1px solid black;
15    padding: 10px 10px;
16    z-index: 1;
17 }
18
19 #two {
20     position: absolute;
21     top: 40px;
22     left: 40px;
23     width: 200px;
24     height: 200px;
25     background-color: red;
26     border: 1px solid blue;
27     padding: 10px 10px;
28     z-index: 0;
29 }
30
31 #two span {
32     position: absolute;
33     bottom: 10px;
```

```
34     right: 10px;  
35 }  
36  
37 #two-one {  
38     position: absolute;  
39     background-color: yellow;  
40     border: 1px solid blue;  
41     width: 400px;  
42     height: 100px;  
43     z-index: 2;  
44 }
```

(the above code snippet online)

If you save those files and reload the page, you will see this:



Higher z-index But Still Behind

Do you see that? Even if the two-one div has higher `z-index` value if compared to div one, it is still behind it, because they do not belong to the same stack context. No matter how much you increase the `z-index` value of two-one, it will still be behind one div. Its stack level, position, is only affected by the elements and stack level positions of the elements of its parent, div two, i.e. of its own stack context. The only way to bring two-one div in front of the one div again, is either

by removing the `z-index` value from the two `div`, or setting the two `div` `z-index` value to a value greater than or equal to 1. In that latter case, both two and two-one will be brought in front of the one `div`, as a group.

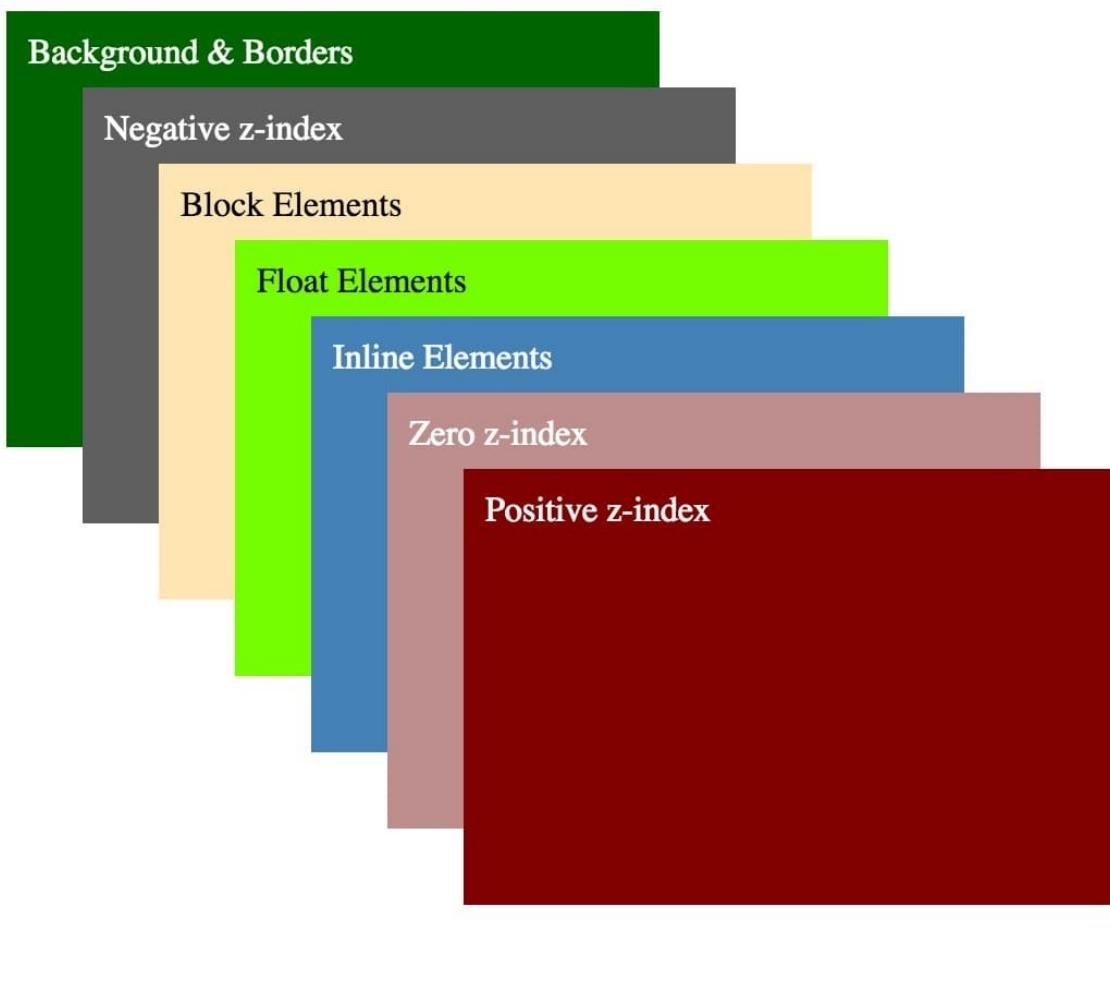
Finally, one might ask:

What is the stacking order when, inside the same stack context, except from the positioned elements with specific `z-index` values, we also have elements that are not positioned elements and hence, they do not have a `z-index` property applicable?

In other words, what happens if we have, inside the same stack context:

- positioned elements with `z-index` set, and
- block elements and
- inline elements and
- float elements

The stacking order in that case is the following:



The above diagram explains the order in which elements are drawn. For example:

- The float elements are drawn in front of the block elements and they are both drawn in front of the negative z-index elements.
- The background and borders are always drawn behind all other elements.

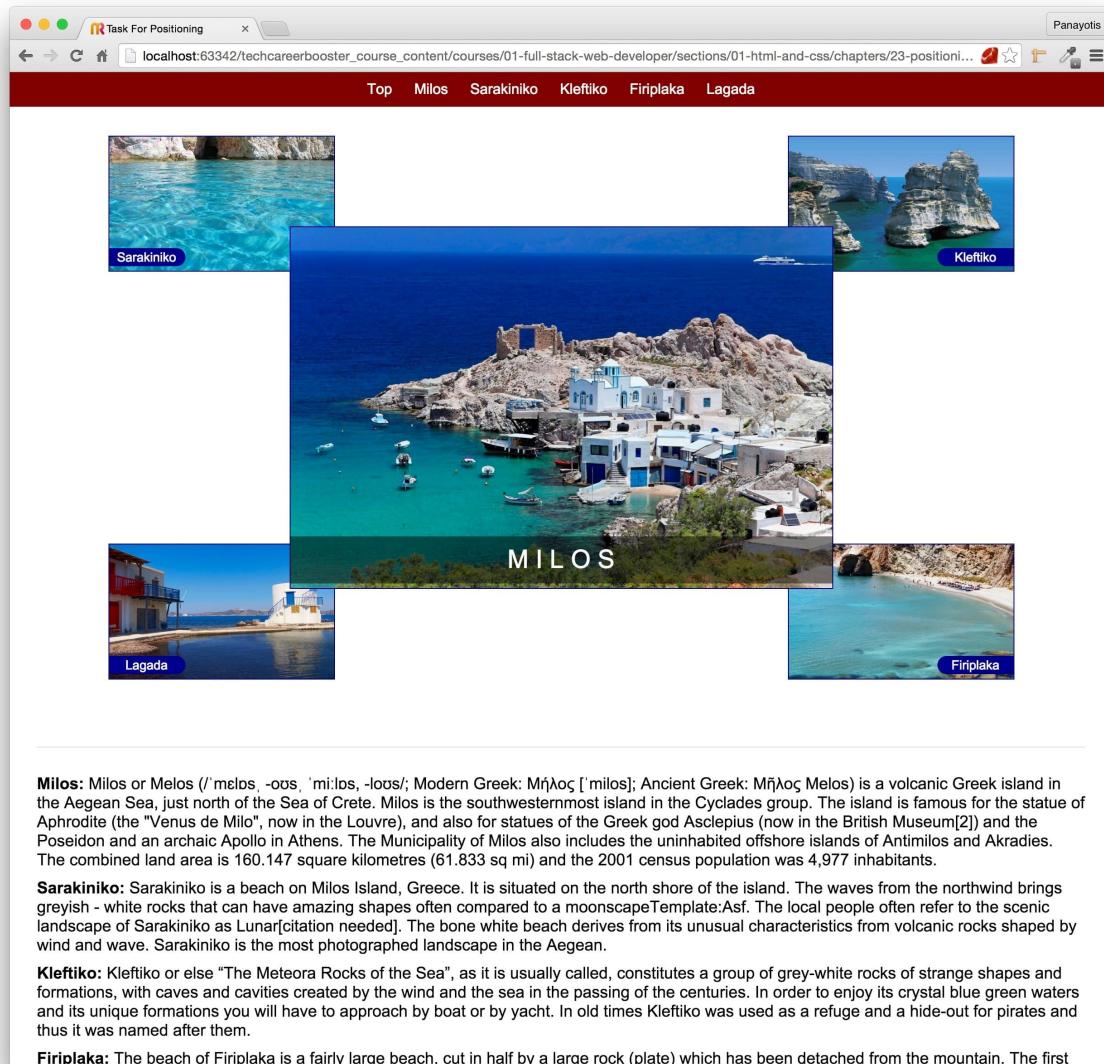
Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. You need to build an HTML page like this:



Task for Positioning

2. Here is a video that demonstrates how the page works:

How your task site should be functioning

3. There is a top navigation bar, which needs to be fixed and stay visible even if we scroll the content of the page.
4. The navigation bar has links inside the main content paragraphs.
5. The navigation links change color when we mouse over them.
6. The images that you will need are already inside the images folder of the chapter.
 1. milos-firopotagmos.jpg. This is for the central image.
 2. sarakiniko-water.jpg. This is for the top left image.
 3. kleftiko.jpg. This is for the top right image.
 4. firiplaka.jpg. This is for the bottom right image.
 5. lagada.jpg. This is for the bottom left image.
7. We give you here the text parts, so that you do not have to type them in:

1. “Milos or Melos (/’mēlōs, -ōs, ‘mi:lōs, -lōs/; Modern Greek: Μήλος ['milos]; Ancient Greek: Μῆλος Melos) is a volcanic Greek island in the Aegean Sea, just north of the Sea of Crete. Milos is the southwesternmost island in the Cyclades group. The island is famous for the statue of Aphrodite (the “Venus de Milo”, now in the Louvre), and also for statues of the Greek god Asclepius (now in the British Museum[2]) and the Poseidon and an archaic Apollo in Athens. The Municipality of Milos also includes the uninhabited offshore islands of Antimilos and Akradies. The combined land area is 160.147 square kilometres (61.833 sq mi) and the 2001 census population was 4,977 inhabitants.”
 2. “Sarakiniko is a beach on Milos Island, Greece. It is situated on the north shore of the island. The waves from the northwind brings greyish - white rocks that can have amazing shapes often compared to a moonscapeTemplate:Asf. The local people often refer to the scenic landscape of Sarakiniko as Lunar[citation needed]. The bone white beach derives from its unusual characteristics from volcanic rocks shaped by wind and wave. Sarakiniko is the most photographed landscape in the Aegean.”
 3. “Kleftiko or else “The Meteora Rocks of the Sea”, as it is usually called, constitutes a group of grey-white rocks of strange shapes and formations, with caves and cavities created by the wind and the sea in the passing of the centuries. In order to enjoy its crystal blue green waters and its unique formations you will have to approach by boat or by yacht. In old times Kleftiko was used as a refuge and a hide-out for pirates and thus it was named after them.”
 4. “Welcome to the captivating island of Milos and its greatest hotel, Lagada Beach Hotel. Right in front of Lagada Beach Hotel, you can experience raremoments of relaxation in a quiet and calm region pretty close tothe city center! The blue color of the Aegean and the green hues of hotel’s gardens combinedwith Miloscolors give you the opportunity to live a unique experience of accommodation.”
 5. “The beach of Firiplaka is a fairly large beach, cut in half by a large rock (plate) which has been detached from the mountain. The first part of the beach is the one where you will find the beach bar and sunbeds, probably the second song on your own or with a camper company that has reached the other side of the beach. The beach has shallow but clear waters combined with white sand. It is best to avoid days with southerly winds. Before or after that you can also visit the beach of Tsigrado which is right next to that of Firiplaka.”
8. We would also like to give you some hints in order to succeed with this task:
- All images are background images to corresponding divs
 - The images at the corners should be placed relative to the central image.
 - The link “top”, in the menu, requires the existence of a div like this: `html <div id="top"></div>` exactly after the opening body element. This will also need to have the following CSS rule: `#top { position: absolute; top: 0; }` This is one trick that will allow the Top link to work.
 - Add the following rule to your CSS file: `body { margin-bottom: 1000px; }` This will give enough height to body so that the last links on the top nav bar can work and bring the corresponding bookmark on the top.
 - You will need to make sure that the image containers have reasonable sizes:
 - Central image should be 600px X 400px.

- All image containers should be 250px x 150px.
 - The tags are all of equal size. You might want to use the property `min-width` for that.
 - Make sure that the top nav bar is not hidden by the content when we scroll down.
9. After you finish your task, deploy your site to Netlify.

Chapter 24 - CSS Units

Summary

CSS offers a series of units that we can use to define sizes. For example, we can define `font-size` property using pixels: `px` or root elements: `rem`. This chapter will give a head start to understand the difference between absolute vs relative sizes.

Learning Goals

1. Learn about pixels
2. Learn about the browser default font size
3. Learn about absolute measurement units
4. Learn about relative measurement units
5. Learn how you can use `rem` units to set the font size of your site in more predictable values.

CSS Units

In our CSS files, we have used a couple of measurement units to declare the width and/or height of an element, or the font size of text.

- `px` We have used pixels (`px`) in various occasions. For example to declare the size of a `div` or an `image`. Or in order to set the font size.
- `%` We have used a percentage value to declare the size of an element in relation to the size of its container.

Let's learn some more things about measurement units in CSS.

Everything is pixels

No matter what unit we use to specify the dimensions of an element, finally, the browser is going to calculate its size in pixels. We will later on see how different units are converted to pixels. The point here is that 1 pixel does not necessary occupy the same space (e.g. calculated in inches or centimeters) on the display. It depends on the quality of the display. So, a display with high number of dpi (dots per inch) or ppi (pixels per inch) will display an area of 10×10 pixels in less, more compact, space from another display with lower dpi. In other words, if we have a display that has 10dpi resolution, it will require 1 inch² area to display 10×10 pixels. Whereas if the display has 20dpi resolution, it will require 0.25 inch² (0.5×0.5). Or in other words, a 10px font-size letter will be displayed smaller on higher resolution displays.

Absolute Units

px is an absolute unit of measurement. Some other absolute units that you can use are the following:

- cm: centimeters
- mm: millimeters
- in: inches
- pt: points. Each point is 1/72 of 1 inch.
- pc: picas. Each pica is 12 points.

Which measurement unit you would use, depends on the device medium you are trying to design a page for. If it is for a display, we usually use pixels. If it is for a printer, we might use points or inches or other absolute measurement unit. But again, there is no rule here. As long as you know what you are doing, you can use any of the above absolute units.

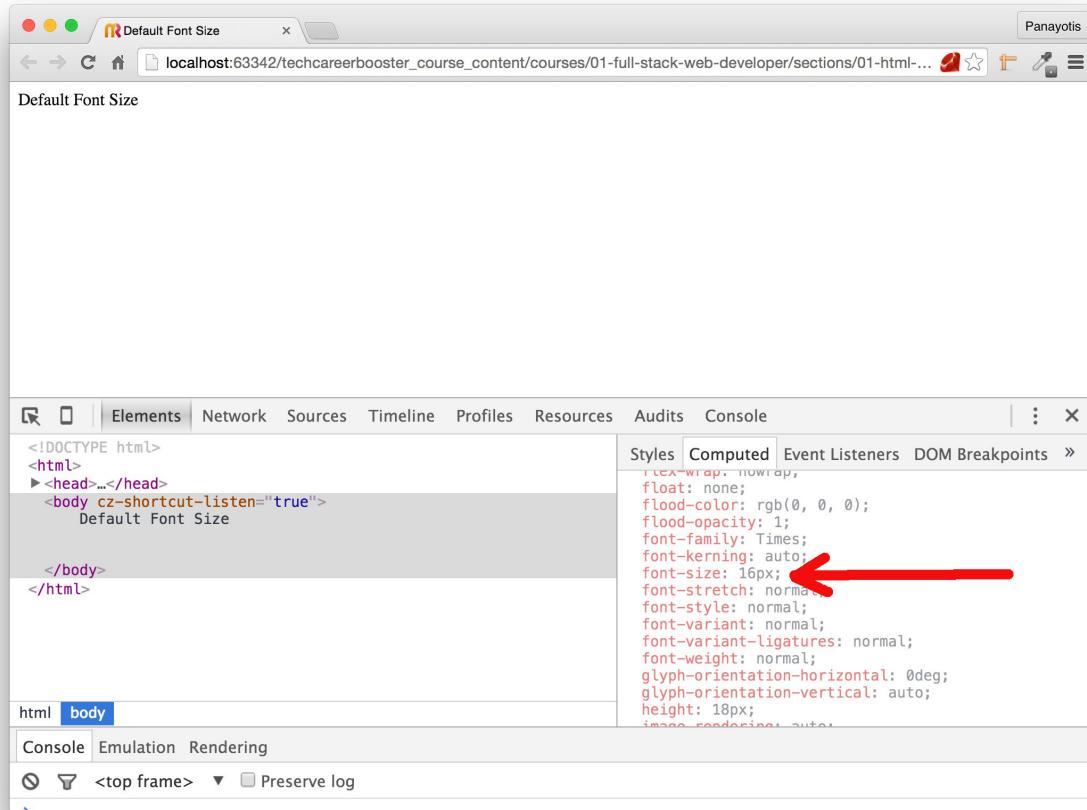
Default Font Size

When you have an empty, bare minimum, web page, the browsers are using a default font size for the content of the body. Let's see that:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Default Font Size</title>
6   </head>
7
8   <body>
9     Default Font Size
10  </body>
11
12 </html>
```

(the above code snippet online)

Save the above page and load it on your browser. Then open the developer tools, select the body on the elements and go on the right panel on Computed values. Scroll down to find the font-size property. You will see that it has the value 16px.



Default Font Size is 16px

All popular browsers are using the default font-size of 16px if you do not specify any font size. This, however, can be changed via the settings of your browser. For Chrome, if one goes to Settings and then to Advanced Settings, he can change the default font size from Medium to other value.

Watch this video here to see how we change the browser default font size value from Medium (which is the default and corresponds to 16px) to other values and then back to Medium.

[How to change browser default font size?](#)

Default Font Size and Absolute Units

There is a problem with default font size and absolute units, that you might expect. The absolute units, ignore the default font size. Hence, if the user changes the default browser font size, but the page uses an absolute value to specify the font size of the text, default value is ignored.

Let's see the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Default Font Size</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         Default Font Size (16px)
11         <p class="lead">
12             32px font-size: Lorem ipsum dolor sit amet, consectetur adipiscing elit, se\
13             d do eiusmod tempor incididunt ut labore et dolore magna aliqua.
14             Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut \
15             aliquip ex ea commodo consequat. Duis aute irure dolor
16                 in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pari\
17             atur. Excepteur sint occaecat cupidatat non proident,
18                 sunt in culpa qui officia deserunt mollit anim id est laborum.
19         </p>
20     </body>
21
22 </html>
```

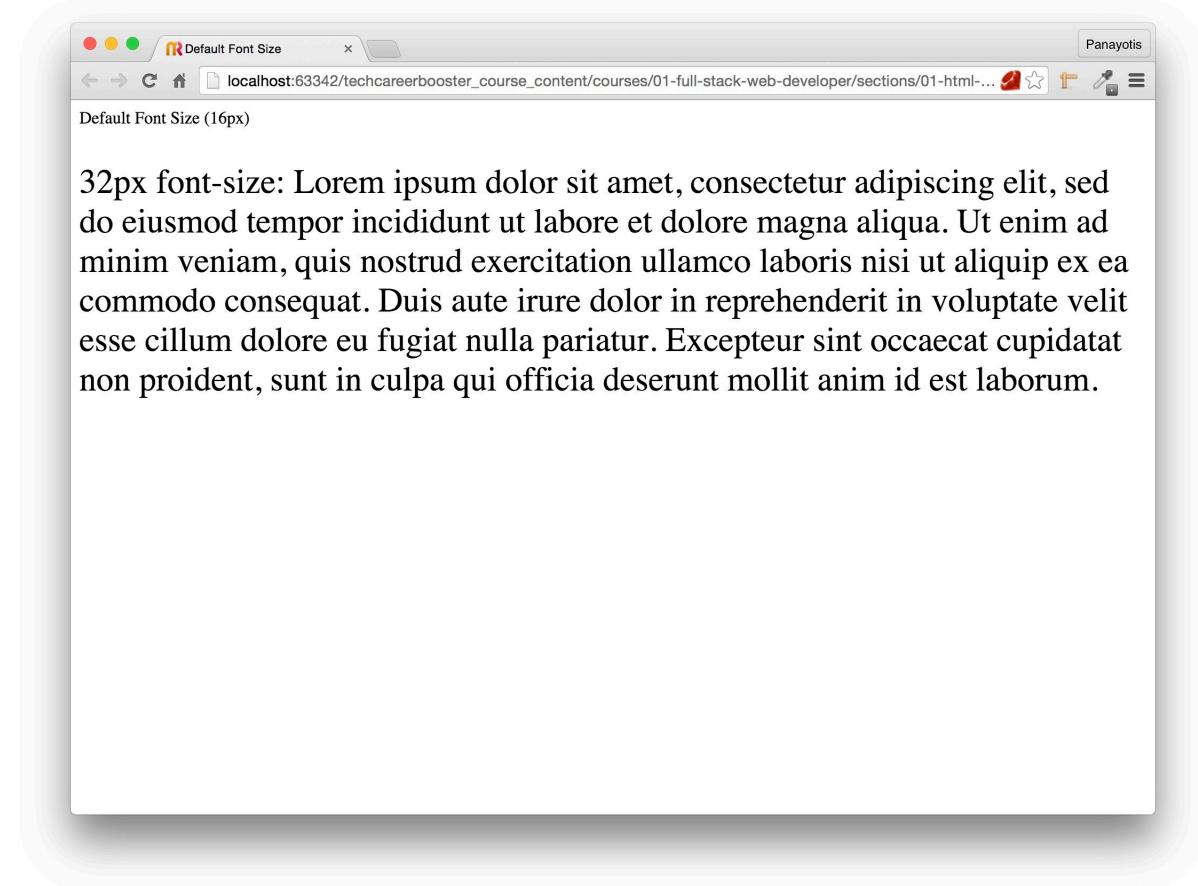
(the above code snippet online)

with the following CSS:

```
1  * {
2      box-sizing: border-box;
3  }
4
5  .lead {
6      font-size: 32px;
7  }
```

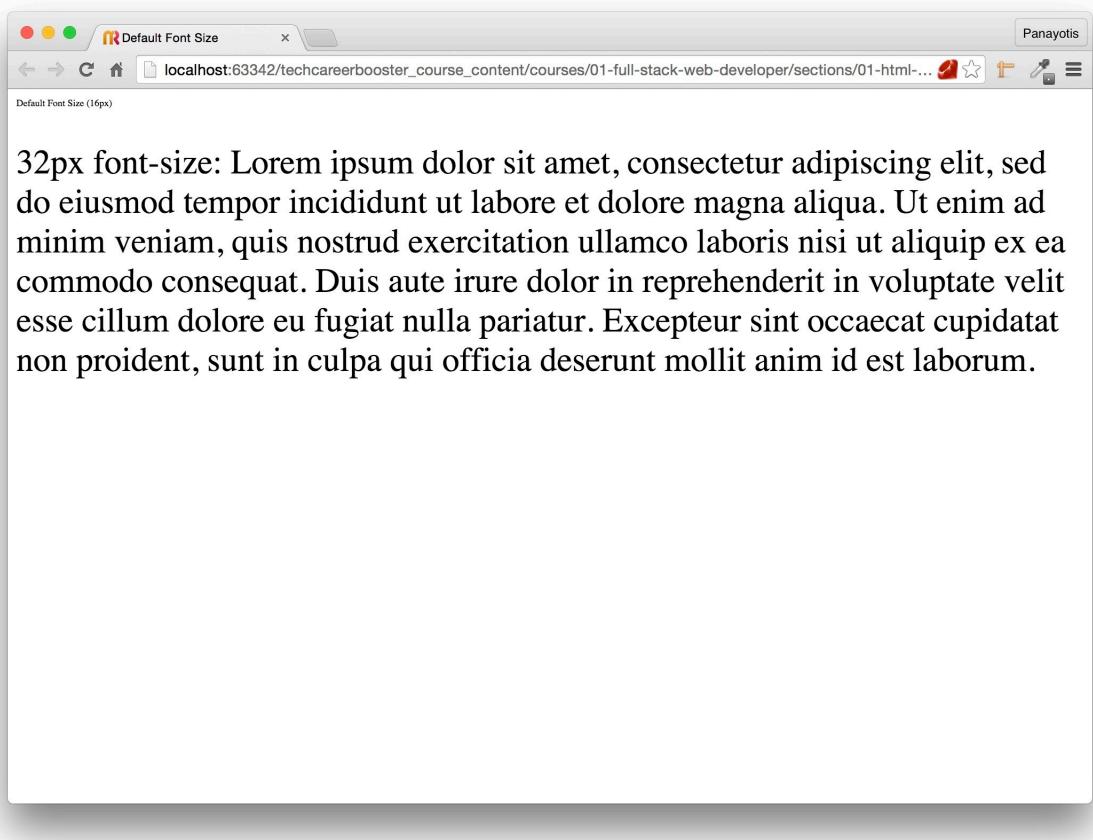
(the above code snippet online)

If you save and reload your page, you will see this:



Page with Absolute Font Size

The above picture is when viewing at the page with the default browser font size to 16px and with the paragraph absolute font size to 32px. Now if you go and change the default browser font size to Very Small, which corresponds to 9px, you will see that the paragraph font size is not changed, because its font size is hard coded to 32px absolute value. You will see this:



Default Browser Font Size Reduced to 9px But Paragraph Font Size Remains 32px

Note: If you changed your default browser settings to Very Small (9px) set it back to Medium

Relative Units

Sometimes, we do not want to specify the font-size with an absolute unit value, but we want to do that in relation to the font-size of another element. Let's see the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Default Font Size</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10    Default Font Size (16px)
11    <p class="lead">
12      120% font-size (19.2px): Lorem ipsum dolor sit amet, consectetur adipiscing\
```

```
13    elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
14        Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut \  
15    aliquip ex ea commodo consequat. Duis aute irure dolor  
16        in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pari\  
17    atur. Excepteur sint occaecat cupidatat non proident,  
18        sunt in culpa qui officia deserunt mollit anim id est laborum.  
19    </p>  
20    </body>  
21  
22 </html>
```

(the above code snippet online)

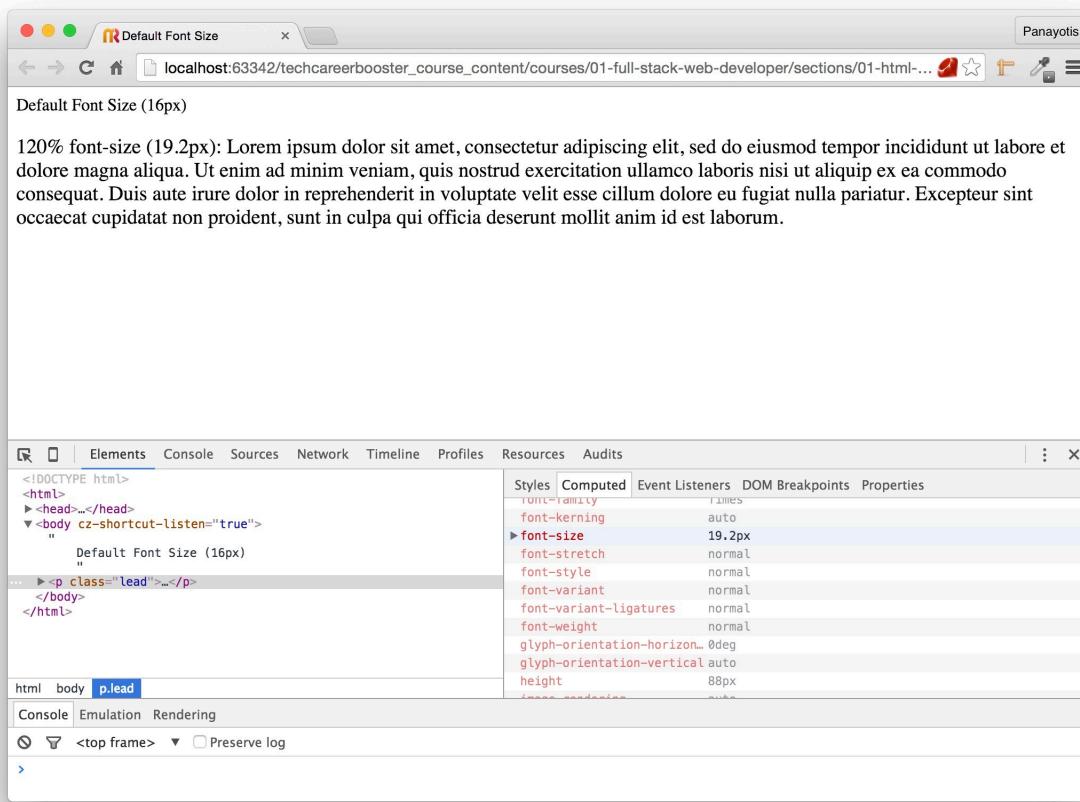
with the following CSS:

```
1 * {  
2     box-sizing: border-box;  
3 }  
4  
5 .lead {  
6     font-size: 120%;  
7 }
```

(the above code snippet online)

As you can see above, the `font-size` for the paragraph is set to `120%`. This means 120% of the `font-size` value of its parent. Its parent is `body`, which has the browser default `font-size` value of `16px`. Hence, the `font-size` of the paragraph is $16 * 120\% = 19.2px$.

If you save the above files and reload the page in your browser you will see this:



Paragraph With Percentage Font Size

% is a *relative unit* of sizing. We have already used that to create multi-column layouts. Here we have used it to define the font size of an element. The good thing with relative units is that, when the size of the element, that size depends on, changes, then that size changes proportionally too. So, this means, that, in our example, if the default browser font size changes, then the paragraph font size will change too. Let's do that. Go to your browser settings and change the default browser font size to Very Small. Then see how both the body and the paragraph font sizes change proportionally.

How relative units are affected by the size of the property they depend on

In the above video, you will see that when we change the browser default font size to 9px, then paragraph font size changes to 10.8px, which is the 120% of 9. When we set the browser default font size to Very Large, i.e. 24px, then paragraph font size changes to 28.8px, which is 120% of 24. This proportional change of the paragraph font size is realized because we have used a relative unit (%) to set its font-size. This didn't happen when we used an absolute unit earlier (px).

There are some other popular relative units used in the web development today. Some of them are the following:

- em: relative to the font size of the current element.
- rem: relative to the font size of the root element, i.e. `html` element. So, 1rem usually is 16px, since the `html` font-size, by default, in most browsers is 16px. The nice thing with rem is

that they are not affected by nesting. Whereas `em` and `%`, they are. Nesting and relative font sizes might cause some difficult problems to tackle.

- `vw`: relative to 1% of the width of the viewport. The viewport is the browser window size.
- `vh`: relative to 1% of the height of the viewport.

`rem` units have become very popular to define font sizes. But it is a little bit difficult to convert them to pixels or vice versa:

Examples of Pixels and their REM

| px | rem |
|------|----------|
| 12px | 0.75rem |
| 14px | 0.875rem |
| 16px | 1rem |
| 18px | 1.125rem |
| 20px | 1.25rem |

This means that if you want to have a paragraph with font size `20px`, you need to remember, calculate, that this is equal to `1.25rem` (assuming that the default font size is `16px`).

There is a trick that you can use to avoid that complex calculation. Applying the following trick, converting pixels to rem would only require you to divide the pixels size by 10. If you use the following CSS rules:

```

1 html {
2     font-size: 62.5%; /* which is equal to 10px (16 * 0.625 = 10) */
3 }
4 body {
5     font-size: 1.6rem; /* which makes it 16px */
6 }
```

(the above code snippet online)

Then, the body will start having `16px` font-size, but still relevant to browser default font size. And you can convert pixels to rem very easily:

Examples of Pixels and their REM

| px | rem |
|------|--------|
| 12px | 1.2rem |
| 14px | 1.4rem |
| 16px | 1.6rem |
| 18px | 1.8rem |
| 20px | 2.0rem |

Look at the following html page:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Default Font Size</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         Body text 1.6rem (16px)
11         <p class="lead">
12             1.92rem font-size (19.2px): Lorem ipsum dolor sit amet, consectetur adipisc\
13             ing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
14             Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut \
15             aliquip ex ea commodo consequat. Duis aute irure dolor
16                 in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pari\
17             atur. Excepteur sint occaecat cupidatat non proident,
18                 sunt in culpa qui officia deserunt mollit anim id est laborum.
19         </p>
20     </body>
21
22 </html>

```

(the above code snippet online)

With the following CSS file:

```

1  * {
2      box-sizing: border-box;
3  }
4
5  html {
6      font-size: 62.5%;
7  }
8
9  body {
10     font-size: 1.6rem;
11 }
12
13 .lead {
14     font-size: 1.92rem;
15 }

```

(the above code snippet online)

If you save the above and you reload the page, you will see that you have the same result as earlier. But now, that you are using this technique:

1. You can calculate the `rem` easily, because you only have to divide pixels by 10.
2. Since you are using `rem` you will not face any problem with nesting. All sizes are relevant to the root element font-size.

The good thing with relative units is that if a browser uses a different default font-size, then all sizes are automatically adjusted accordingly. This capability will become invaluable when we would like our web site to be properly viewed on mobile devices too.

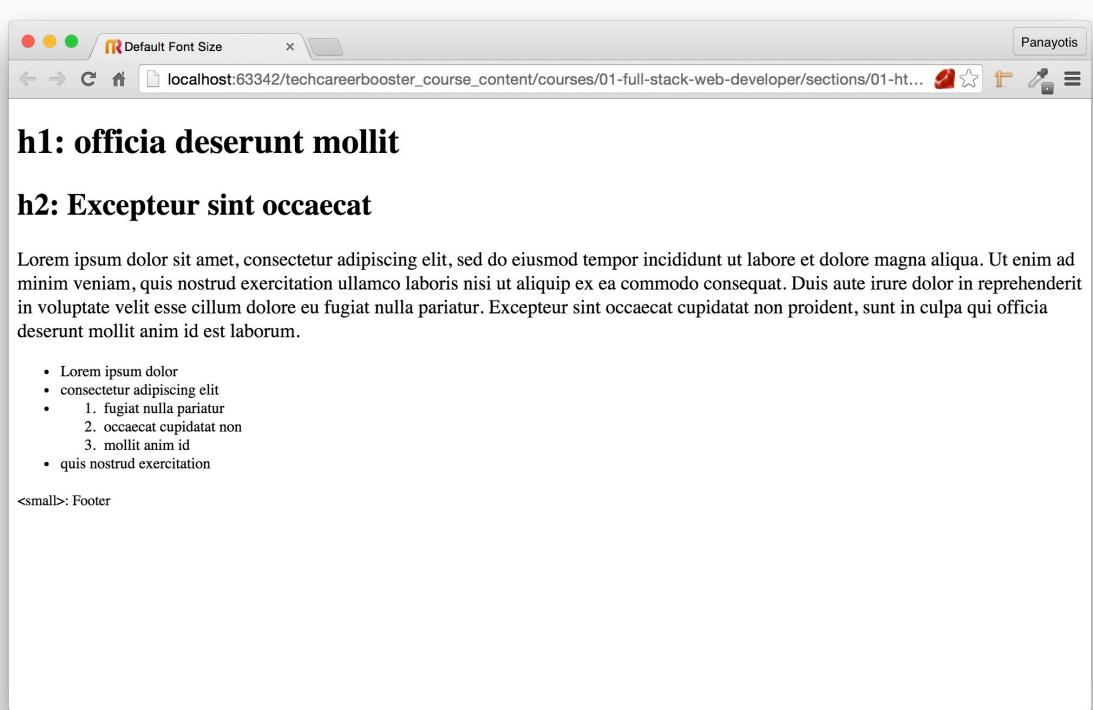
Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Implement a web page like the following:



2. Make sure about the following: 1. Use only relative units for the font sizes 2. Use the trick that is described in the chapter content in order to make the relative units easy to calculate. 3. The

h1 font size needs to be 32px. 4. The h2 font size needs to be 28px. 5. The p font size needs to be 18px. 6. The li font size needs to be 14px. 3. Having implemented the above, upload to your Netlify account. 4. Change the CSS rule for li to be using em instead of rem. What do you notice? Is everything ok? If not, why?

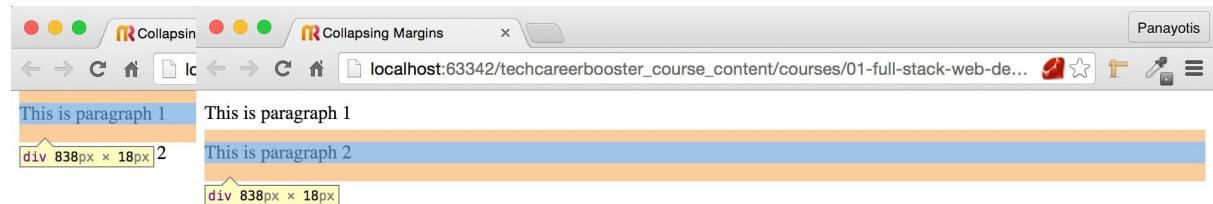
Chapter 25 - Collapsing Margins

Summary

When you are novice to CSS you will get surprised by vertical margins missing from your final HTML page, even if you have clearly set them to non-zero values. This is the phenomenon of the vertical collapsing margins. This chapter explains why and when does it happen. Also it proposes how you should deal with that so that you will have designs that are easier to maintain.

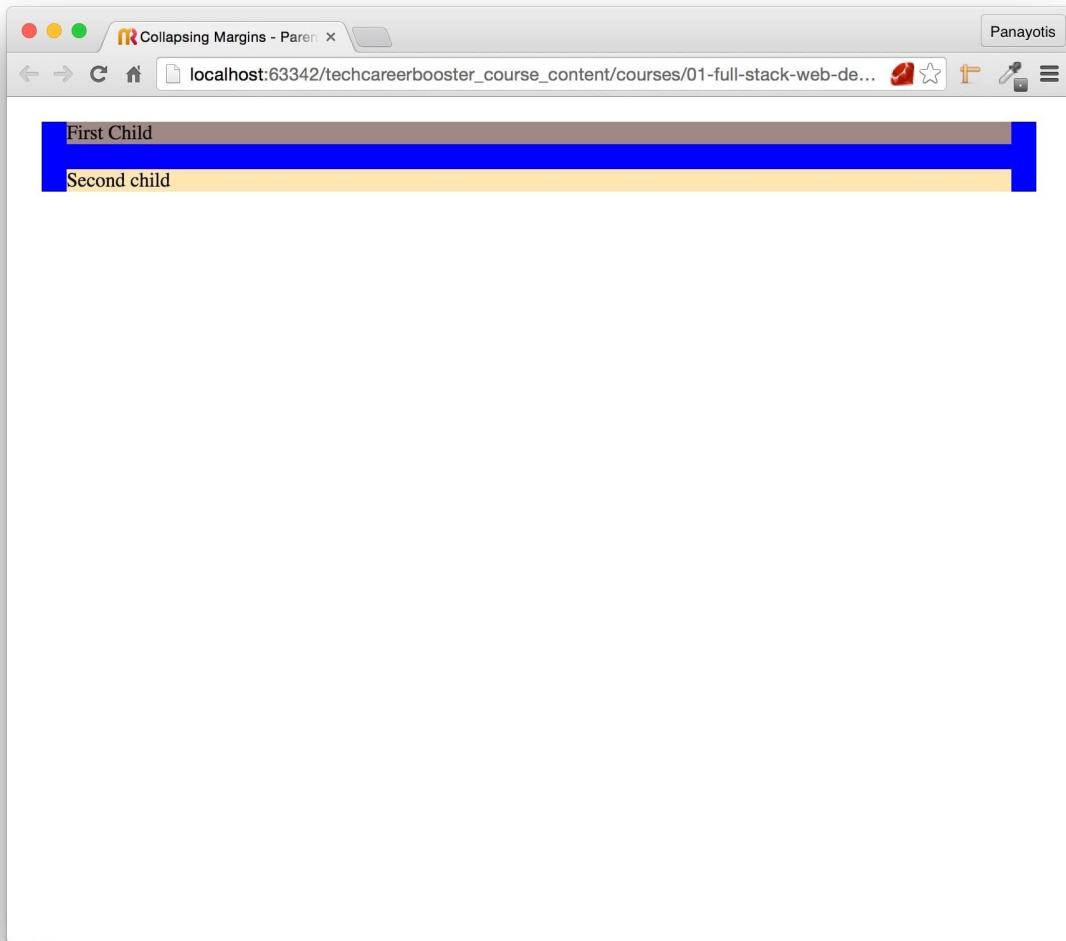
Hence, on this chapter you will encounter the vertical collapsing margins phenomenon.

Like in this case, where we have two divs, one on top of the other:



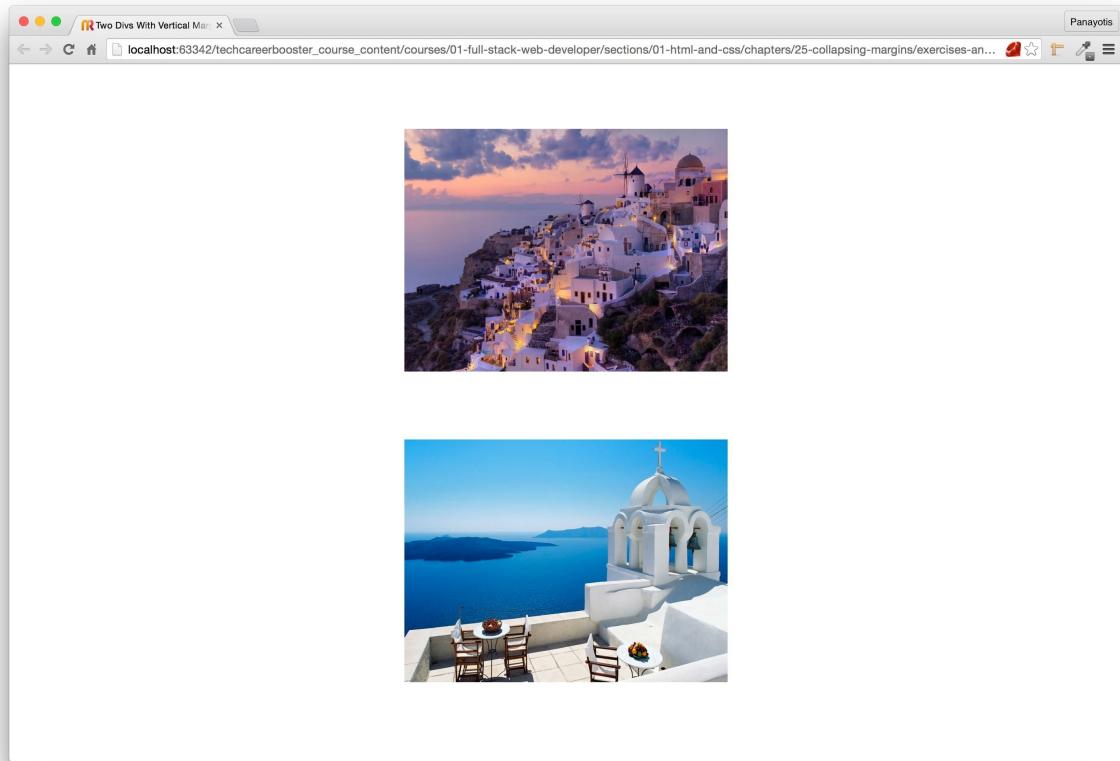
Two divs One On Top of the Other

And like in this case, in which we have nested children divs that they lose their margins:



Nested Children divs Losing their Margins

Finally, you will be requested to implement a page like this:



Vertical Collapsing Margins

Learning Goals

1. Understand when the collapsing margins phenomenon appears.
2. Learn about ways to bring back the vertical margins that are missing from your elements.
3. Learn about the best practice in web design with regards to collapsing margins.

Collapsing Margins

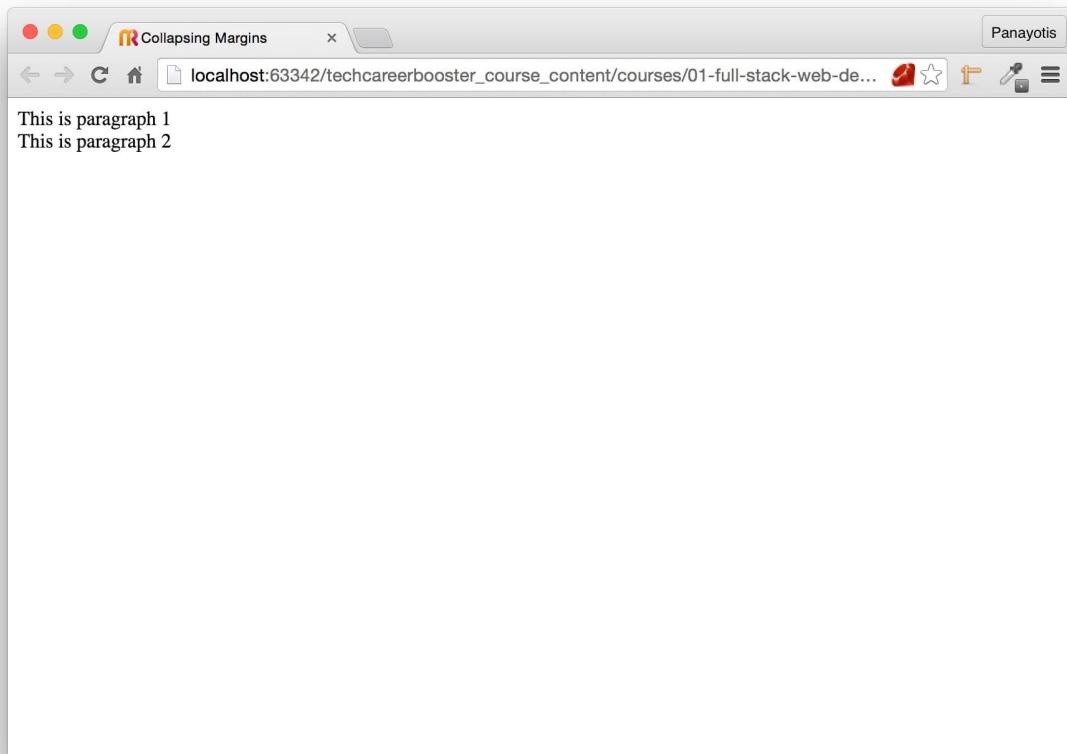
You will often stand in front of a situation in which you are supposed to be having margins, but you do not see them.

Let's see the following simple HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Collapsing Margins</title>
6      </head>
7
8      <body>
9          <div>This is paragraph 1</div>
10         <div>This is paragraph 2</div>
11     </body>
12 </html>
```

(the above code snippet online)

Save the above file as an HTML page and load it on your browser. You will see something like this:



Page with Two Simple Divs One Below the Other

These are two simple `divs`, one on top of the other. If we inspect them with developer tools, we will see that they do not have any top or bottom margins:

The screenshot shows a web browser window titled "Collapsing Margins". The address bar indicates the page is "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-de...". The main content area displays two adjacent

elements with the text "This is paragraph 1" and "This is paragraph 2". The browser's developer tools are open, with the "Elements" tab selected. In the "Styles" panel of the developer tools, the "margin-top" property is listed as "0px", which is crossed out with a large red 'X'. The "margin-bottom" property is also listed as "0px". Other properties like "list-style-type" and "list-style-position" are present but not crossed out.

Divs do not have any margin

Let's create a CSS file that is referenced by our HTML page above. The CSS file should have the following rules:

```
1 div {  
2     margin-top: 10px;  
3     margin-bottom: 15px;  
4 }
```

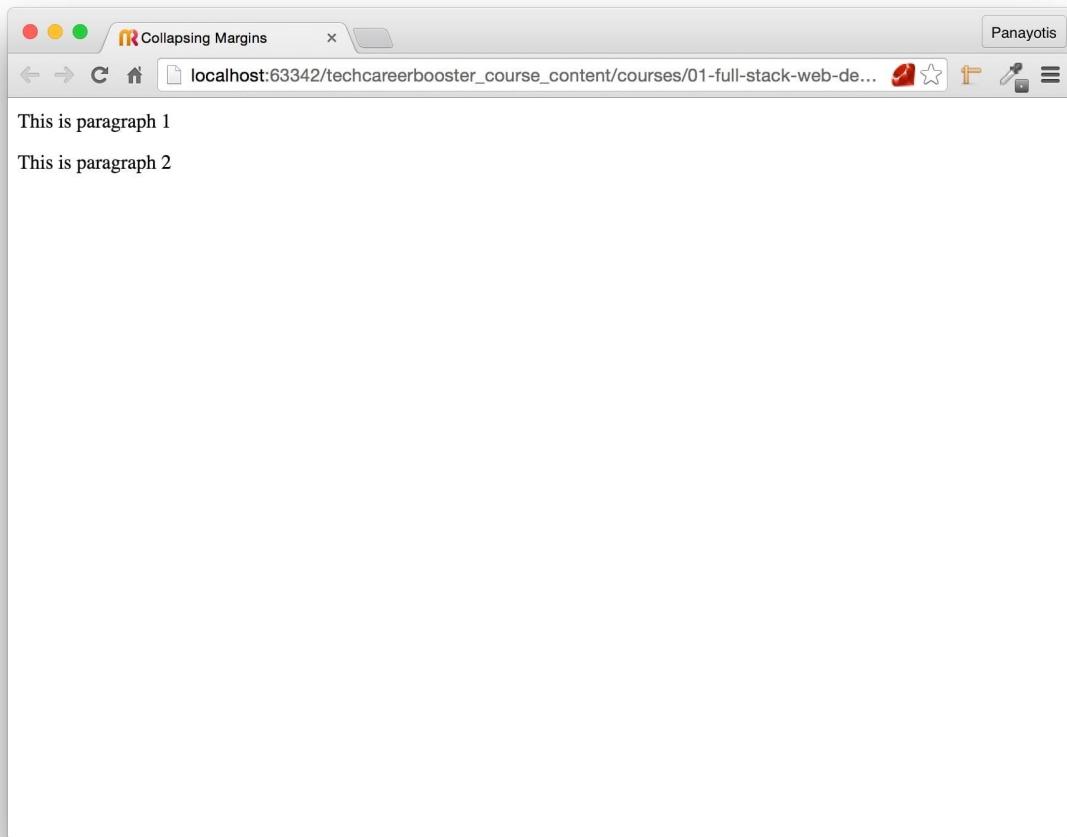
(the above code snippet online)

Make sure that your HTML page references the above CSS content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Collapsing Margins</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div>This is paragraph 1</div>
11         <div>This is paragraph 2</div>
12     </body>
13 </html>
```

(the above code snippet online)

and reload your page. You will see this:



Page with Two Simple Divs One Below the Other - With Vertical Margins

Note that we gave both divs the same top and bottom margins. And we can see that. But, if you carefully inspect with the developer tools, you will see that the two margins that were supposed

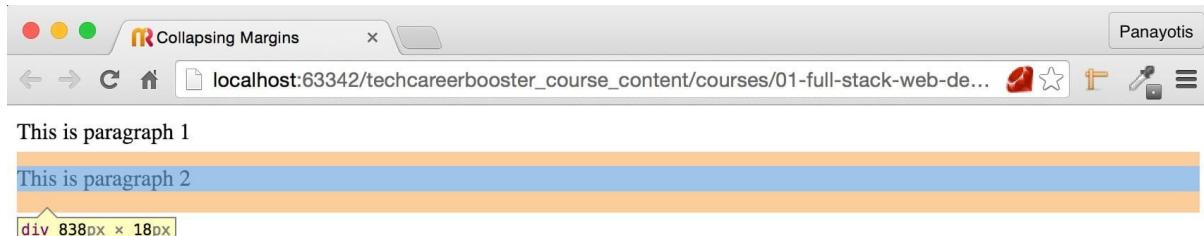
to be touching each other, i.e. the bottom margin of the top div and the top margin of the bottom div, do not both exist.

This is the picture that we see when we inspect the top div with developer tools:

Inspecting the margins of the top div

Top div clearly has 10px top and 15px bottom margins. But, if you look carefully, you will see that the bottom margin edge is exactly above the content area of the bottom div. Why is that? This was not supposed to be like that. It seems that the top margin of the bottom div has disappeared.

What about the picture of the bottom div margins in developer tools? There it is:

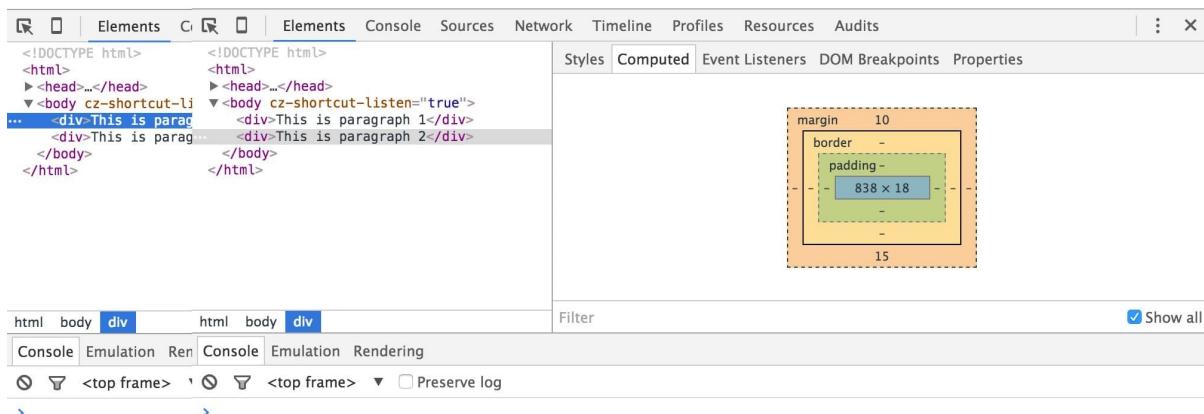
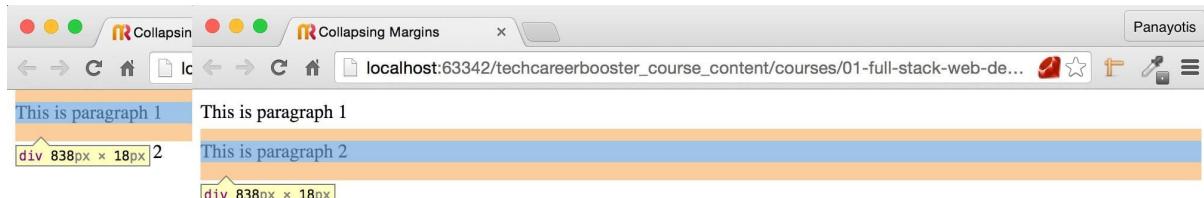


Inspecting the margins of the bottom div

Bottom div too, has 10px top and 15px bottom margins.

But, if you look carefully, the top margin of the bottom div does not start at the expected vertical position, i.e. after the end of the bottom edge of the bottom margin of the top div. It starts **inside** the top div bottom margin area. In fact it is contained, it is completely drawn **inside** the bottom margin area of the top div. That's why the bottom margin of the top div ends exactly before the main content area of the bottom div starts. As if the bottom div didn't have any margin.

You can verify that too, if you put the two pictures of the developer tools one next to the other:



Inspecting Div Margins One Next to the Other

In summary, the distance between the content of the top div to the content of the bottom div, is not the sum of the height of the bottom margin of the top div plus the height of the top margin of the bottom div, a.k.a. 25px. It is only 15px, which is the maximum of the two margins.

This is an example of the collapsing margins phenomenon. And it can cause a lot of headaches when trying to design our page.

What is collapsing margins?

This phenomenon occurs when two vertical margins are in contact. If one margin is greater than the other, then the larger margin overrides the smaller one, leaving us with only one margin. If the two margins are of equal size, only one of them stays in.

In fact:

- When we have two positive margins, e.g. 15px and 10px, the total margin is the bigger one, i.e. 15px on our example.
- When we have two equal margins, e.g. 15px each, the total margin is equal to the one of the two, i.e. 15px on our example.
- When we have one positive margin and one negative margin, e.g. 15px and -10px, then the final margin is the sum of the positive and negative margin. On our example $15px + (-10px)$ turns out to be 5px.

- When we have two negative margins, e.g. -15px and -10px, then the final margin is the bigger negative margin, i.e. -15px on our example.

Parents and Children

Things are getting a little bit more complicated when you have nested elements with margins.

Take for example the following HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Collapsing Margins - Parent Child</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div id="parent">
11             <div id="first-child">
12                 First Child
13             </div>
14             <div id="second-child">
15                 Second child
16             </div>
17         </div>
18     </body>
19 </html>
```

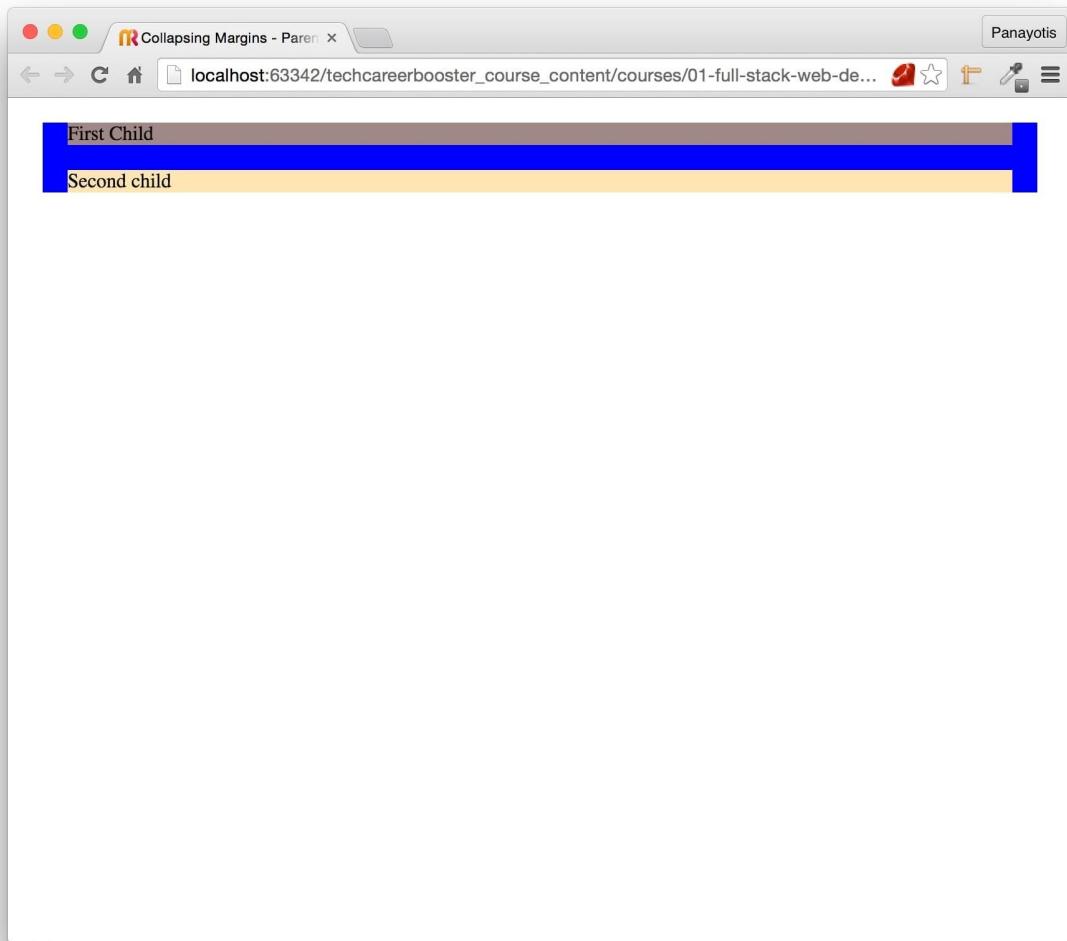
(the above code snippet online)

with the following CSS rules:

```
1  div {
2      background-color: Blue;
3      margin: 20px;
4  }
5
6  #first-child {
7      background-color: #A08888;
8  }
9
10 #second-child {
11     background-color: #FFE4B5;
12 }
```

(the above code snippet online)

As you can see, we have given a margin around all divs. The parent div has 2 nested children divs. Save those files and load the page on your browser. The result will be the following:



Parent and Children with Collapsing Margins

As you can see, the parent has the expected margins, but:

- The first child top margin has been merged with the parent div top margin.
- The first child bottom margin has been merged with the second child top margin.
- The second child bottom margin has been merged with the parent div bottom margin.

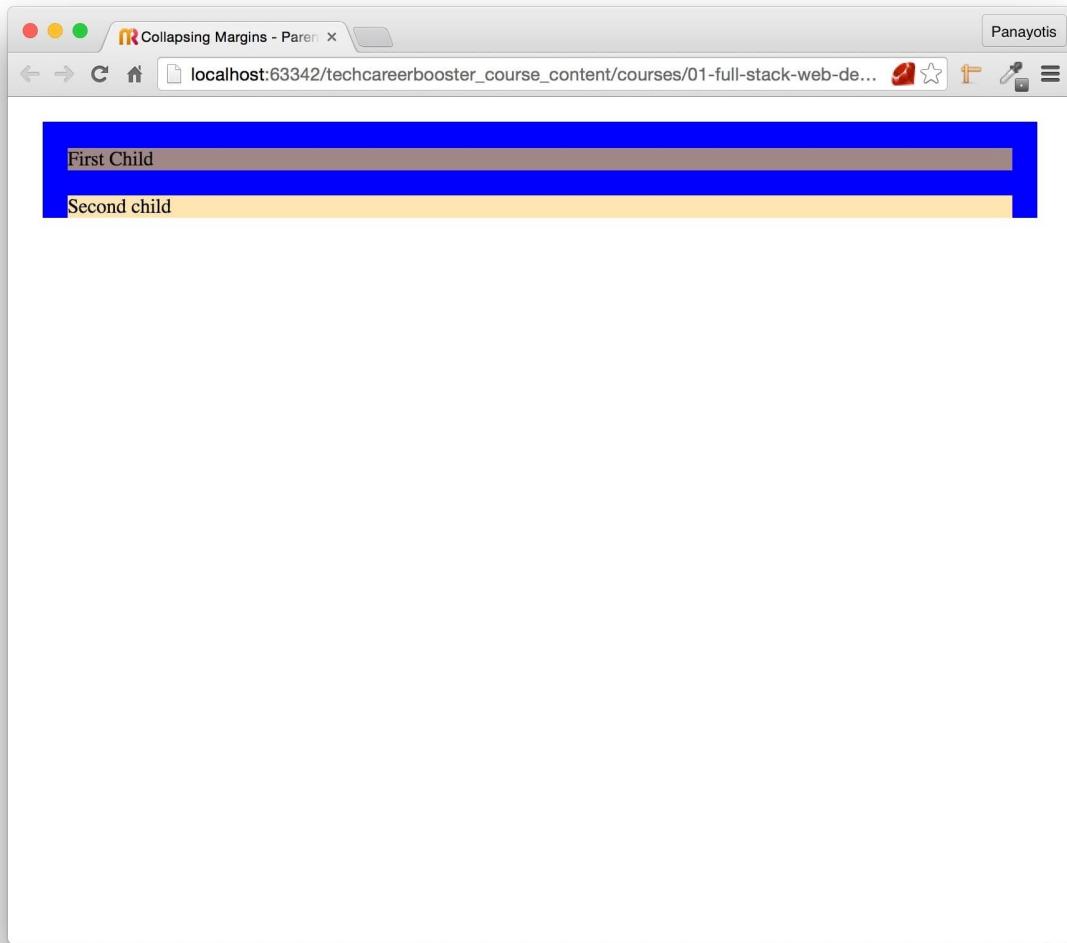
How can we preserve both parent and children margins?

We need to add something *solid* in the parent content area (border to center area). For example, we can add 1px top padding. Let's do that. Change the CSS file content to be as follows:

```
1  div {
2      background-color: Blue;
3      margin: 20px;
4  }
5
6  #parent {
7      padding-top: 1px;
8  }
9
10 #first-child {
11     background-color: #A08888;
12 }
13
14 #second-child {
15     background-color: #FFE4B5;
16 }
```

(the above code snippet online)

Save and reload your page. You will see this:



First Child Top Margin is preserved

As you can see, the first child top margin is preserved. It is not merged into the top margin of the parent div. You can do the same for the bottom margin of the parent div. You can add a padding bottom on the parent div and this will make the second child bottom margin to be preserved.

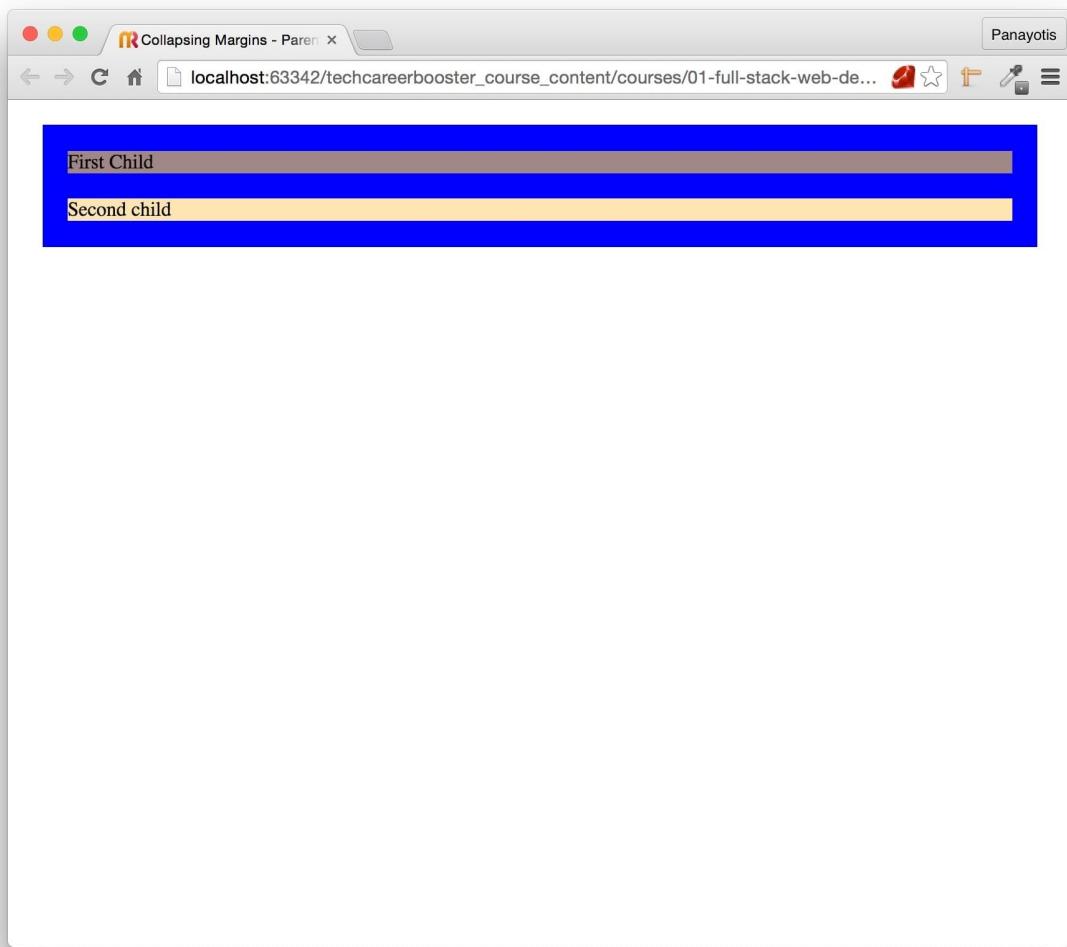
Let's do that:

```
1 div {  
2     background-color: Blue;  
3     margin: 20px;  
4 }  
5  
6 #parent {  
7     padding-top: 1px;  
8     padding-bottom: 1px;  
9 }  
10  
11 #first-child {
```

```
12     background-color: #A08888;  
13 }  
14  
15 #second-child {  
16     background-color: #FFE4B5;  
17 }
```

(the above code snippet online)

If you reload the page you will see this:



Second child bottom margin is now preserved

As you can see, the second child bottom margin is now preserved.

How can we eliminate the collapsing vertical margins phenomenon?

Is there a way we can bring back the vertical margins that are collapsed?

If you really want to avoid your vertical margins being collapsed, you can create an empty div in between the two divs and apply `overflow: auto;` style. Let's do that on the first page that we created:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Collapsing Margins</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div>This is paragraph 1</div>
11         <div class="preserve-vertical-margins"></div>
12         <div>This is paragraph 2</div>
13     </body>
14 </html>
```

(the above code snippet online)

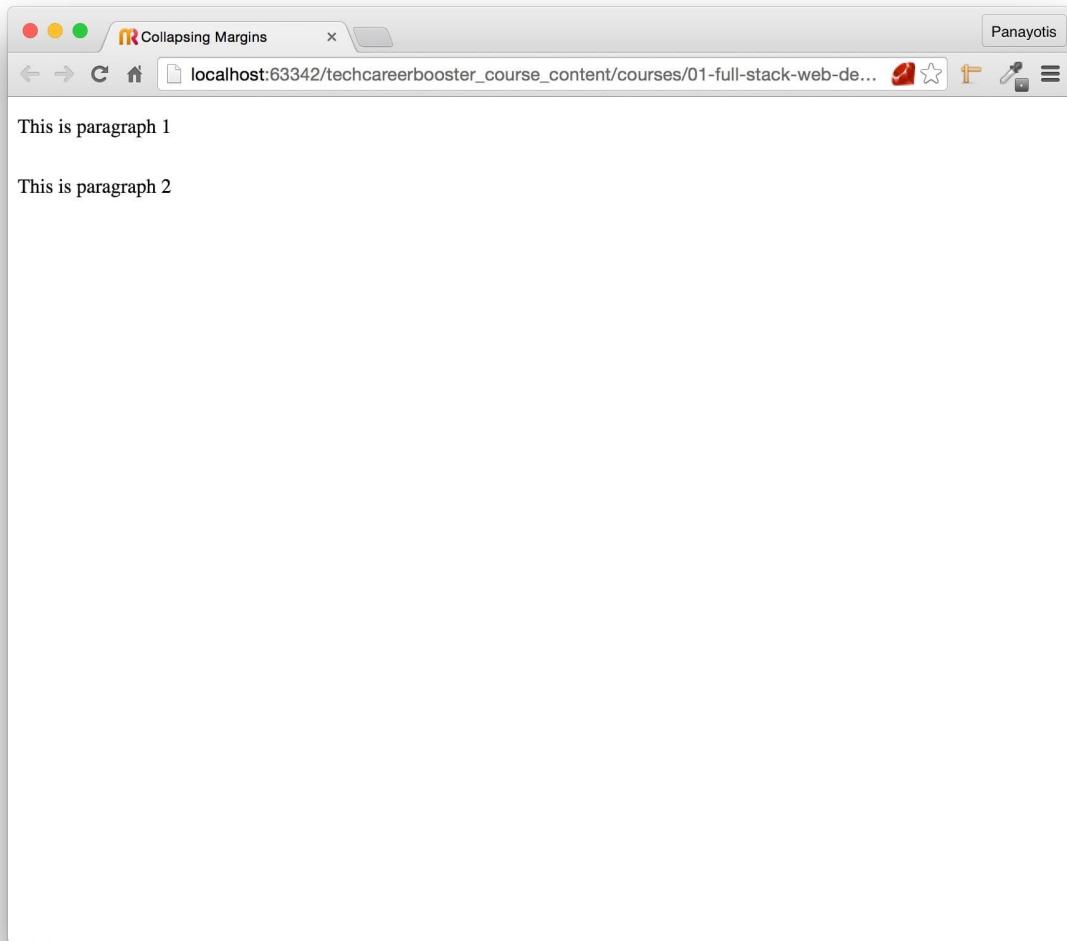
As you can see above, we have added an empty div in between the two original divs. Let's define the class `preserve-vertical-margins` inside the CSS. The CSS should be:

```
1  div {
2      margin-top: 15px;
3      margin-bottom: 15px;
4  }
5
6 .preserve-vertical-margins {
7     overflow: auto;
8 }
```

(the above code snippet online)

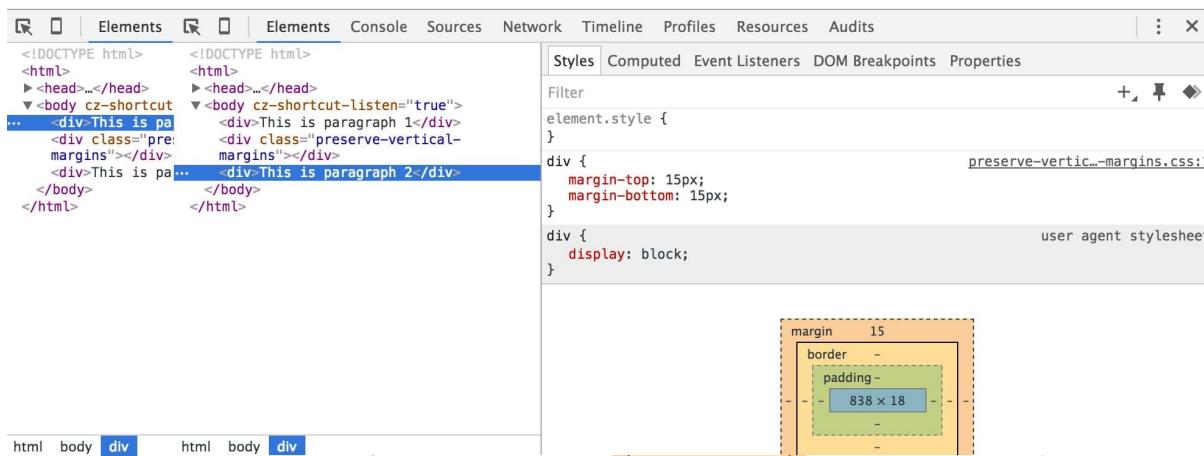
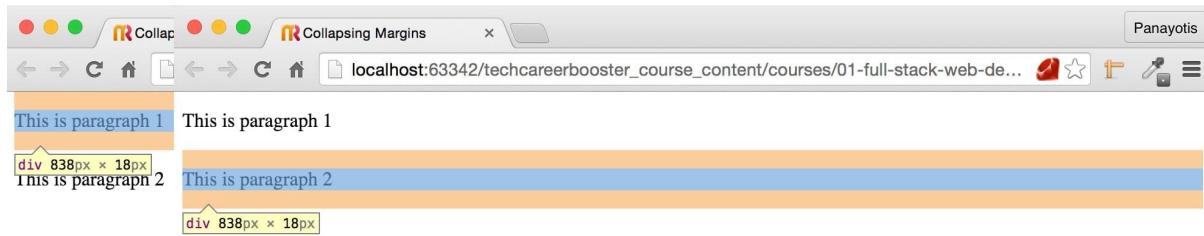
Note: we have made the top and bottom margins equal, just to be easier to inspect with developer tools how the two divs are now positioned at the expected positions on the page.

Save and reload the page. You will see this:



Divs with vertical margins not collapsing

If you inspect the margins of these two divs you will see that they both preserve the margins that they touch each other. The bottom margin of the top div and the top margin of the bottom div. Here is a picture that shows that the margins are preserved.



Divs with vertical margins not collapsing - Inspecting with Developer Tools

As you can see above, the vertical margins are not collapsing and are preserved. This is done with the help of this little hack, with the overflow set to auto.

Why do the vertical margins collapse?

The vertical margins collapsing comes from graphic design. There, you have margins to title and subtitles, but when a subtitle comes just after the title you should not double the margins. This is why they developed the concept of collapsed margins and this is why it happens just on vertical margins.

Closing Suggestion

Generally, vertical margin collapsing can cause great headaches. So, our suggestion here is to stop using both top and bottom margins on your elements. In fact, stick to one direction for all your block level elements. Use only bottom margin. Then your layout will be much more easy to maintain and change.

If you want to read more about why using vertical margin on one direction only is a very good practice, you [can read this article here](#).

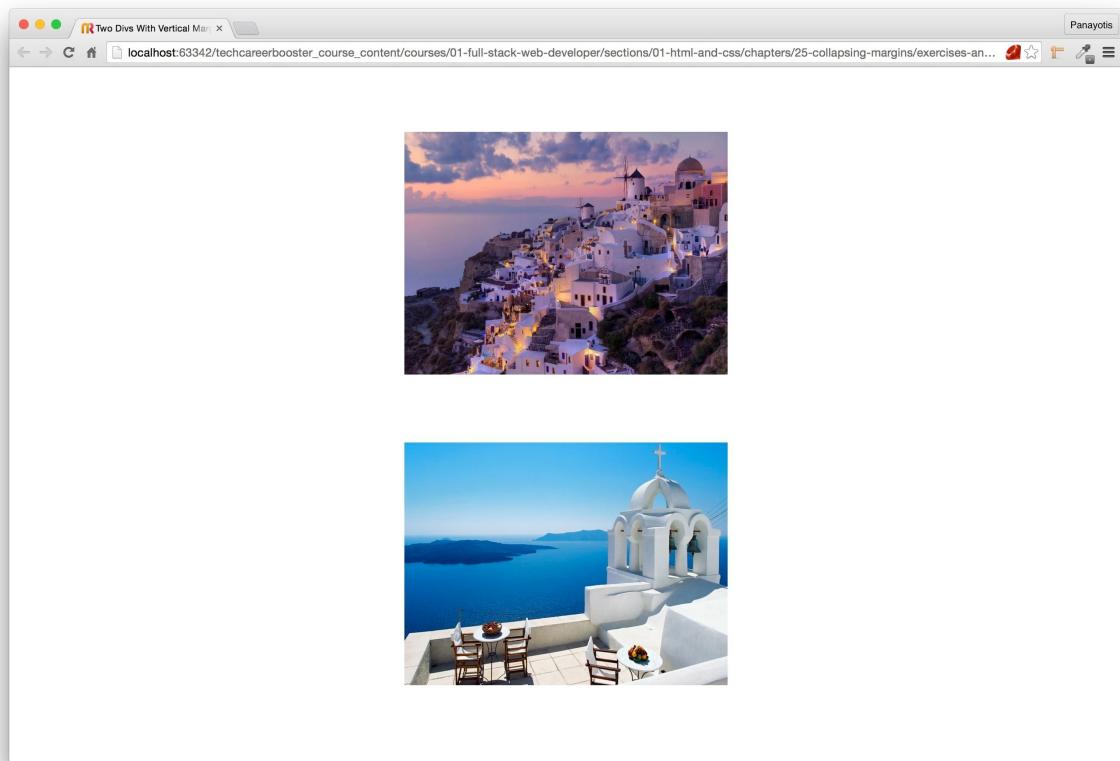
Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Implement the following page:



Task For Vertical Collapsing Margins

2. Watch out about the following:
 1. The pictures have some fixed space in between them.
 2. There is some blank space above the top picture.
 3. There is some blank space below the bottom picture.
 4. The pictures are displayed in the center of the horizontal direction.
 5. Both pictures have same fixed width and height.
3. You need to implement the above page in 2 versions
 1. Version 1: Using a hack to preserve vertical margins
 2. Version 2: Without using vertical margins on both directions

Chapter 26 - Fonts and Typeface

Summary

In this chapter, we are going to learn how we can choose different shapes for the characters of our HTML text. Hence, we are going to learn about font families.

There are some default commonly used font families like these here:

Programming is all about Creativity (Times New Roman)

Programming is all about Creativity (Arial)

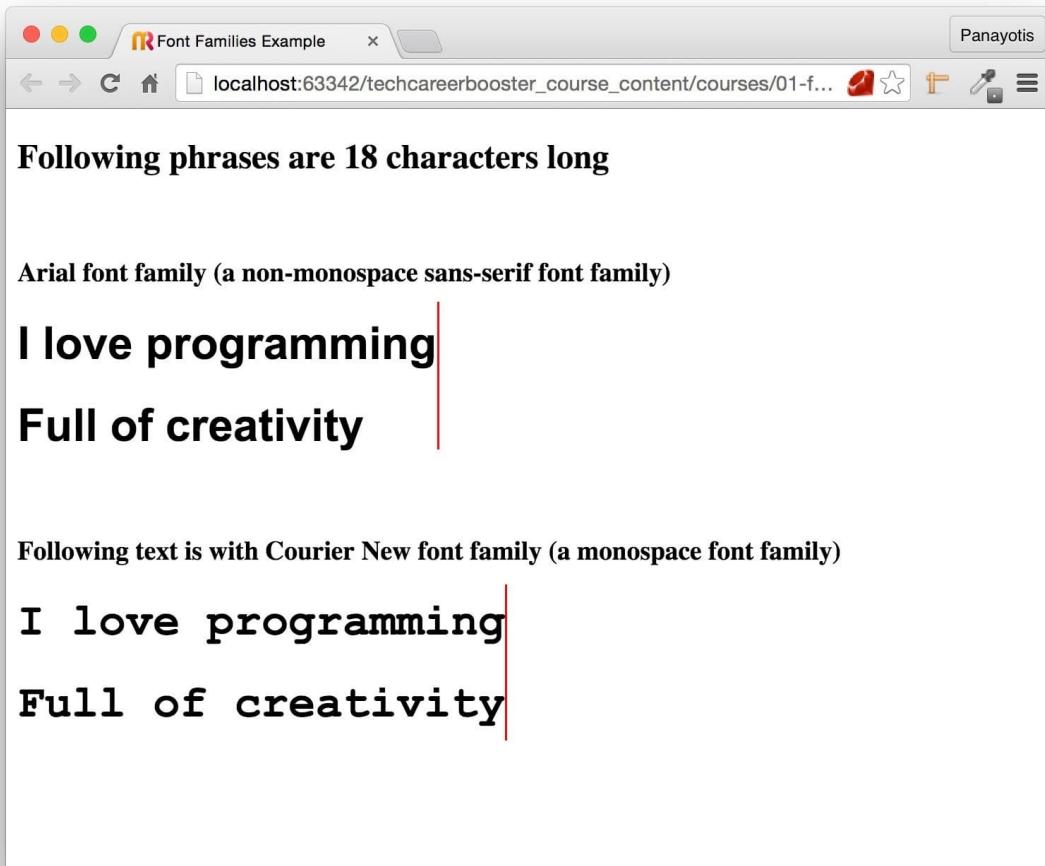
Programming is all about Creativity (Verdana)

Programming is all about Creativity (Courier New)

Programming is all about Creativity (Monaco)

Examples of a Phrase in various Font Families

We are going to teach you the small but important differences between them. For example, you are going to learn how the length of the text is affected by the font family chosen, and not only by the font size.



Comparison of non-monospace VS monospace font families

You are going to build a Web page like this:



Marked the Fonts We Want to Apply

with different font shapes, in order to practice applying different font families.

You are also going to learn about the default font families and how they might differ from one computer to the other.

This is a phrase written with ***serif*** general font family.

This is a phrase written with *sans-serif* general font family.

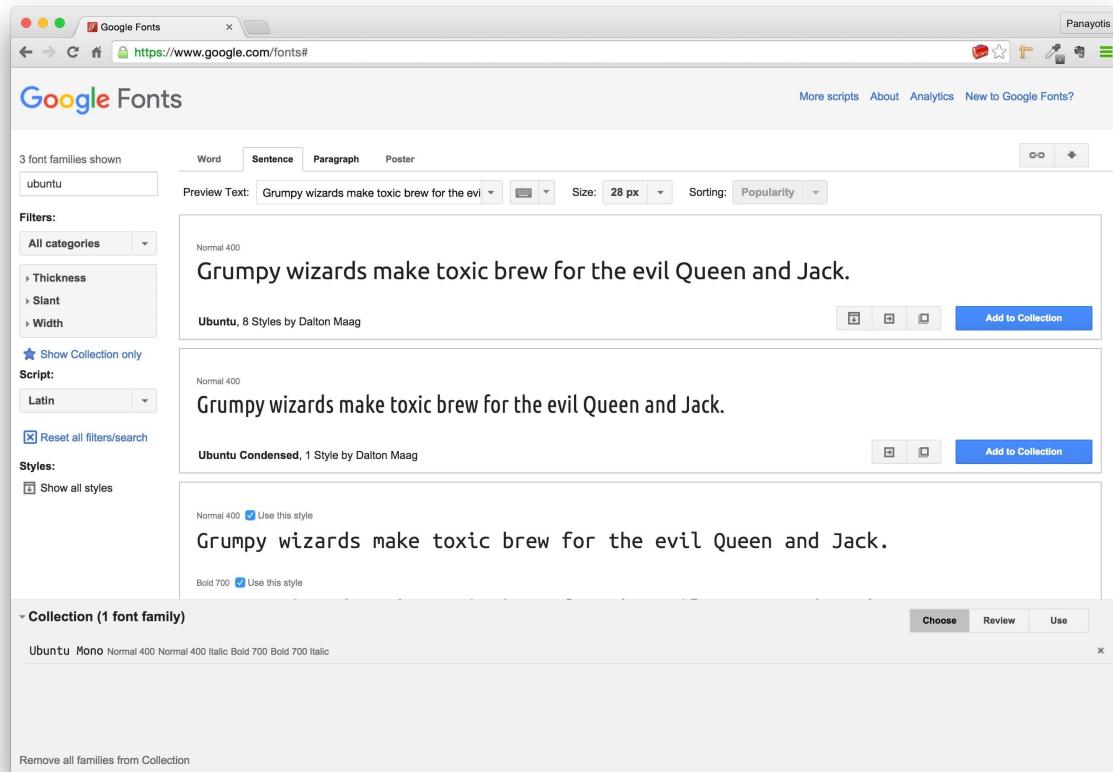
This is a phrase written with *monospace* general font family.

This is a phrase written with *cursive* general font family.

This is a phrase written with *fantasy* general font family.

All General Families with Default Fonts

Finally, you will also learn how to pick up a font family from one font providing service like Google Fonts:



Using a font family from Google Fonts

Learning Goals

1. Learn about the `font-family` CSS property.
2. Learn about the General Families.
 1. serif
 2. sans-serif
 3. monospace
 4. cursive
 5. fantasy
3. Learn how to specify the font family using CSS rules.
4. Learn about CSS Font Stack and how you specify fallback font families
5. Learn about the default font families based on the general family
6. Learn about the Safe CSS Font Stacks
7. Learn how to ask browser to use specific font that we want.
8. Learn how to use a font offered by a service like Google Fonts.
9. Learn about other font properties:
 1. `font-style`
 2. `font-weight`

Until now, in this course, we have used text on our pages, but we have not paid any particular attention to the shape of the characters. In this chapter we are talking about character shapes, but we will also expand on other properties of fonts.

Fonts and Typeface

Font Families

In order for us to be able to specify a particular shape of characters, we have decided to give different shapes a different name. So, the following are some examples of different shape names:

- Times New Roman
- Arial
- Verdana
- Courier New

We officially call them **Font Families**.

Having done that, we are able to specify which shape of characters we would like to use for our text in our HTML page.

Note that all characters in a font family follow the same design characteristics. And that's the reason of being in the same family. Which means, for example, that all lower case letters have the same height.

Look at the following pictures. You can see the same phrase printed on an HTML page, "Programming is all about Creativity", using different font families:

Programming is all about Creativity (Times New Roman)

Programming is all about Creativity (Arial)

Programming is all about Creativity (Verdana)

Programming is all about Creativity (Courier New)

Programming is all about Creativity (Monaco)

Examples of a Phrase in various Font Families

General Families

Besides the distinction of different shapes of characters, we have decided to group font families in the following, more generic families:

1. Serif
2. Sans-Serif
3. Monospace
4. Cursive
5. Fantasy

Serif

The Serif family groups all the font families that are using characters that have small lines at the ends of their characters. Look at the following picture. You can see a serif font family (Times New Roman). We have marked with red some of the lines that exist at the edges of the character shapes. These are the serifs.



Serif font family with some serifs marked with red color

Sans-Serif

The Sans-Serif family groups all the font families that do not have serifs, i.e. do not have small lines extending the edges of the characters. The word “sans” is a French word and means “without”. Look at the following picture and compare it to the previous one. The font family used is Arial, which is a font family without serifs, a sans-serif font family.

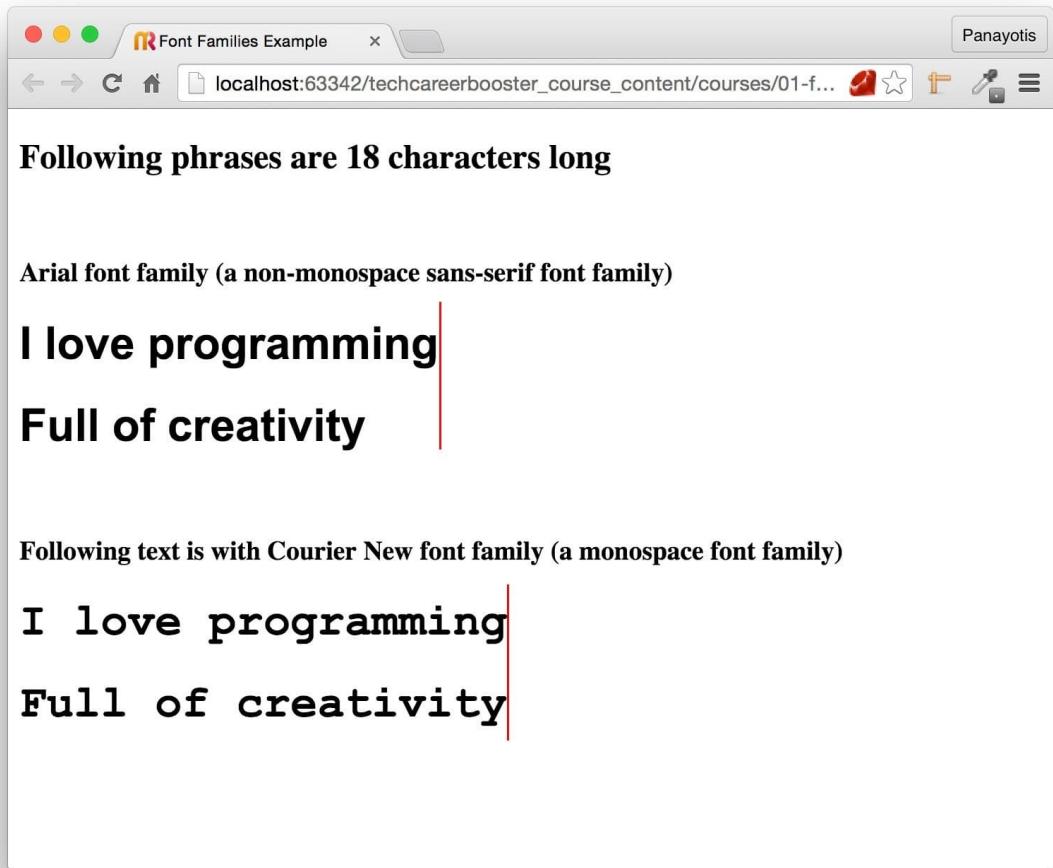


Sans-Serif font family without serifs

Monospace

The Monospace family groups all the font families that are using the same width for all of their characters. These families make sure that 1 character occupies the same horizontal space like any other in the same font family. So, they are more predictable when we have width concerns that we need to take care of. For example, two phrases, one on top of the other will occupy the same width space if they have the same number of characters, irrespective of which are the exact characters used.

Look at the following picture.

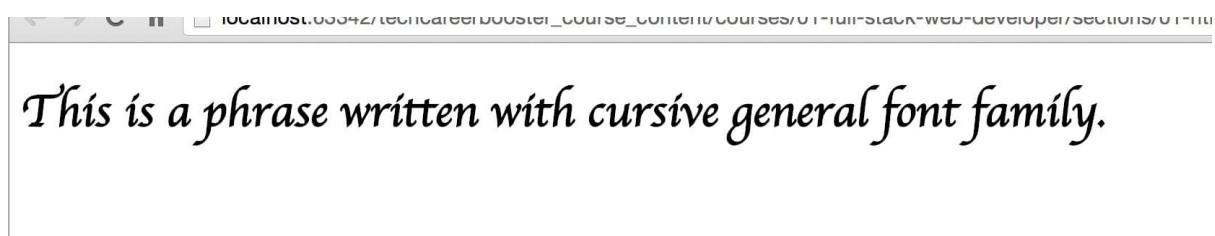


Comparison of non-monospace VS monospace font families

As you can see above, the same length phrases occupy different horizontal space, if we use non-monospace font families and if the phrases have different characters. On the other hand, the monospace font families use the same horizontal space for the same length phrases, even if these phrases have different characters.

Cursive

The cursive family groups all the font families that simulate the hand writing. Look at the following page fragment:



A cursive font family used here

Fantasy

The fantasy family groups all the font families that have decorative fonts, used for titles, e.t.c. Look at the following page fragment:



A fantasy font family used here

Specifying Font Family with CSS

In order to specify the font family that we want to use, we need to give value to the CSS property `font-family`. In fact, we usually give more than one alternatives, separated by commas, and we end up with the last alternative to be the name of one of the five general families. We give alternatives to the browser, because the font family that we primarily ask for, i.e. the one that we specify first, might not be supported by the browser of the user that is accessing our page.

So, if we want to use the `Verdana` font family, we can specify that as follows:

```
1 body {
2   font-family: Verdana, sans-serif;
3 }
```

(the above code snippet online)

In the above example, we ask browser to use `Verdana` for the typeface of our page. But if `Verdana` cannot be used for any reason, we ask browser to pick up any `sans-serif` font that it supports.

Note: Usually, we specify, as last alternative, a general family that is the same as the general family the primary font family belongs to. In the previous example, `Verdana` is a `sans-serif` family. Hence, we ask browser to pick up any other `sans-serif` family, in case `Verdana` cannot be applied.

We can specify more than one font families before giving the last one general family. Here is another example:

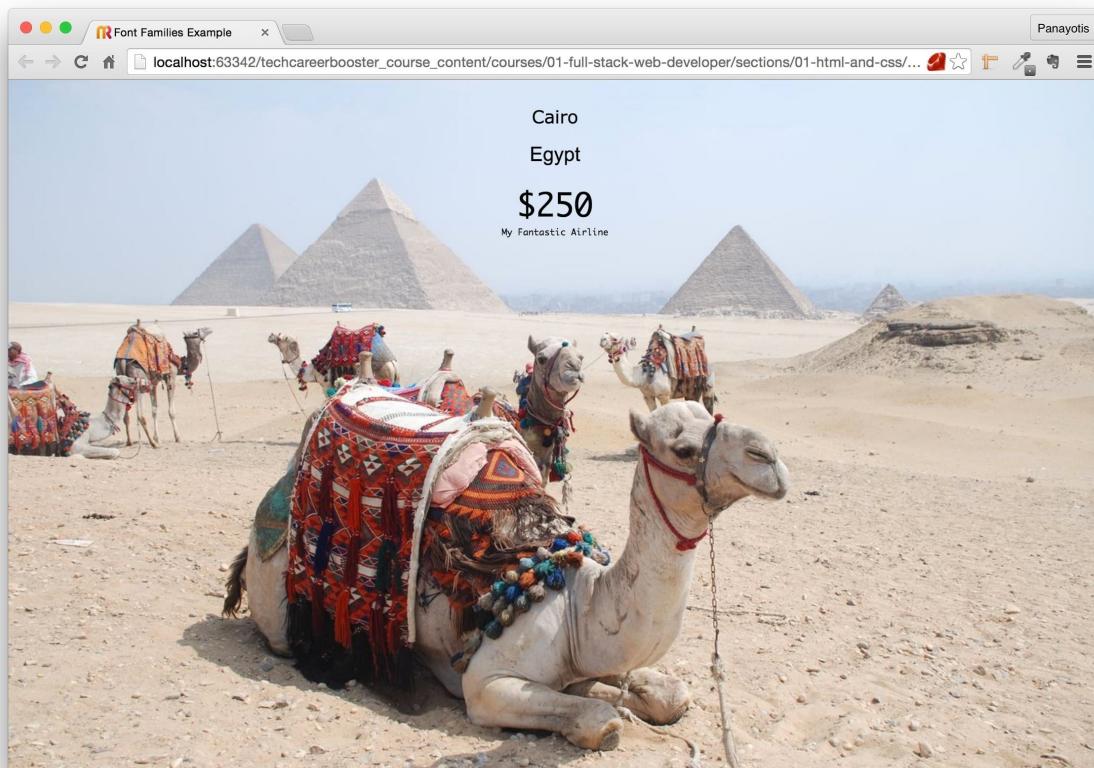
```
1 body {
2   font-family: Verdana, Arial, sans-serif;
3 }
```

(the above code snippet online)

In this example, we give as a second option `Arial`, which is another `sans-serif` font family. If browser cannot apply `Verdana`, it will try with `Arial` before resorting to any other `sans-serif` font family that it can support.

Note: If a font family name is composed of more than one word, then you need to enclose the name in double quotes.

Let's try our first example here. We are going to implement the following HTML page:



Simple Example with Font Families

Here is the content of this HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Font Families Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div id="city">
11             Cairo
12         </div>
13
14         <div id="country">
15             Egypt
```

```
16      </div>
17
18      <div id="price">
19          $250
20      </div>
21
22      <div id="airline">
23          <a href="#">My Fantastic Airline</a>
24      </div>
25  </body>
26 </html>
```

(the above code snippet online)

The content of the HTML page is very simple. What we need to take care of is the CSS content. In the following picture you can see which font families we want to apply on the texts of this page:



Marked the Fonts We Want to Apply

Here is the CSS content for this page:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     text-align: center;
14     margin-top: 1.6em;
15     background-image: url("../images/cairo.jpg");
16     background-repeat: no-repeat;
17     background-attachment: fixed;
18     background-position: center center;
19     background-size: cover;
20 }
21
22 #city {
23     font-size: 1.8rem;
24     font-family: Verdana, Arial, sans-serif;
25     margin-bottom: 1.6rem;
26 }
27
28 #country {
29     font-size: 2.0rem;
30     font-family: "Open Sans", Arial, sans-serif;
31     margin-bottom: 1.8rem;
32 }
33
34 #price, #airline {
35     font-size: 3.2rem;
36     font-family: "Lucida Console", Monaco, monospace;
37 }
38
39 #airline {
40     font-size: 0.8rem;
41 }
42
43 a {
44     text-decoration: none;
45     color: black;
46 }
```

(the above code snippet online)

The CSS rules relevant to our discussion are the following:

1) The word with the city:

```
1 #city {  
2     font-family: Verdana, Arial, sans-serif;  
3 }
```

(the above code snippet online)

It is requesting the browser to use Verdana font. If this font cannot be applied, we are requesting Arial font. If this is not applicable, then we are asking browser to apply any available sans-serif font.

2) The word for the country:

```
1 #country {  
2     font-family: "Open Sans", Arial, sans-serif;  
3 }
```

(the above code snippet online)

It is requesting the browser to use Open Sans font. If this font cannot be applied, we are requesting Arial font. If this is not applicable, then we are asking browser to apply any available sans-serif font.

3) The word for price and airline:

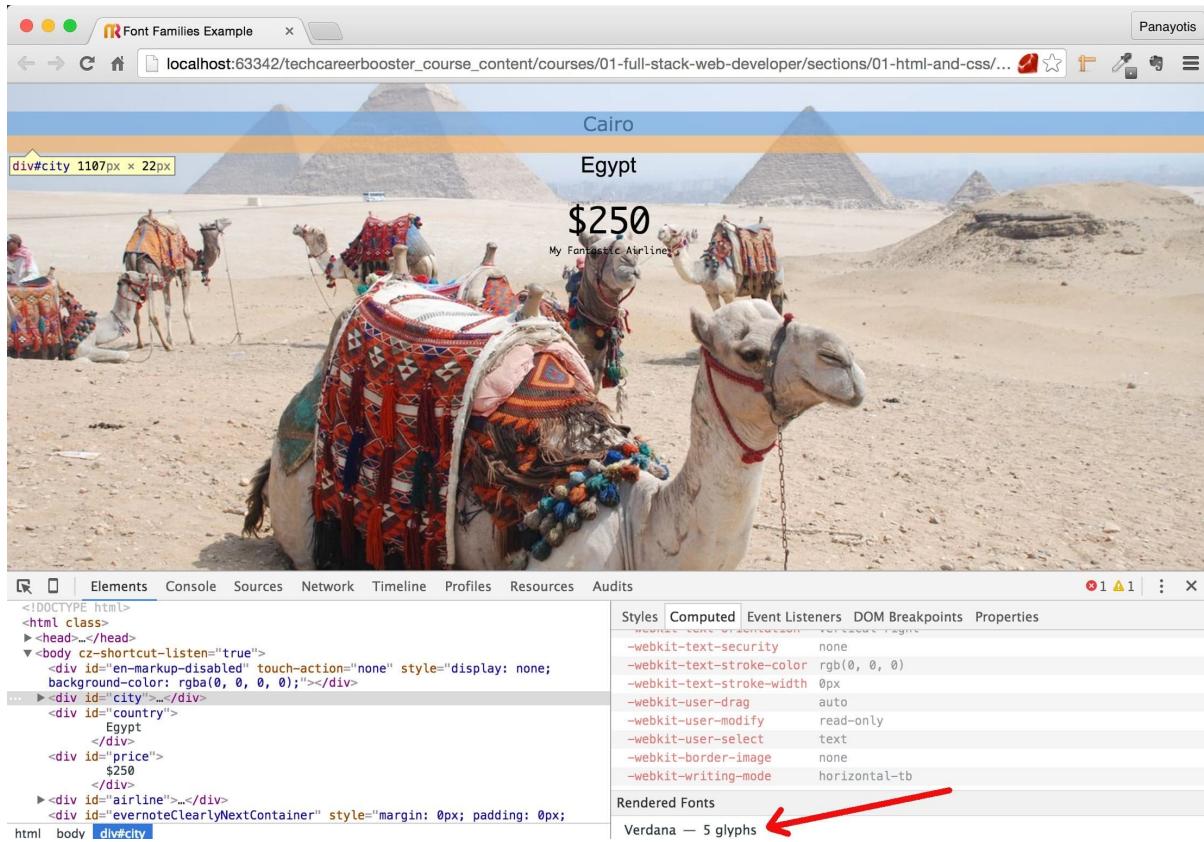
```
1 #price, #airline {  
2     font-family: "Lucida Console", Monaco, monospace;  
3 }
```

(the above code snippet online)

It is requesting Lucida Console, which is a monospace font family. If this font cannot be applied, we are requesting Monaco font. If this is not applicable, then we are asking browser to apply any available monospace font.

After we load the Web page on our browser, we can use the developer tools, in order to see which font was finally applied by the browser.

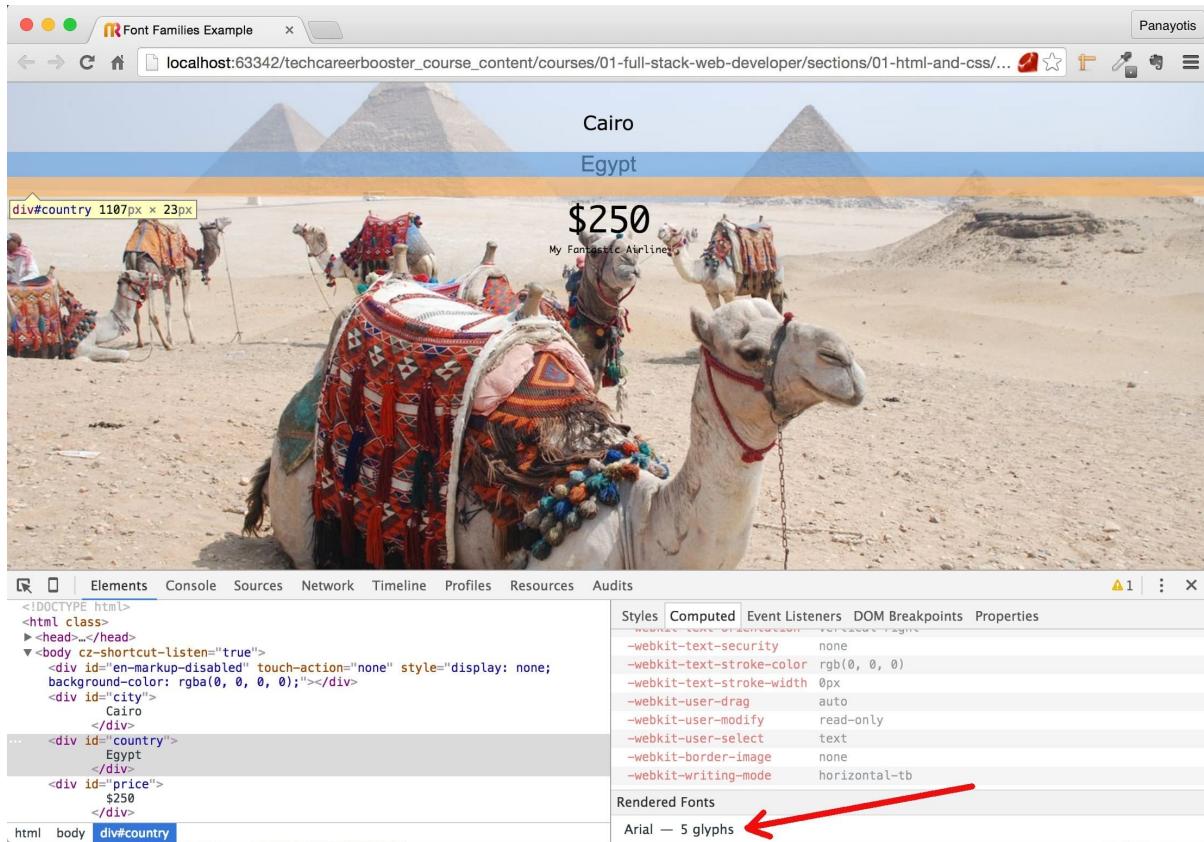
Look at the following picture:



City: Verdana Finally Applied

It shows how we can identify the actual font finally used for the city word ("Cairo"). If you have selected the item for the city on the left, and the "Computed" tab on the right, then scroll down at the bottom. You will see "Rendered Fonts" panel, in which it displays "Verdana - 5 glyphs". This proves that when we requested `Verdana, Arial, sans-serif;` for the city div, the browser has managed to apply the `Verdana` font and didn't have to resort to `Arial`.

Let's do the same for the country word. Inspect the element and identify the actual font used by the browser. You will see this:



Arial is applied for country

As you can see above, the “Open Sans” font has not been applied, because it was not found installed and supported on the client browser. The Arial font family was finally applied which was the second alternative.

CSS Font Stack

The fact that we can specify a series of font families, starting with the one that we want to be applied, but listing other alternatives too, ending with a general font family, is called CSS Font Stack. This is because the font families are stacked one on top of the other, and browser tries to apply first the one that is on top of the stack.

Default Font Families

If you want to see which font families your browser would pick up for any one of the general families, you can do that by just specifying the general family name and not being specific about the font family you would like to use. Let's see that:

Here is an HTML page that uses all the general family names:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Cursive Example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <h1 class="serif">This is a phrase written with <em>serif</em> general font f\
12 amily.</h1>
13         <h1 class="sans-serif">This is a phrase written with <em>sans-serif</em> gene\
14 ral font family.</h1>
15         <h1 class="monospace">This is a phrase written with <em>monospace</em> genera\
16 l font family.</h1>
17         <h1 class="cursive">This is a phrase written with <em>cursive</em> general fo\
18 nt family.</h1>
19         <h1 class="fantasy">This is a phrase written with <em>fantasy</em> general fo\
20 nt family.</h1>
21
22     </body>
23 </html>
```

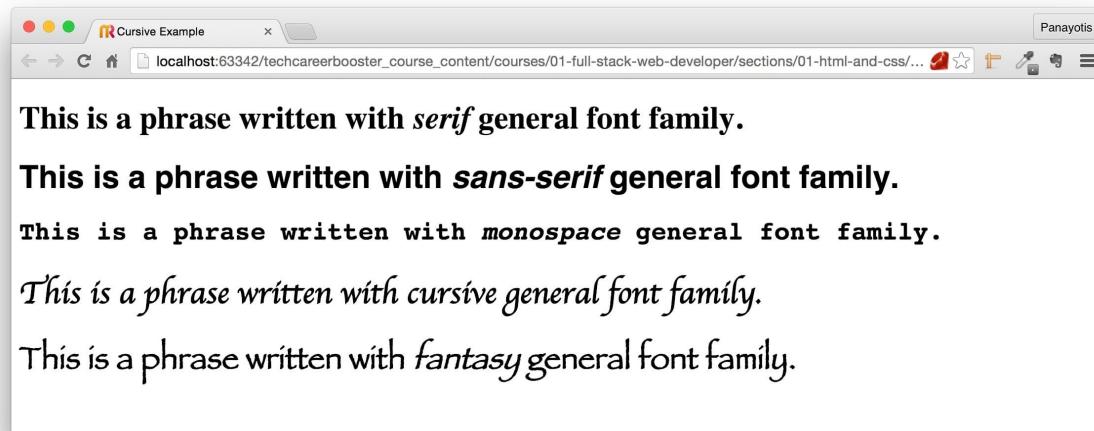
(the above code snippet online)

and the corresponding CSS:

```
1 .serif {
2     font-family: serif;
3 }
4
5 .sans-serif {
6     font-family: sans-serif;
7 }
8
9 .monospace {
10    font-family: monospace;
11 }
12
13 .cursive {
14     font-family: cursive;
15 }
16
17 .fantasy {
18     font-family: fantasy;
19 }
```

(the above code snippet online)

If you save the above files and load the page on your browser you will see this:



All General Families with Default Fonts

Let's inspect, one-by-one, the text lines above, in order to see what is the exact font that browser has finally applied on each case.

Important Note: The following results, i.e. the default font family picked up by your browser on your operating system, might be different from the one presented below. This is because the decision that browser takes depends on both the actual browser brand and the operating system the browser is hosted in.

- The serif case:

This is a phrase written with **serif** general font family.

This is a phrase written with **sans-serif** general font family.

This is a phrase written with **monospace** general font family.

This is a phrase written with **cursive** general font family.

This is a phrase written with **fantasy** general font family.

The screenshot shows a browser window with the URL localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-developer/sections/01-html-and-css/.... The developer tools are open, specifically the Elements tab, showing the HTML structure:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body cz-shortcut-listen="true">
    <h1 class="serif">...</h1>
    <h1 class="sans-serif">...</h1>
    <h1 class="monospace">...</h1>
    <h1 class="cursive">...</h1>
    <h1 class="fantasy">...</h1>
  </body>
</html>
```

The Computed tab in the developer tools shows the following styles for the selected **h1.serif** element:

- webkit-text-security: none
- webkit-text-stroke-color: rgb(0, 0, 0)
- webkit-text-stroke-width: 0px
- webkit-user-drag: auto
- webkit-user-modify: read-only
- webkit-user-select: text
- webkit-border-image: none
- webkit-writing-mode: horizontal-tb

The Rendered Fonts section shows "Times — 56 glyphs" with a red arrow pointing to it.

Times font is the default for serif general family

As you can see above, the default font for the serif general family is Times.

- The sans-serif case:

This is a phrase written with **serif** general font family.

This is a phrase written with **sans-serif** general font family.

This is a phrase written with **monospace** general font family.

This is a phrase written with **cursive** general font family.

This is a phrase written with **fantasy** general font family.

The screenshot shows a browser window with the same URL as the previous one. The developer tools are open, showing the HTML structure:

```
<!DOCTYPE html>
<html>
  <head></head>
  <body cz-shortcut-listen="true">
    <h1 class="serif">...</h1>
    <h1 class="sans-serif">...</h1>
    <h1 class="monospace">...</h1>
    <h1 class="cursive">...</h1>
    <h1 class="fantasy">...</h1>
  </body>
</html>
```

The Computed tab in the developer tools shows the following styles for the selected **h1.sans-serif** element:

- webkit-text-security: none
- webkit-text-stroke-color: rgb(0, 0, 0)
- webkit-text-stroke-width: 0px
- webkit-user-drag: auto
- webkit-user-modify: read-only
- webkit-user-select: text
- webkit-border-image: none
- webkit-writing-mode: horizontal-tb

The Rendered Fonts section shows "Helvetica — 61 glyphs" with a red arrow pointing to it.

Helvetica font is the default for sans-serif general family

As you can see above, the default font for the sans-serif general family is Helvetica.

- The monospace case:

This is a phrase written with ***serif*** general font family.

This is a phrase written with ***sans-serif*** general font family.

This is a phrase written with ***monospace*** general font family.

This is a phrase written with ***cursive*** general font family.

This is a phrase written with ***fantasy*** general font family.

The browser's developer tools are open, showing the DOM structure and the computed styles for the **h1.monospace** element. A red arrow points to the 'Rendered Fonts' section, which shows 'Courier — 60 glyphs'.

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body cz-shortcut-listen="true">
    <h1 class="serif">...</h1>
    <h1 class="sans-serif">...</h1>
    <h1 class="monospace">...</h1>
    <h1 class="cursive">...</h1>
    <h1 class="fantasy">...</h1>
  </body>
</html>
```

| Styles | Computed | Event Listeners | DOM Breakpoints | Properties |
|---------------------------|---------------|-----------------|-----------------|------------|
| -webkit-text-security | none | | | |
| -webkit-text-stroke-color | rgb(0, 0, 0) | | | |
| -webkit-text-stroke-width | 0px | | | |
| -webkit-user-drag | auto | | | |
| -webkit-user-modify | read-only | | | |
| -webkit-user-select | text | | | |
| -webkit-border-image | none | | | |
| -webkit-writing-mode | horizontal-tb | | | |

Rendered Fonts
Courier — 60 glyphs

Courier font is the default for monospace general family

As you can see above, the default font for the monospace general family is Courier.

- The cursive case:

This is a phrase written with ***serif*** general font family.

This is a phrase written with ***sans-serif*** general font family.

This is a phrase written with ***monospace*** general font family.

This is a phrase written with ***cursive*** general font family.

This is a phrase written with ***fantasy*** general font family.

The browser's developer tools are open, showing the DOM structure and the computed styles for the **h1.cursive** element. A red arrow points to the 'Rendered Fonts' section, which shows 'Apple Chancery — 58 glyphs'.

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body cz-shortcut-listen="true">
    <h1 class="serif">...</h1>
    <h1 class="sans-serif">...</h1>
    <h1 class="monospace">...</h1>
    <h1 class="cursive">...</h1>
    <h1 class="fantasy">...</h1>
  </body>
</html>
```

| Styles | Computed | Event Listeners | DOM Breakpoints | Properties |
|---------------------------|---------------|-----------------|-----------------|------------|
| -webkit-text-security | none | | | |
| -webkit-text-stroke-color | rgb(0, 0, 0) | | | |
| -webkit-text-stroke-width | 0px | | | |
| -webkit-user-drag | auto | | | |
| -webkit-user-modify | read-only | | | |
| -webkit-user-select | text | | | |
| -webkit-border-image | none | | | |
| -webkit-writing-mode | horizontal-tb | | | |

Rendered Fonts
Apple Chancery — 58 glyphs

Apple Chancery font is the default for cursive general family

As you can see above, the default font for the cursive general family is Apple Chancery.

- The fantasy case:

This is a phrase written with ***serif*** general font family.

This is a phrase written with ***sans-serif*** general font family.

This is a phrase written with ***monospace*** general font family.

This is a phrase written with ***cursive*** general font family.

Papyrus — 58 glyphs

Papyrus font is the default for fantasy general family

As you can see above, the default font for the fantasy general family is Papyrus.

In Summary:

Results for Chrome 47.0, Mac OS X 10 - Yosemite

| General Family | Default Font |
|----------------|----------------|
| serif | Times |
| sans-serif | Helvetica |
| monospace | Courier |
| cursive | Apple Chancery |
| fantasy | Papyrus |

We ran the same test on Chrome 47.0 on Windows 7. Here are the results:

Results for Chrome 47.0, Windows 7

| General Family | Default Font |
|----------------|-----------------|
| serif | Times New Roman |
| sans-serif | Arial |
| monospace | Consolas |
| cursive | Comic Sans MS |
| fantasy | Impact |

We ran the same test on Chrome 47.0 on Ubuntu 12. Here are the results:

Results for Chrome 47.0, Ubuntu 12

| General Family | Default Font |
|----------------|------------------|
| serif | Times New Roman |
| sans-serif | Arial |
| monospace | DejaVu Sans Mono |
| cursive | Comic Sans MS |
| fantasy | Impact |

As you can understand, when you specify generic font families for your `font-family` property, the result is not always the same. It depends on the operating system and browser. Hence, do not specify general font families. Only as final fallbacks. User specific font families.

Safe CSS Font Stacks

The above discussion reveals the fact that, it is very important to specify a CSS font stack that makes sense and has quite good chances to be applied as specified.

Here are some good examples:

- sans-serif
 - `font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;`
 - `font-family: Verdana, Geneva, sans-serif;`
- serif
 - `font-family: TimesNewRoman, "Times New Roman", Times, Baskerville, Georgia, serif;`
 - `font-family: "Gill Sans", "Gill Sans MT", Calibri, sans-serif;`
- monospace
 - `font-family: "Arial Black", "Arial Bold", Gadget, sans-serif;`
 - `font-family: "Lucida Sans Typewriter", "Lucida Console", Monaco, "Bitstream Vera Sans Mono", monospace;`
- cursive
 - `'font-family: "Apple Chancery", "Comic Sans MS", cursive;`

There are a lot of sites on Internet that can give you nice CSS Font Stacks. Google for `web safe font stacks`.

Make Sure Browser Is Using the Font We Want

It has become quite common, nowadays, to not rely on the font families that are supported by default by the browser/operating system combination. We want our Web sites to be more specific and use the fonts that we want, without being very common (like using Arial or Verdana), and without falling back to other font families. In order to make sure that our site uses the font that we want, we can tell the browser to download the font and install it from a specific location on the Internet, or have it as a resource, already part of the resource bundle of our site.

Using Fonts Provided by Google

One very good resource for you to use a font family is Google. In fact, the [Google Fonts](#) web site. Visit this site and pick up the font family that best matches your requirements.

One way you can work with Google Fonts is the following:

- 1) You need to create a Collection, by first choosing one or more of the font families that you want and Adding them to the Collection.

Here is a video that demonstrates how we use the filters on the Google Fonts site, to select a font family that matches specific criteria and add it to our collection.

[How we can select a font family from Google Fonts and add it to our collection](#)

In this video, the user selects the Catamaran font family.

- 2) You need to review the font and add any extra style that you want and it is not pre-selected. For example, you may want to finally select all the bold styles.

Here is the next part of the video that shows how the user reviews the properties of the font family selected and selects any style that is not already selected.

[How we can review the properties of a font family selected](#)

- 3) You finally need to use the font, by first following the directions on the “Choose” tab of the Google Fonts. Watch the following video:

[How we can use a font family”>How we can use a font family](#)

One common way to use a font is to create a `link` tag in the `head` section of our HTML page that would reference the font resource. This is actually a remotely loaded CSS file.

For example, let's use the font selected by the user in the video above. As you can see from the last part of the video, the one that user accesses the “Use” tab, there is a `link` tag code, that user can copy and paste directly to his HTML head section.

This is the `link` code:

```
1 <link href="https://fonts.googleapis.com/css?family=Catamaran:400,200,300,900,100\ ,500,600,700,800&subset=latin,latin-ext" rel="stylesheet" type="text/css">
```

[\(the above code snippet online\)](#)

Let's put that inside an HTML page and write some text and see whether the font is indeed Catamaran:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Using a Google Font</title>
6          <link href="https://fonts.googleapis.com/css?family=Catamaran:400,200,300,900\,
7 ,100,500,600,700,800&subset=latin,latin-ext" rel="stylesheet" type="text/css">
8          <link href="stylesheets/main.css" rel="stylesheet" type="text/css">
9      </head>
10
11     <body>
12
13         <h1>This is a phrase with font family 'Catamaran'</h1>
14
15     </body>
16
17 </html>

```

(the above code snippet online)

As you can see above, we put our own CSS file (`stylesheets/main.css`) after the reference to the Google font Catamaran resource. This is necessary in order for us to be able to use the Catamaran font family within our own CSS rules. Our own CSS rules have as follows:

```

1  html {
2      font-size: 62.5%;
3  }
4
5  body {
6      font-size: 1.6rem;
7  }
8
9  h1 {
10     font-family: Catamaran, sans-serif;
11 }

```

(the above code snippet online)

Very simply, we specify that the `h1` tag content will be drawn with the font family Catamaran. If this cannot be loaded for some reason, we fallback to `sans-serif`. We chose `sans-serif` as a fallback because Catamaran is a `sans-serif` font family. As long as Google font Catamaran can be loaded, then falling back to `sans-serif` will never take place. And Google site with fonts being unavailable has very small, if any, chances. Hence, fall back to `sans-serif` is very unlikely. But still, we include that to be on the safe side.

If you save the above two files and load the page on your browser you will see this:



A Simple Page with Catamaran Font Used

And if you inspect the h1 tag with developer tools, you will see this:

| Computed | |
|---------------------------|---------------|
| -webkit-text-security | none |
| -webkit-text-stroke-color | rgb(0, 0, 0) |
| -webkit-text-stroke-width | 0px |
| -webkit-user-drag | auto |
| -webkit-user-modify | read-only |
| -webkit-user-select | text |
| -webkit-border-image | none |
| -webkit-writing-mode | horizontal-tb |

| Rendered Fonts | |
|----------------|-------------|
| Catamaran | — 45 glyphs |

Verifying Catamaran Font Family Used

As you can see above, the font family used for the content of h1 tag, is Catamaran.

Other Properties of the Fonts

Besides the font family, there are some other CSS properties that are related to the font one wants to apply to an HTML text.

font-style

Takes the three values:

1. **normal**. In that case the text is shown normally.
2. **italic**. In that case the text is shown in italics.
3. **oblique**. In that case the text is shown similar to italic. This property is not widely supported.

font-size

This specifies the size of the font. We have extensively talked about this in the chapter CSS Units.

font-weight

This specifies the weight of the font. It can take values like:

1. **normal**. Which means no bold.
2. **bold**. Bold text.
3. **bolder**. For even more bold text.
4. An integer that corresponds to a weight according to the font family used. For example, earlier we used the font family Catamaran from the Google Fonts. There, we have said that we want to be able to use the weight versions: 100,200,300,400,500,600,700,800,900, which are all the weight version this font family has been designed for. This means that we can write CSS rules like the following:

```
1 body {  
2   font-family: Catamaran, sans-serif;  
3   font-weight: 200;  
4 }
```

(the above code snippet online)

which picks up the font weight with value 200.

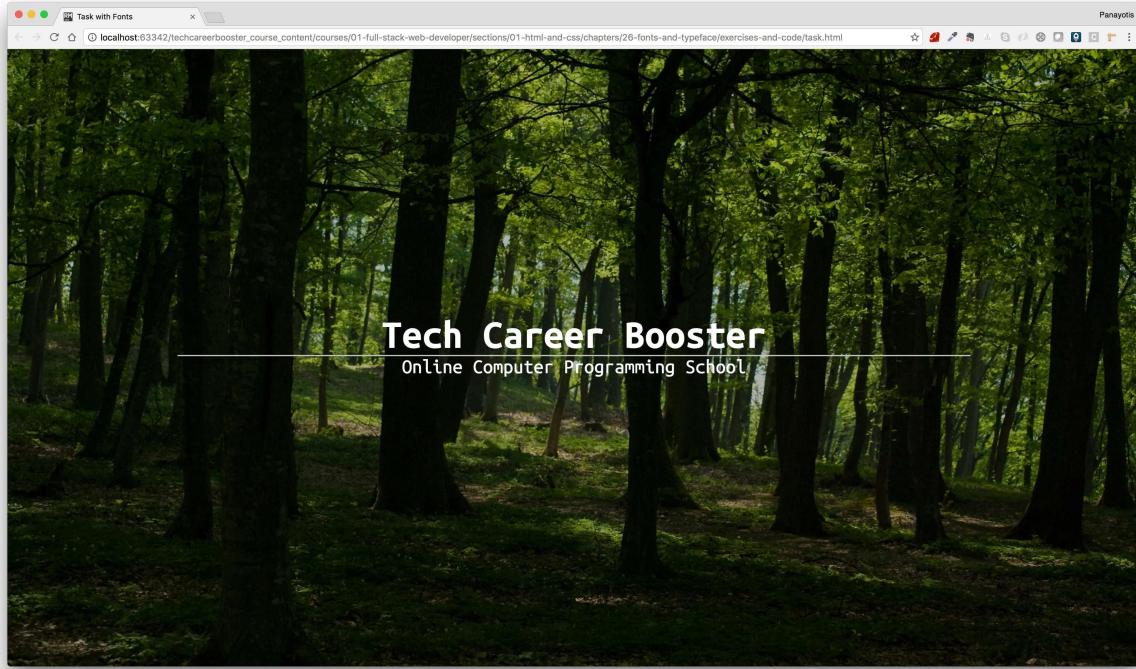
Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. You need to implement the following HTML page:

**Task with Font Families**

2. Make sure you take care of the following:
 1. You use the Google Font with name “Ubuntu Mono”.
 2. The styles that you should include in your font collection are:
 - 400
 - 400 Italic
 - 700 and
 - 700 Italic
 3. “Tech Career Booster” should be included inside an `h1`.
 4. “Online Computer Programming School” should be included inside a `small` tag.
 5. There is a `div` that will help you make the image a little bit darker than its original color so that the text stands out clearly. This `div` needs to cover the whole background, like the background image does. And it has to have a background color `rgba(0, 0, 0, 0.3)`.
 6. There is a horizontal line that does not extend from left to right. It has 70% width. But it is centered. Hint: the value `auto` on left and right margins allows elements with specific width to be centered.
 7. All elements, by default, should be using `border-box` for `box-sizing`. Also 0 padding and 0 margin.
 8. Use `rem` unit for the size of text.

9. The `h1` font size should end up being 64px.
10. the `small` font size should end up being 32px.
11. There is some margin on top of the body element, which needs to be calculated as relative unit of the viewport height (vh). 43 is a good value for that.
3. Upload your site to your Netlify account.

Important: Try to use your own brand, instead of Tech Career Booster brand, logos and motos. These are pages that you create and you should promote your profile and your work. This is a general rule and does not apply to this task only.

Chapter 27 - HTML Forms

Summary

This chapter enters the interactive world of HTML. We will have an introduction on the `form` element and the most important UI elements that are used to collect information from the user.

You will build pages like the following:

The screenshot shows a web browser window with a title bar 'Billing Details' and a status bar 'localhost:63342/techcareerbooster_course_content/courses/01-full-sta...'. The main content area contains the following form:

Enter your address details:

First Name:

Last Name:

Address:

Zip Code:

Country: Afghanistan

Submit

Simple Form for Collecting Address Details

And not only that. We will spend some time tuning the input element properties. For example, we will add validations like this:

A basic input example

localhost:63342/techcareerbooster_course_content/courses/01-full-s... Panayotis

Enter your address details:

First Name:

Last Name:

Address:

Zip Code:

Country:

Submit

First Name is Flagged As Required

We will learn to use specific types of input elements, like emails and passwords. For example, in order to build a sign up form:

The screenshot shows a web browser window with the title bar "Sign Up Form". The address bar displays "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-...". The main content area is titled "Sign Up" and contains the following fields:

- First Name: [input field]
- Last Name: [input field]
- Email (*): [input field]
- Password (*): [input field]

Below the input fields is a checkbox labeled "Receive newsletter" followed by a "Sign Up" button.

A Sign Up Form

We will learn to use a series of checkboxes in order to let the user tell us which options would he prefer:

Pick Up Playstation Games

localhost:63342/techcareerbooster_course_content/courses/01-full-stack-... Panayotis

Which are your favourite PS Games?

Star Wars Battlefront
 Just Cause 3
 Fall Out 4
 EA SPORTS FIFA 16

Submit

Multiple Checkboxes

We will also learn how we could select one or more options using multiple select boxes:

Pick Up Playstation Games

localhost:63342/techcareerbooster_course_content/courses/01-full-stack-... Panayotis

Which are your favourite PS Games?

Select Game:

Star Wars Battlefront
Just Cause 3
Fall Out 4
EA SPORTS FIFA 16

Submit

Multiple Select Box

Next we will learn how we could select one option amongst many using radio buttons:

Pick Up The Age Group

Please, specify your age range

0-18
 19-30
 31-45
 46-60
 61-80
 >80

Submit

Specify Age Range Using Radio Buttons

And we will learn how we can ask the user to enter a multi-line message like in a contact form:

A screenshot of a web browser window showing a contact form. The title bar says "Contact Us" and the address bar shows "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-w...". The page title is "Send Us Your Message". The form contains fields for First Name, Last Name, Email (*), and Your Message (*). A red "Send" button is at the bottom.

Send Us Your Message

First Name:

Last Name:

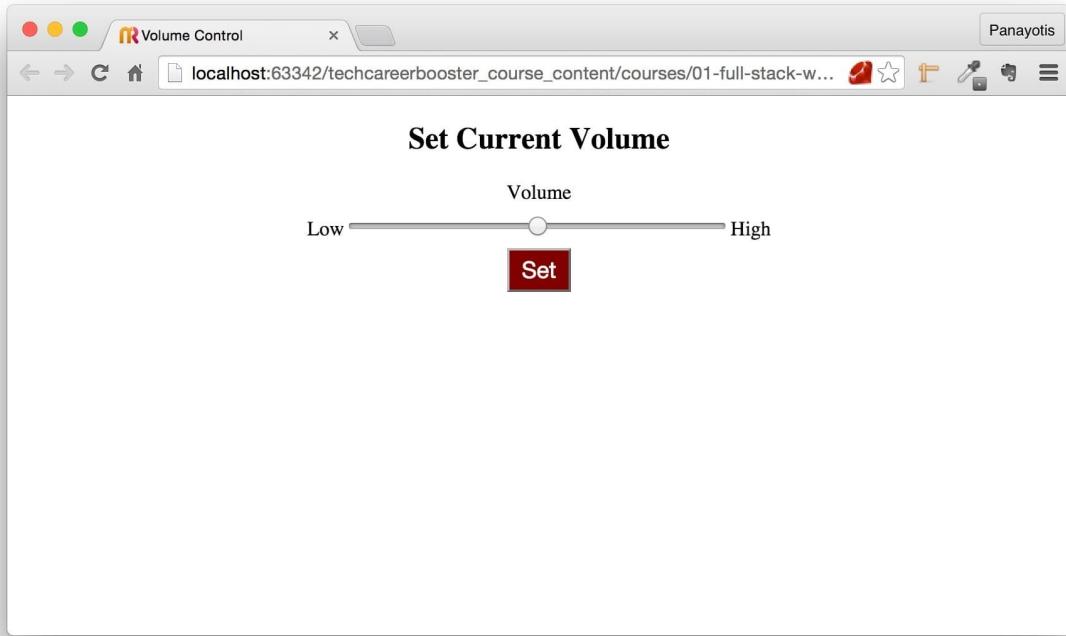
Email (*):

Your Message (*):

Send

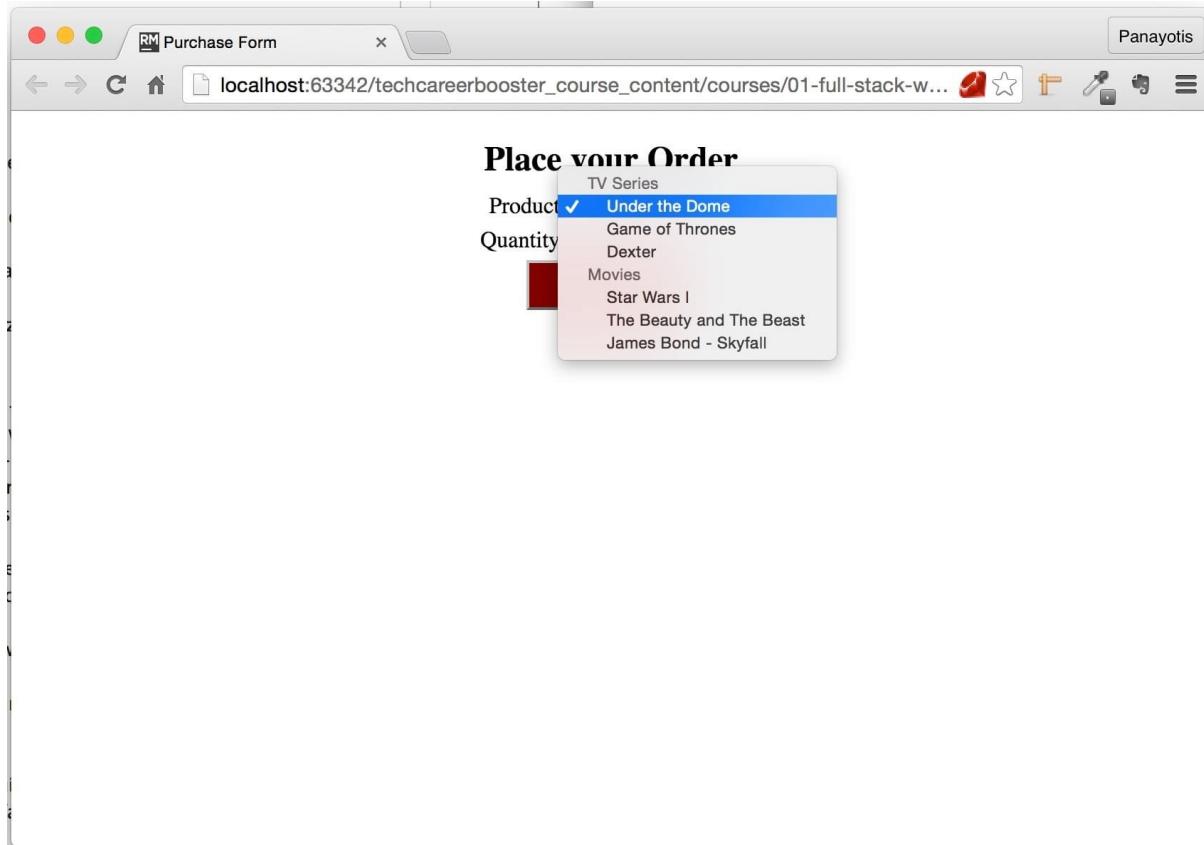
Contact Us Form

We will proceed in explaining other features of input elements too, like placeholders and titles, min and max attributes. We will use UI elements that visualize a range of values, like in the following volume slider:



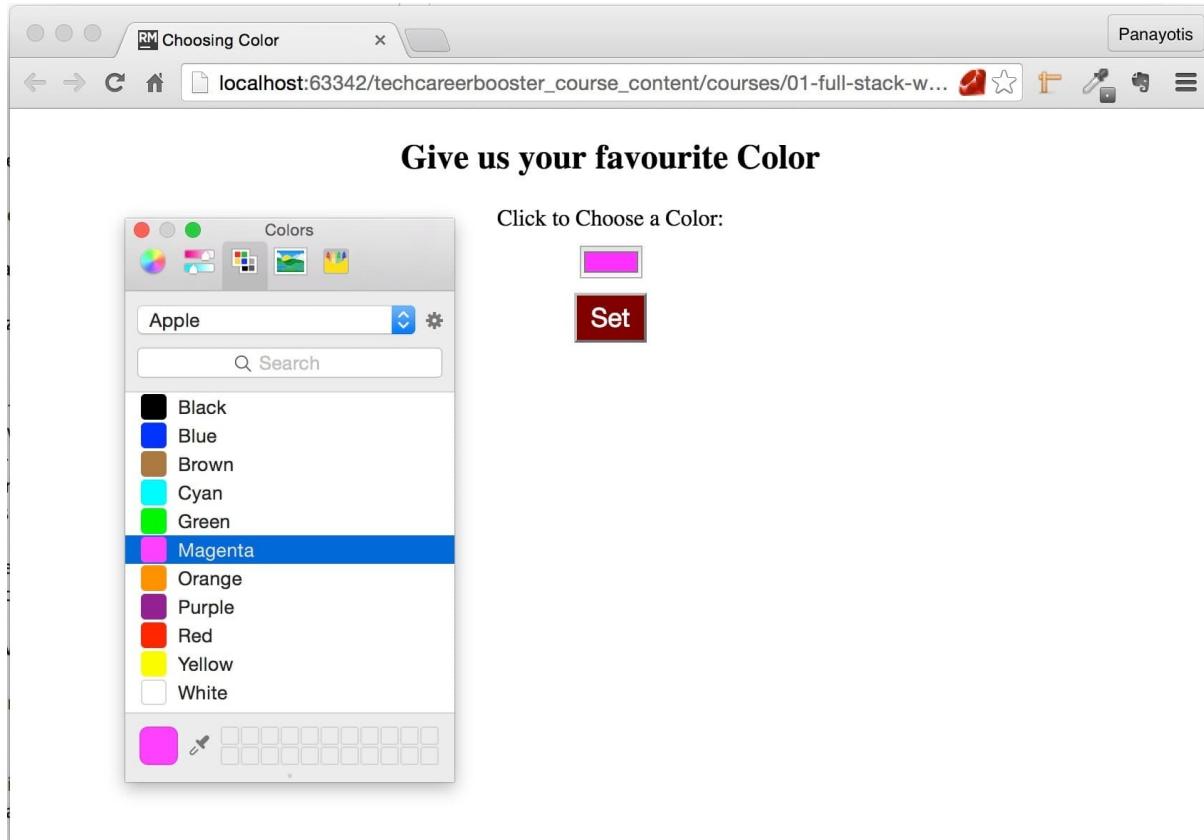
Form that is using range input

We will learn how we can group options in a drop down list:



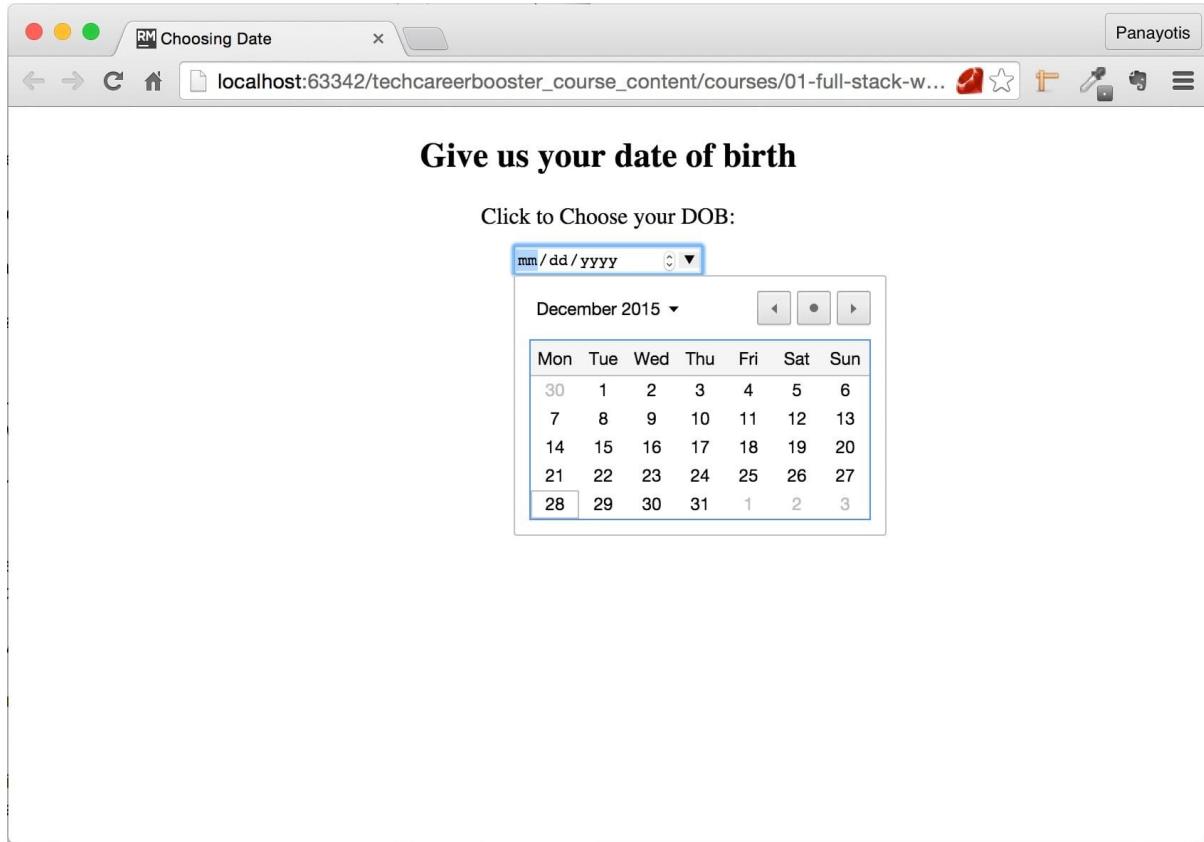
Grouping Options in a Drop Down List

We will learn how we can use a color picker to let user pick up a color:



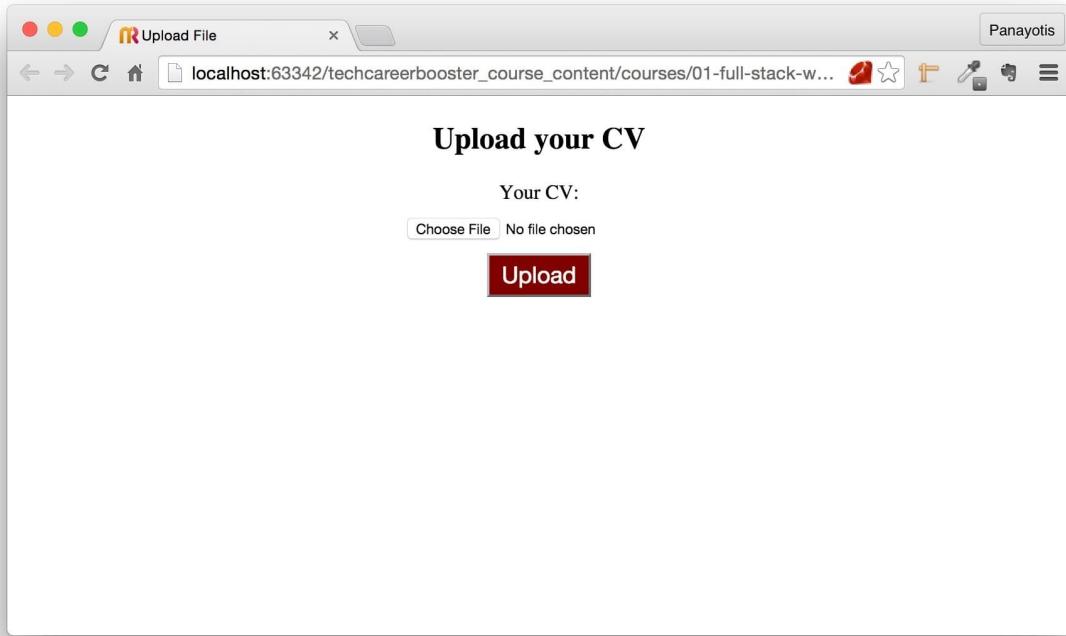
User selects color from a color picker

We will learn how we can use a date picker to let user pick up a date:



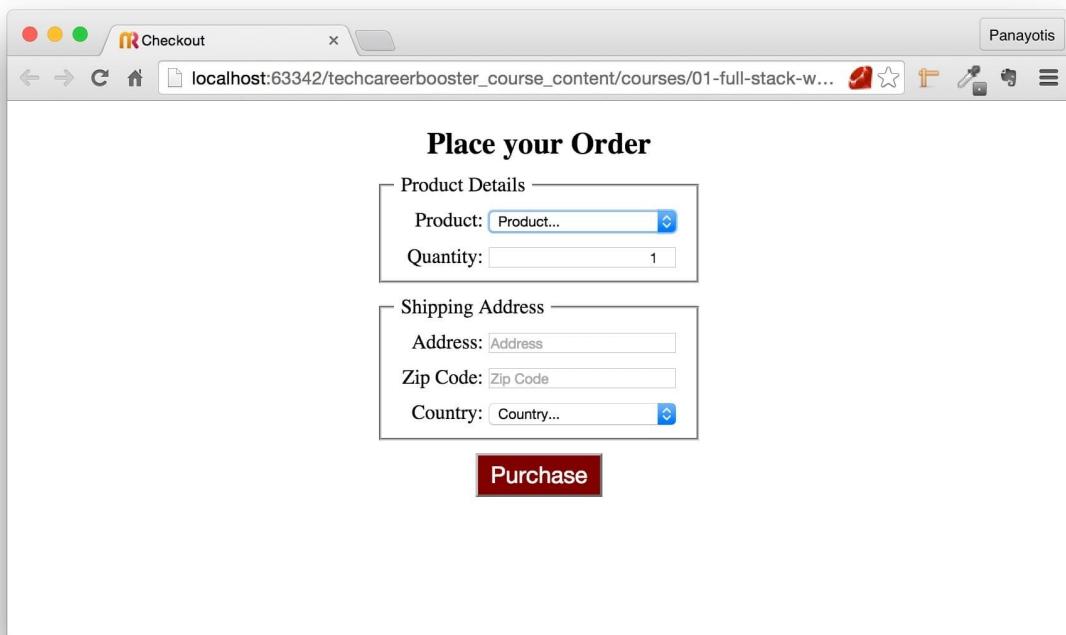
User picks up a date using a date picker

We will learn how we can ask the user to upload a file, like in this example:



Form to Upload a File

And we will finally learn how we can visually group form fields like in this example:



Checkout Form With Grouping of Fields

Your task, at the end of the chapter, will be to build a page with this form here:

The screenshot shows a web page titled "US Domestic Flights". The form is divided into several sections:

- Flight Details:** Includes fields for Departure City, Destination City, On Way checkbox, Dep. Date, and Ret. Date.
- Contact Details:** Includes fields for Email, Mobile, Address, City, Zip Code, and Country (with a "Select Country..." dropdown).
- Ticket Details:** Shows a section for "Passenger #1" and "Passenger #2". Each section has fields for First Name, Last Name, Passport, and DoB. There are also gender selection radio buttons (Male/Female) and a "Number of Tickets" input field set to 1.
- Your Message To Booking Agent:** A text area with placeholder text "Type in a message to booking agent (optional)".
- Search:** A red "Search" button at the bottom.

Task With Form

Learning Goals

1. Learn about the `form` HTML element.
2. Learn about `input` elements and their various types:
 1. `text` elements that allow the user to enter a one line text value.
 2. `email` elements that allow the user to type in an email address.
 3. `password` elements that allow the user to type in a password with actual characters being displayed as black bullets.
 4. `checkbox` elements that allow the user to check or uncheck an option.
 5. `radio` elements that allow the user to select one only option amongst many.
 6. `number` elements that are used to ask user to type in a number.
 7. `range` elements that are used to ask user to select a number from within a range.
 8. `color` elements that allow the user to define a color preference using a color picker UI element.
 9. `date` elements that allow the user to specify a date, using a date picker UI element.
 10. `file` elements that allow the user to upload a file.
 11. `submit` element which creates a button-like input element.
3. Learn about `label` element and how it is used to prompt the user about the information he has to key in.
4. Learn about the `select` element, which allows the user to select one or more options.

1. The single select version.
2. The multiple select version.
3. The grouping of options inside the list of options in the select box.
5. Learn how labels work together with `input` IDs, via the `for` attribute on the `label` element.
6. Learn about automatically setting the focus on a specific input element, using the `autofocus` attribute.
7. Learn about the `required` attribute that allows us to define which pieces of information are mandatory for the user to fill in.
8. Learn about the `checked` attribute that can be used to pre-select a checkbox or a radio button.
9. Learn about the `textarea` element which allows the user to type in long multiple-line text.
10. Learn about the `placeholder` attribute which can, sometimes, replace the labels and works as an alternative prompt.
11. Learn about the `title` attribute which is used to give more explanation about the information required by an input field.
12. Learn about the `autocomplete` attribute and how this can be switched on / off.
13. Learn about the `min` and `max` attributes that can limit the range of values a user can type in in a `number` input.
14. Learn about the `step` attribute that allows us to define the granularity of the numbers that are allowed to be keyed in in a `number` input.
15. Learn about the `value` attribute and its various uses in input elements.
16. Learn about visually grouping the fields of a form using the elements `fieldset` and `legend`.

HTML Forms

We have learned a lot about how we can style our HTML pages, and it is about time to return back to core HTML and continue learning about more HTML tags. In this chapter, we introduce one of the most important tags in HTML. It is the `form` tag which is used to collect input from the user.

Important: This chapter deals with the `form` tag without taking into account the actual processing of the data that we collect from the users. Posting data to the server and further processing will be dealt with in later chapters.

Let's start. Assume that we want to ask our users to key in their address details. We will build the following HTML page:

The screenshot shows a web browser window with a title bar 'Billing Details'. The address bar displays 'localhost:63342/techcareerbooster_course_content/courses/01-full-sta...'. The main content area has a heading 'Enter your address details:'. Below it is a form with five fields: 'First Name' (with a blue border), 'Last Name', 'Address', 'Zip Code', and 'Country' (a dropdown menu showing 'Afghanistan'). A large red 'Submit' button is at the bottom.

Simple Form for Collecting Address Details

On this video, you can see the form being used by a user:

How the Address Details Form is being filled in by the user

As you can see from above, there are 4 fields that the user has to fill in with free text and 1 field that the user is being asked to select one of the possible values.

input of type="text"

The first 4 fields, for first name, last name, address and zip code, are being constructed using the HTML tag `input`. There are different types of `input` HTML tags, and each one of them has a different value for the tag attribute with name `type`. The ones that we are using here, they are all `input` tags of type with value `text`. Later on, we are going to learn about other types of `input` tags.

This one here:

```
1 <input type="text"/>
```

(the above code snippet online)

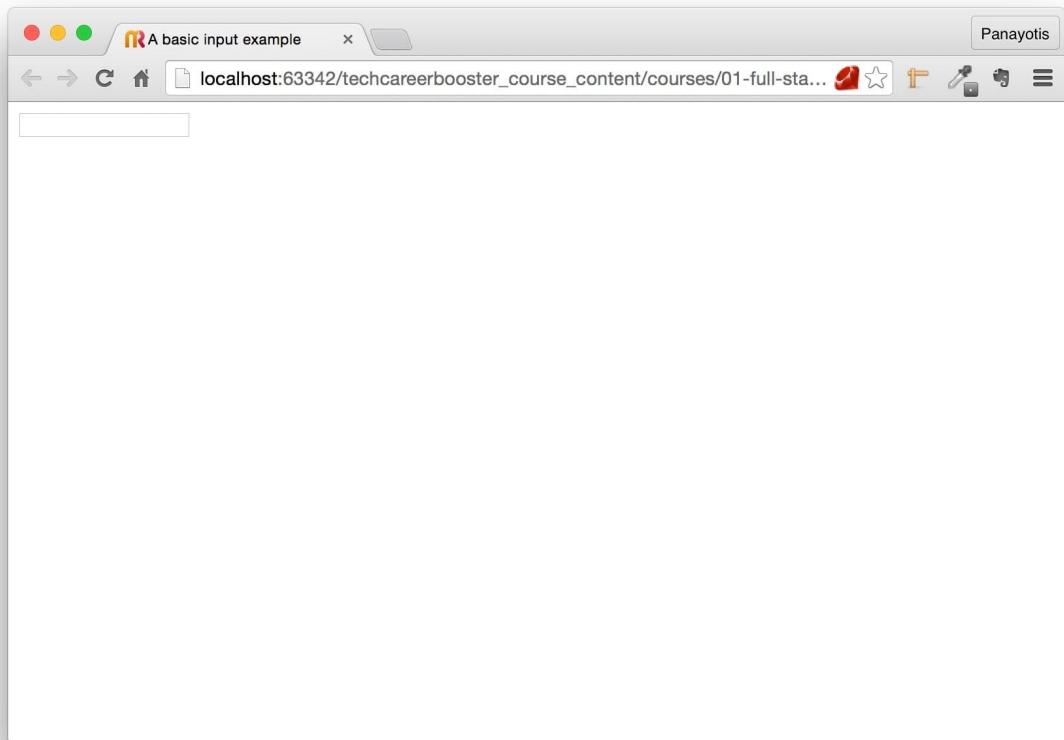
declares an input field that allows the user to key in free text. The free text that the user can type in is limited in 1 line only. As you can see the `input` HTML tag does not have any HTML content. That is why we are using the shorthand notation to close it (`/>`).

Try this HTML code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6      </head>
7
8      <body>
9          <input type="text"/>
10     </body>
11 </html>
```

(the above code snippet online)

If you save the above HTML code into an html file and load it on your browser, you will see this:



Simple Page with Bare Minimum Input Field of Type text

You will be able to click inside the box area of this input field and type in anything that you like. The fact that we have defined the type of the input to be text allows us to use this input like that.

label

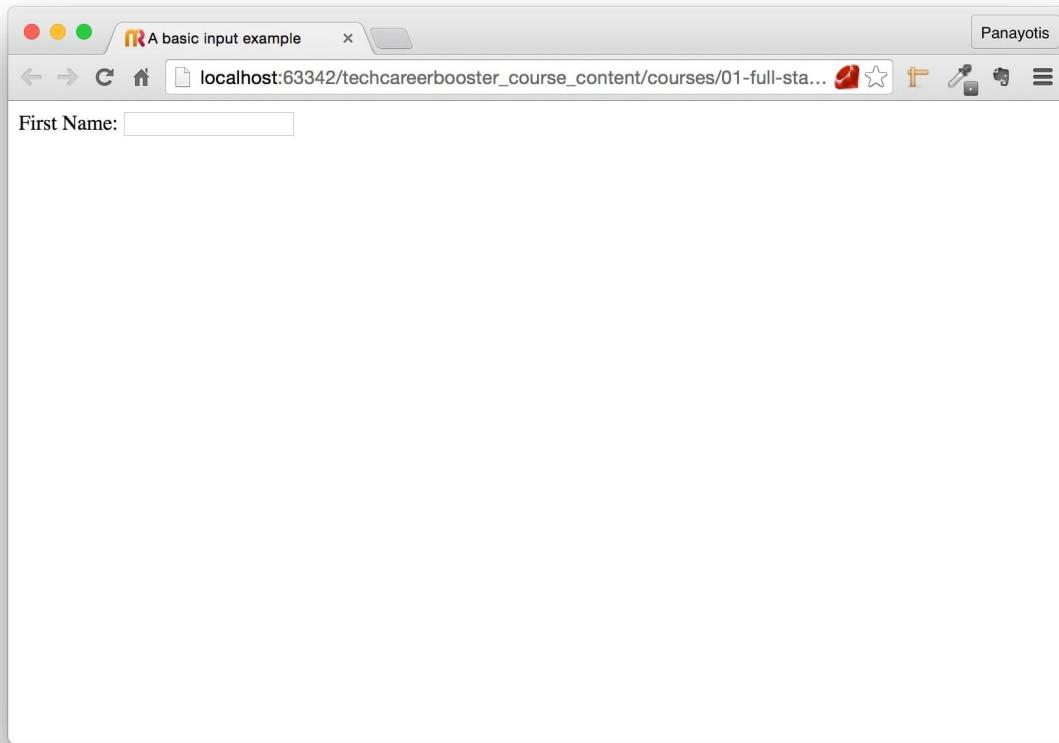
The problem with this input field, is that, if presented alone, does not prompt the user about the data that they have to key in. Will that be the first name? Will that be the last name? Will that be something else? In order to help user understand what we are expecting as input, we traditionally use labels. The labels are constructed with the `label` tag. This is very simple. You define the tag and you set as its HTML content the exact content that you want to use as a prompt to the user.

Let's add our first label to the above HTML page.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6      </head>
7
8      <body>
9          <label>First Name:</label>
10         <input type="text"/>
11     </body>
12 </html>
```

(the above code snippet online)

If you save the above HTML code and reload your page, you will see this:



A label is used before the input

Now, the user of our Web page, knows that on the first field he has to type in his first name.

Let's add the other 3 fields on the form. They will be added using the same technique that we did for the first name. Pretty easy.

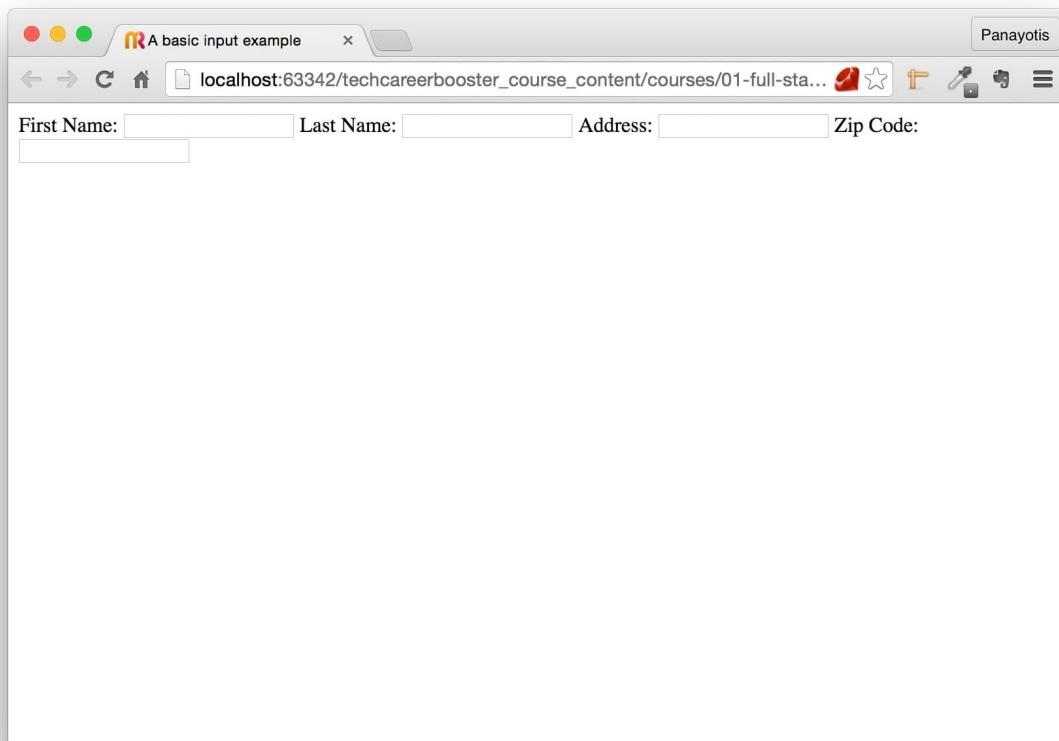
Here is the HTML code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6      </head>
7
8      <body>
9          <label>First Name:</label>
10         <input type="text"/>
11
12         <label>Last Name:</label>
13         <input type="text">
14
15         <label>Address:</label>
16         <input type="text">
```

```
17
18     <label>Zip Code:</label>
19     <input type="text">
20   </body>
21 </html>
```

(the above code snippet online)

If you save the above code and reload the page, you will see this:



All 4 Fields One Next to the Other

Don't worry about the style. All the labels and inputs have been put one next to the other. This is because both the `label` and the `input` HTML tags are inline elements. In fact the `label` tags are `inline` elements and the `input` tags are `inline-block` elements. We will later on introduce CSS to our page in order to make it look much better than it does now.

`select`

The next input that we want to ask user to key in, is their country. For the country, we do not allow the user to type in anything that he likes, but we want him to choose one from a specific list of values. We call this UI element a `select` box, a `single-select` box. You may hear about that as `drop down select box` or `drop down select list` or `drop down list` too.

One input HTML tag that we can use to accomplish this is the `select` HTML tag.

The `select` tag is a container of options. The options need to be given using the `option` tag.

Here is an example of a select box created using `select` and `option` tags. It contains 2 options for the user to choose.

```
1 <select>
2   <option>Male</option>
3   <option>Female</option>
4 </select>
```

(the above code snippet online)

You can clearly see the `select` element starting with the opening tag `<select>` and ending with the closing tag `</select>`. Inside it, its content is another HTML fragment, which is composed of a series of `option` elements. Each `option` element has HTML content, which are the options presented to the user in order for them to choose what they prefer.

On our address page example, we want to list all the countries in alphabetical order. Let's do that:

Here is how our HTML page content turns to be, after we add all the countries as options to the select box. Note that we also use a label to help user understand what this list is about.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>A basic input example</title>
6   </head>
7
8   <body>
9     <label>First Name:</label>
10    <input type="text"/>
11
12    <label>Last Name:</label>
13    <input type="text">
14
15    <label>Address:</label>
16    <input type="text">
17
18    <label>Zip Code:</label>
19    <input type="text">
20
21    <label>Country:</label>
22    <select>
23      <option>Afghanistan</option>
24      <option>Åland Islands</option>
25      <option>Albania</option>
```

```
26      <option>Algeria</option>
27      <option>American Samoa</option>
28      <option>Andorra</option>
29      <option>Angola</option>
30      <option>Anguilla</option>
31      <option>Antarctica</option>
32      <option>Antigua and Barbuda</option>
33      <option>Argentina</option>
34      <option>Armenia</option>
35      <option>Aruba</option>
36      <option>Australia</option>
37      <option>Austria</option>
38      <option>Azerbaijan</option>
39      <option>Bahamas</option>
40      <option>Bahrain</option>
41      <option>Bangladesh</option>
42      <option>Barbados</option>
43      <option>Belarus</option>
44      <option>Belgium</option>
45      <option>Belize</option>
46      <option>Benin</option>
47      <option>Bermuda</option>
48      <option>Bhutan</option>
49      <option>Bolivia, Plurinational State of</option>
50      <option>Bonaire, Sint Eustatius and Saba</option>
51      <option>Bosnia and Herzegovina</option>
52      <option>Botswana</option>
53      <option>Bouvet Island</option>
54      <option>Brazil</option>
55      <option>British Indian Ocean Territory</option>
56      <option>Brunei Darussalam</option>
57      <option>Bulgaria</option>
58      <option>Burkina Faso</option>
59      <option>Burundi</option>
60      <option>Cambodia</option>
61      <option>Cameroon</option>
62      <option>Canada</option>
63      <option>Cape Verde</option>
64      <option>Cayman Islands</option>
65      <option>Central African Republic</option>
66      <option>Chad</option>
67      <option>Chile</option>
68      <option>China</option>
69      <option>Christmas Island</option>
70      <option>Cocos (Keeling) Islands</option>
71      <option>Colombia</option>
```

```
72      <option>Comoros</option>
73      <option>Congo</option>
74      <option>Congo, the Democratic Republic of the</option>
75      <option>Cook Islands</option>
76      <option>Costa Rica</option>
77      <option>Côte d'Ivoire</option>
78      <option>Croatia</option>
79      <option>Cuba</option>
80      <option>Curaçao</option>
81      <option>Cyprus</option>
82      <option>Czech Republic</option>
83      <option>Denmark</option>
84      <option>Djibouti</option>
85      <option>Dominica</option>
86      <option>Dominican Republic</option>
87      <option>Ecuador</option>
88      <option>Egypt</option>
89      <option>El Salvador</option>
90      <option>Equatorial Guinea</option>
91      <option>Eritrea</option>
92      <option>Estonia</option>
93      <option>Ethiopia</option>
94      <option>Falkland Islands (Malvinas)</option>
95      <option>Faroe Islands</option>
96      <option>Fiji</option>
97      <option>Finland</option>
98      <option>France</option>
99      <option>French Guiana</option>
100     <option>French Polynesia</option>
101     <option>French Southern Territories</option>
102     <option>Gabon</option>
103     <option>Gambia</option>
104     <option>Georgia</option>
105     <option>Germany</option>
106     <option>Ghana</option>
107     <option>Gibraltar</option>
108     <option>Greece</option>
109     <option>Greenland</option>
110     <option>Grenada</option>
111     <option>Guadeloupe</option>
112     <option>Guam</option>
113     <option>Guatemala</option>
114     <option>Guernsey</option>
115     <option>Guinea</option>
116     <option>Guinea-Bissau</option>
117     <option>Guyana</option>
```

```
118      <option>Haiti</option>
119      <option>Heard Island and McDonald Islands</option>
120      <option>Holy See (Vatican City State)</option>
121      <option>Honduras</option>
122      <option>Hong Kong</option>
123      <option>Hungary</option>
124      <option>Iceland</option>
125      <option>India</option>
126      <option>Indonesia</option>
127      <option>Iran, Islamic Republic of</option>
128      <option>Iraq</option>
129      <option>Ireland</option>
130      <option>Isle of Man</option>
131      <option>Israel</option>
132      <option>Italy</option>
133      <option>Jamaica</option>
134      <option>Japan</option>
135      <option>Jersey</option>
136      <option>Jordan</option>
137      <option>Kazakhstan</option>
138      <option>Kenya</option>
139      <option>Kiribati</option>
140      <option>Korea, Democratic People's Republic of</option>
141      <option>Korea, Republic of</option>
142      <option>Kuwait</option>
143      <option>Kyrgyzstan</option>
144      <option>Lao People's Democratic Republic</option>
145      <option>Latvia</option>
146      <option>Lebanon</option>
147      <option>Lesotho</option>
148      <option>Liberia</option>
149      <option>Libya</option>
150      <option>Liechtenstein</option>
151      <option>Lithuania</option>
152      <option>Luxembourg</option>
153      <option>Macao</option>
154      <option>Macedonia, the former Yugoslav Republic of</option>
155      <option>Madagascar</option>
156      <option>Malawi</option>
157      <option>Malaysia</option>
158      <option>Maldives</option>
159      <option>Mali</option>
160      <option>Malta</option>
161      <option>Marshall Islands</option>
162      <option>Martinique</option>
163      <option>Mauritania</option>
```

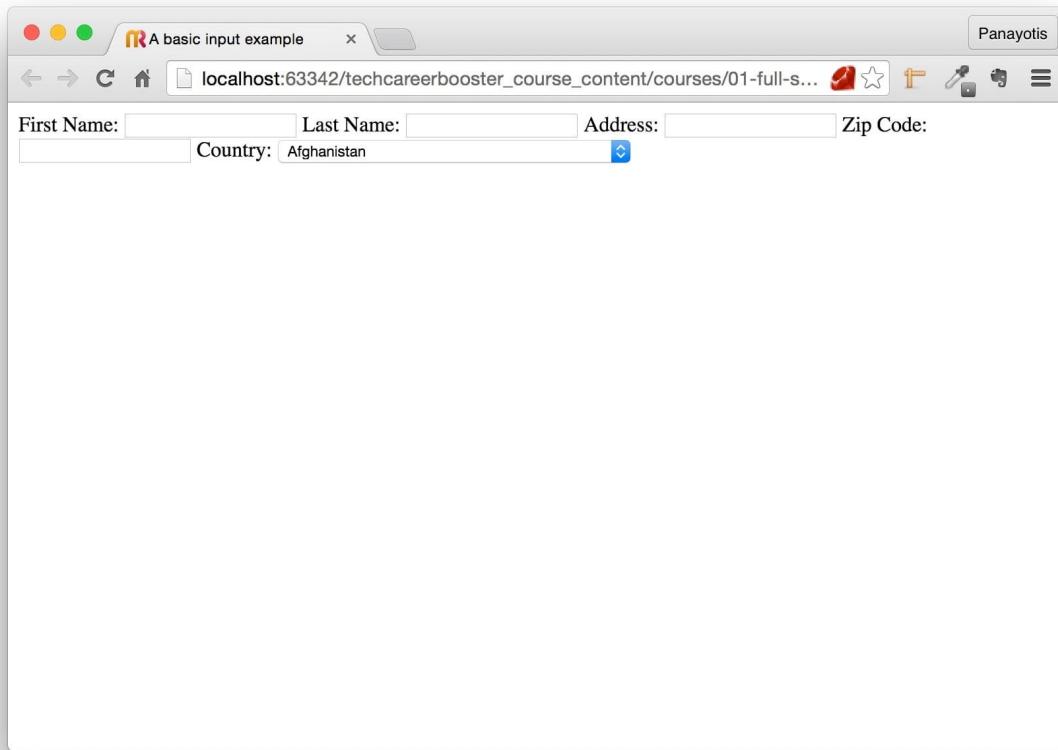
```
164      <option>Mauritius</option>
165      <option>Mayotte</option>
166      <option>Mexico</option>
167      <option>Micronesia, Federated States of</option>
168      <option>Moldova, Republic of</option>
169      <option>Monaco</option>
170      <option>Mongolia</option>
171      <option>Montenegro</option>
172      <option>Montserrat</option>
173      <option>Morocco</option>
174      <option>Mozambique</option>
175      <option>Myanmar</option>
176      <option>Namibia</option>
177      <option>Nauru</option>
178      <option>Nepal</option>
179      <option>Netherlands</option>
180      <option>New Caledonia</option>
181      <option>New Zealand</option>
182      <option>Nicaragua</option>
183      <option>Niger</option>
184      <option>Nigeria</option>
185      <option>Niue</option>
186      <option>Norfolk Island</option>
187      <option>Northern Mariana Islands</option>
188      <option>Norway</option>
189      <option>Oman</option>
190      <option>Pakistan</option>
191      <option>Palau</option>
192      <option>Palestinian Territory, Occupied</option>
193      <option>Panama</option>
194      <option>Papua New Guinea</option>
195      <option>Paraguay</option>
196      <option>Peru</option>
197      <option>Philippines</option>
198      <option>Pitcairn</option>
199      <option>Poland</option>
200      <option>Portugal</option>
201      <option>Puerto Rico</option>
202      <option>Qatar</option>
203      <option>Réunion</option>
204      <option>Romania</option>
205      <option>Russian Federation</option>
206      <option>Rwanda</option>
207      <option>Saint Barthélemy</option>
208      <option>Saint Helena, Ascension and Tristan da Cunha</option>
209      <option>Saint Kitts and Nevis</option>
```

```
210      <option>Saint Lucia</option>
211      <option>Saint Martin (French part)</option>
212      <option>Saint Pierre and Miquelon</option>
213      <option>Saint Vincent and the Grenadines</option>
214      <option>Samoa</option>
215      <option>San Marino</option>
216      <option>Sao Tome and Principe</option>
217      <option>Saudi Arabia</option>
218      <option>Senegal</option>
219      <option>Serbia</option>
220      <option>Seychelles</option>
221      <option>Sierra Leone</option>
222      <option>Singapore</option>
223      <option>Sint Maarten (Dutch part)</option>
224      <option>Slovakia</option>
225      <option>Slovenia</option>
226      <option>Solomon Islands</option>
227      <option>Somalia</option>
228      <option>South Africa</option>
229      <option>South Georgia and the South Sandwich Islands</option>
230      <option>South Sudan</option>
231      <option>Spain</option>
232      <option>Sri Lanka</option>
233      <option>Sudan</option>
234      <option>Suriname</option>
235      <option>Svalbard and Jan Mayen</option>
236      <option>Swaziland</option>
237      <option>Sweden</option>
238      <option>Switzerland</option>
239      <option>Syrian Arab Republic</option>
240      <option>Taiwan, Province of China</option>
241      <option>Tajikistan</option>
242      <option>Tanzania, United Republic of</option>
243      <option>Thailand</option>
244      <option>Timor-Leste</option>
245      <option>Togo</option>
246      <option>Tokelau</option>
247      <option>Tonga</option>
248      <option>Trinidad and Tobago</option>
249      <option>Tunisia</option>
250      <option>Turkey</option>
251      <option>Turkmenistan</option>
252      <option>Turks and Caicos Islands</option>
253      <option>Tuvalu</option>
254      <option>Uganda</option>
255      <option>Ukraine</option>
```

```
256      <option>United Arab Emirates</option>
257      <option>United Kingdom</option>
258      <option>United States</option>
259      <option>United States Minor Outlying Islands</option>
260      <option>Uruguay</option>
261      <option>Uzbekistan</option>
262      <option>Vanuatu</option>
263      <option>Venezuela, Bolivarian Republic of</option>
264      <option>Viet Nam</option>
265      <option>Virgin Islands, British</option>
266      <option>Virgin Islands, U.S.</option>
267      <option>Wallis and Futuna</option>
268      <option>Western Sahara</option>
269      <option>Yemen</option>
270      <option>Zambia</option>
271      <option>Zimbabwe</option>
272      </select>
273  </body>
274 </html>
```

(the above code snippet online)

If you save the above HTML content and reload your page, you will see this:



We have Introduced the Country Select Box

input of type=submit

What is missing from our form is a way for the user to tell the browser, the Web page, that they have finished filling in the form and that they are ready to submit the data to the Web page, to the Web application or Web site. In order to do that we need a UI control like a button:

A large, rectangular red button with the word "Submit" written in a bold, white, sans-serif font in the center. The button has a slight shadow effect at the bottom.

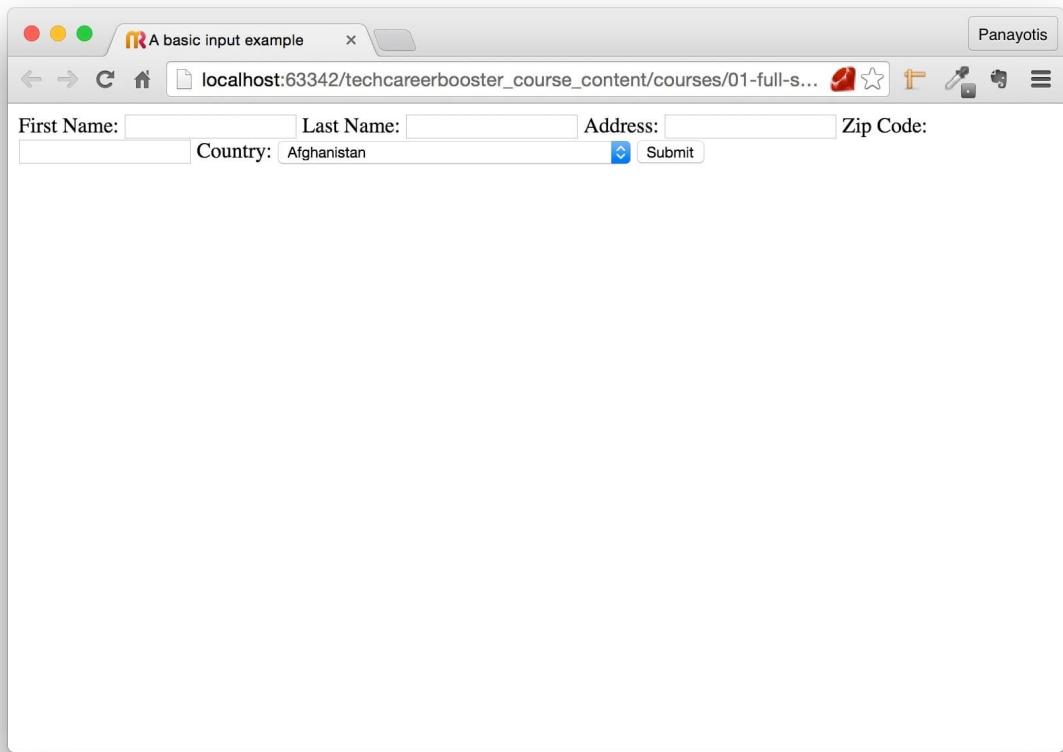
A Submit Button

Such buttons are created, again, with the HTML element `input`, but the `type` attribute needs to have the value `submit`. Try to add the following after the end of the select box for countries:

```
1  <input type="submit"/>
```

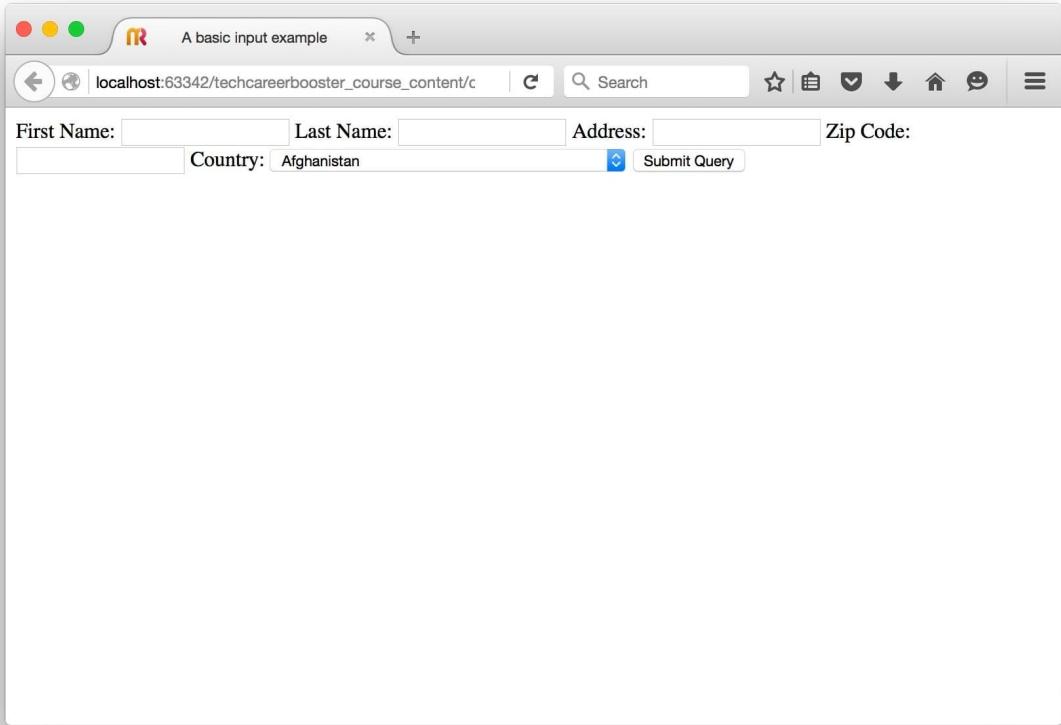
(the above code snippet online)

If you save and load the page you will see this:



With the Submit Input - Chrome Version

This is the Chrome version. If you load the same page on Firefox (43.0.2), you will see this:



With Submit Input - Firefox Version

As you can see above, the label on top of the button is different when the page is loaded on Firefox. And, if you try other browser brand, you may be presented with a different label.

In order to make sure that the same label is printed on top of the button, irrespective of the browser brand, you need to use the attribute `value` on the `input` element and give it the value equal to the label text that you want to use. Hence, one correct option for our `submit` type `input` is the following:

```
1 <input type="submit" value="Submit"/>
```

(the above code snippet online)

If you do this, then, even on Firefox, the button will have the label `Submit`.

Hence, our HTML page currently has the following content:

Note: In order to keep the HTML source code below, as small as possible, we will not repeat here the whole list of the countries. Your actual page needs to have all the countries in the list.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6      </head>
7
8      <body>
9          <label>First Name:</label>
10         <input type="text"/>
11
12         <label>Last Name:</label>
13         <input type="text">
14
15         <label>Address:</label>
16         <input type="text">
17
18         <label>Zip Code:</label>
19         <input type="text">
20
21         <label>Country:</label>
22         <select>
23             <option>Afghanistan</option>
24             <!-- rest of the countries go here -->
25             <option>Zimbabwe</option>
26         </select>
27
28         <input type="submit" value="Submit"/>
29
30     </body>
31 </html>
```

(the above code snippet online)

Again. Dont' worry about the styling of the page. We will fix that now. In order to improve our styling, we will wrap every label and input pair into a div and we will assign on each group the class `input-group`. The fact that we are using divs, will make each label and input pair to be on their own line. Here is the new HTML content:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <div class="input-group">
11             <label>First Name:</label>
12             <input type="text"/>
13         </div>
14
15         <div class="input-group">
16             <label>Last Name:</label>
17             <input type="text"/>
18         </div>
19
20         <div class="input-group">
21             <label>Address:</label>
22             <input type="text"/>
23         </div>
24
25         <div class="input-group">
26             <label>Zip Code:</label>
27             <input type="text"/>
28         </div>
29
30         <div class="input-group">
31             <label>Country:</label>
32             <select>
33                 <option>Afghanistan</option>
34                 <!-- rest of the countries go here -->
35                 <option>Zimbabwe</option>
36             </select>
37         </div>
38
39         <div class="input-group">
40             <input type="submit" value="Submit"/>
41         </div>
42
43     </body>
44 </html>
```

(the above code snippet online)

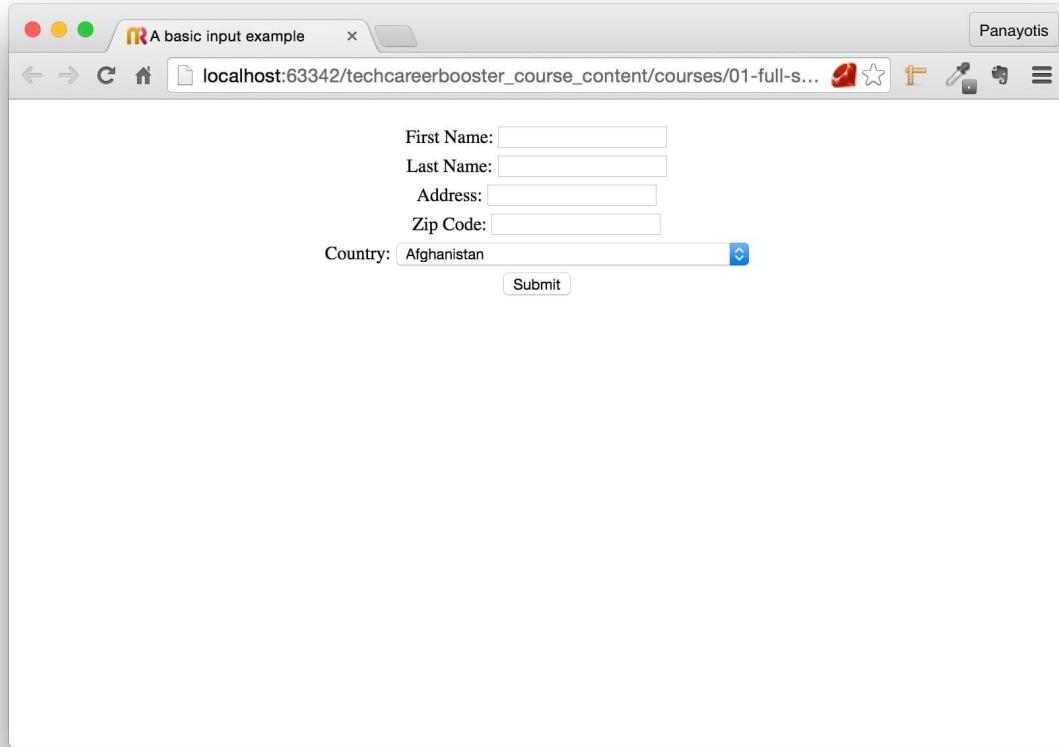
We also need to define the CSS rules for the class `input-group`. Let's create a new file with CSS rules and make sure that our HTML page references it.

Here is the content of our CSS file:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 10px;
14     text-align: center;
15 }
16
17 .input-group {
18     font-size: 1.4rem;
19     margin-bottom: 5px;
20 }
```

(the above code snippet online)

If you save both files and reload the page, you will see this:



Basic Styling on Address Details Form

Things are getting better. We will leave it like that, for the time being. We will improve later.

Although the form seems to be complete and allows any user to key in their address details, in fact, there are some required properties that are missing. These are necessary in order for the form to function as expected.

Here we go:

form

All the input elements that we use to collect information from the user need to be grouped into an HTML `form` element. The `form` element is a block level element. Only the input elements that are inside the `<form>` tag are actually used as useful data collected by the user.

Hence, the HTML code needs to be like that:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <form>
11             <div class="input-group">
12                 <label>First Name:</label>
13                 <input type="text"/>
14             </div>
15
16             <div class="input-group">
17                 <label>Last Name:</label>
18                 <input type="text"/>
19             </div>
20
21             <div class="input-group">
22                 <label>Address:</label>
23                 <input type="text"/>
24             </div>
25
26             <div class="input-group">
27                 <label>Zip Code:</label>
28                 <input type="text"/>
29             </div>
30
31             <div class="input-group">
32                 <label>Country:</label>
33                 <select>
34                     <option>Afghanistan</option>
35                     <!-- rest of the countries go here -->
36                     <option>Zimbabwe</option>
37                 </select>
38             </div>
39
40             <div class="input-group">
41                 <input type="submit" value="Submit"/>
42             </div>
43
44         </form>
45     </body>
46 </html>
```

(the above code snippet online)

Input elements' ID

It is a very good practice that all the input elements have a distinct id value. The id values are useful to link input controls to their corresponding label too. More about this later on.

Hence, let's give a unique id on all of our input elements:

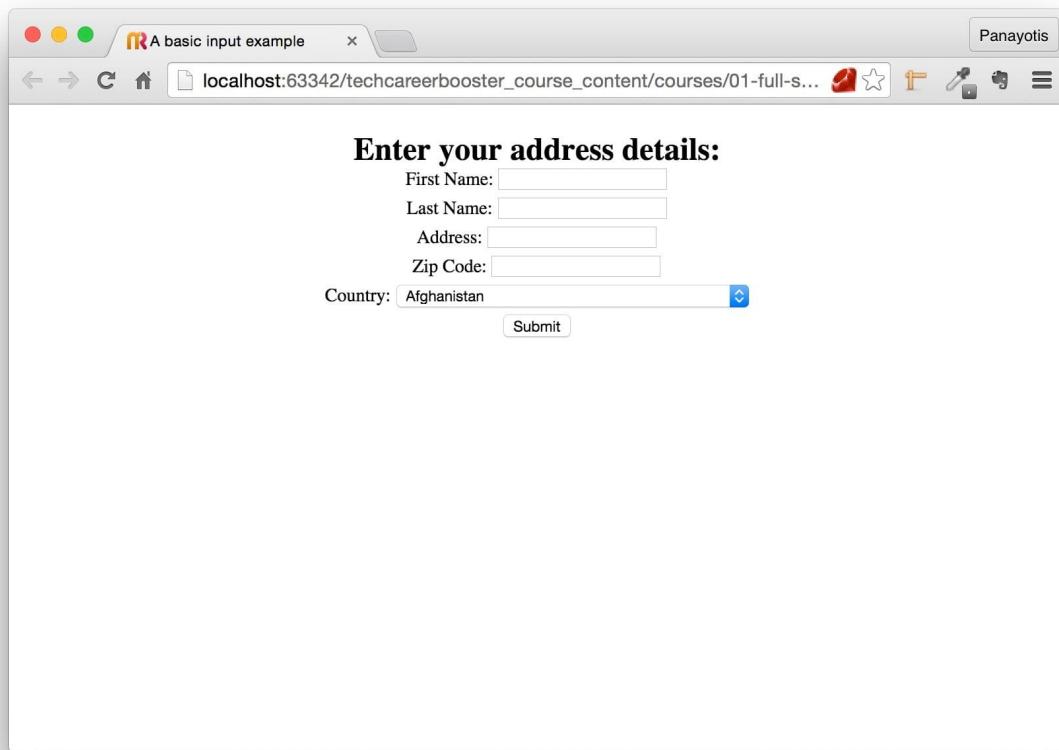
```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Enter your address details:</h2>
11         <form>
12             <div class="input-group">
13                 <label>First Name:</label>
14                 <input type="text" id="first-name"/>
15             </div>
16
17             <div class="input-group">
18                 <label>Last Name:</label>
19                 <input type="text" id="last-name"/>
20             </div>
21
22             <div class="input-group">
23                 <label>Address:</label>
24                 <input type="text" id="address"/>
25             </div>
26
27             <div class="input-group">
28                 <label>Zip Code:</label>
29                 <input type="text" id="zip-code"/>
30             </div>
31
32             <div class="input-group last">
33                 <label>Country:</label>
34                 <select id="country">
35                     <option>Afghanistan</option>
36                     <!-- rest of countries go here -->
37                     <option>Zimbabwe</option>
```

```
38      </select>
39  </div>
40
41  <div class="input-group">
42    <input type="submit" value="Submit" id="address-details-form-submit-button\
43 n"/>
44  </div>
45
46  </form>
47  </body>
48 </html>
```

(the above code snippet online)

Note that we have added the heading on top of the form.

Let's save the above content and load the page on our browser. You will see the following:



Form with Heading

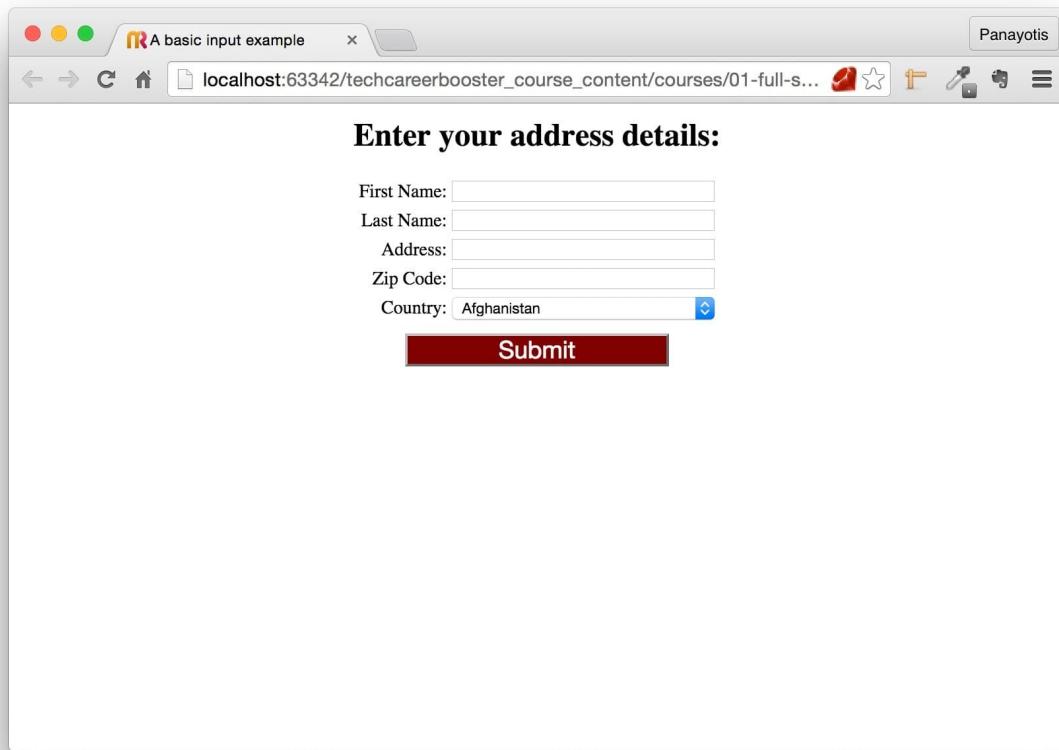
It's about time to apply some more styling. Change the CSS file content to be as follows:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 10px;
14     text-align: center;
15 }
16
17 h2 {
18     margin-bottom: 20px;
19 }
20
21 form {
22     width: 350px;
23     margin: 0 auto;
24 }
25
26 .input-group {
27     font-size: 1.4rem;
28     margin-bottom: 5px;
29 }
30
31 input, select {
32     width: 200px;
33 }
34
35 label {
36     display: inline-block;
37     text-align: right;
38     width: 67px;
39 }
40
41 .input-group.last {
42     margin-bottom: 10px;
43 }
44
45 form input[type='submit'] {
46     font-size: 1.8rem;
```

```
47     background-color: Maroon;
48     color: White;
49 }
50
51 form input[type='submit']:hover {
52     background-color: Black;
53     cursor: pointer;
54 }
```

(the above code snippet online)

Save the file and reload the page. You will see this:



Form with its final Styling

You can also move your mouse over the button. You will see how it changes color.

Let me draw your attention on some of the CSS selectors and rules that we have used:

- 1) Labels. They are inline elements. But we turn them to `inline-block` elements so that we can assign them specific width and text align the text inside

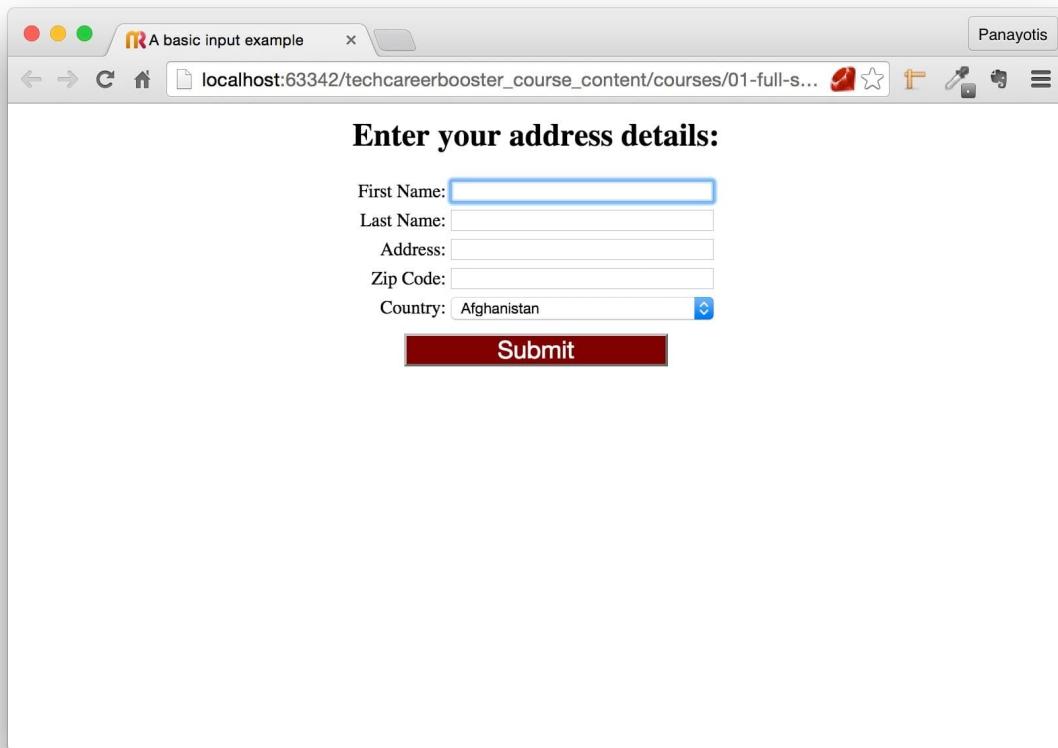
```
1  label {  
2      display: inline-block;  
3      text-align: right;  
4      width: 67px;  
5  }
```

2) The input group for the select box has some increased bottom margin so that we make the distance to button larger.

3) Selecting elements by attribute values. You can see that we are selecting the form submit button, by selecting all the input elements that belong to the form and that they have type attribute with value submit. `form input[type='submit']` And similar selector for the mouse over case (`:hover`): `form input[type='submit']:hover` > *Note:* We could have used the submit button id as a selector (`#address-details-form-submit-button`), but we preferred to use those above now for educational reasons.

autofocus

We have not finished yet with our form. There are some things that we can do to improve its functionality and the way it interacts with the user. For example, when the page loads, we want to make sure that the cursor is ready to accept input on the first name input element. Like that:



Form ready to accept input on the first name

In order to achieve that, we need to use the attribute `autofocus` on the first name input element. Let's do that:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Enter your address details:</h2>
11         <form>
12             <div class="input-group">
13                 <label>First Name:</label>
14                 <input type="text" id="first-name" autofocus="autofocus"/>
15             </div>
16
17             <div class="input-group">
18                 <label>Last Name:</label>
19                 <input type="text" id="last-name">
20             </div>
21
22             <div class="input-group">
23                 <label>Address:</label>
24                 <input type="text" id="address">
25             </div>
26
27             <div class="input-group">
28                 <label>Zip Code:</label>
29                 <input type="text" id="zip-code">
30             </div>
31
32             <div class="input-group last">
33                 <label>Country:</label>
34                 <select id="country">
35                     <option>Afghanistan</option>
36                     <!-- rest of countries go here -->
37                     <option>Zimbabwe</option>
38                 </select>
39             </div>
40
41             <div class="input-group">
42                 <input type="submit" value="Submit" id="address-details-form-submit-butto\
43 n"/>
```

```

44      </div>
45
46      </form>
47  </body>
48 </html>
```

(the above code snippet online)

Save the above file and reload your page. Do you see that the cursor is blinking inside the first name text input element? This is thanks to the `autofocus` attribute set on the first name input element.

Note: `autofocus` is one of the examples of attributes that do not have to have a value. They are called **boolean HTML element attributes**. This means that we can use the attribute without actually giving any value to it. The existence of the attribute switches corresponding property of the HTML tag on, renders it true, and the absence of it sets its value to its default value (true or false, depending on attribute). Like that:

```
1  <input type="text" id="first-name" autofocus/>
```

(the above code snippet online)

There are other attributes like this. You can either use it with a value (equal to the name of the attribute) like we did in the main HTML content, or you can use it only with its name. The presence of `autofocus` means that `autofocus` is switched on/true. The absence of `autofocus` means that `autofocus` is switched off/false.

label for attribute

One other important feature of the form is that allows the user to click on a label in order to set the input focus on the corresponding input control. Hence, if the user clicks on the label `Last Name:`, then the input focus moves automatically to the last name input element. In order for this feature to be enabled, we use the attribute `for` on each one of the labels and we set its value to be equal to the `id` value of the corresponding input element.

Here is the case for the first name:

```

1  <label for="first-name">First Name:</label>
2  <input type="text" id="first-name" autofocus="autofocus"/>
```

(the above code snippet online)

Do you see the `for` attribute on the `label` tag? It has the value `first-name` which is equal to the value of the `id` on the input element for the first name. This makes the two elements, `label` and `input` to be working together, with the functional feature that we have explained above. When user clicks on `First Name:` label, the focus moves to the input element for first name.

Let's do the label attribution with `for` on all labels:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Enter your address details:</h2>
11         <form>
12             <div class="input-group">
13                 <label for="first-name">First Name:</label>
14                 <input type="text" id="first-name" autofocus="autofocus"/>
15             </div>
16
17             <div class="input-group">
18                 <label for="last-name">Last Name:</label>
19                 <input type="text" id="last-name">
20             </div>
21
22             <div class="input-group">
23                 <label for="address">Address:</label>
24                 <input type="text" id="address">
25             </div>
26
27             <div class="input-group">
28                 <label for="zip-code">Zip Code:</label>
29                 <input type="text" id="zip-code">
30             </div>
31
32             <div class="input-group last">
33                 <label for="country">Country:</label>
34                 <select id="country">
35                     <option>Afghanistan</option>
36                     <!-- rest of countries go here -->
37                     <option>Zimbabwe</option>
38                 </select>
39             </div>
40
41             <div class="input-group">
42                 <input type="submit" value="Submit" id="address-details-form-submit-button"/>
43             </div>
44         </form>
45
46     </body>
```

```
47    </body>
48 </html>
```

(the above code snippet online)

If you save the above and reload your page, then try to click on the labels. Do you see the focus changing position according to which label you click?

Here is also a video that demonstrates that:

[Clicking on label moves focus on corresponding input](#)

required

Some times, we need to make sure that the user does not forget to fill in some of the required information. For example, on our address details field, we might consider all the information important and required to be filled in. In that case, we can use the `required` attribute (without value or with value equal to its name - this is another example of a boolean HTML element attribute, like the `autofocus` we saw earlier).

Let's put the `required` attribute on all of our input elements (excluding button of course):

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A basic input example</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Enter your address details:</h2>
11         <form>
12             <div class="input-group">
13                 <label for="first-name">First Name:</label>
14                 <input type="text" id="first-name" autofocus="autofocus" required="required"/>
15             </div>
16
17             <div class="input-group">
18                 <label for="last-name">Last Name:</label>
19                 <input type="text" id="last-name" required="required">
20             </div>
21
22             <div class="input-group">
23                 <label for="address">Address:</label>
24                 <input type="text" id="address" required="required">
25             </div>
26
```

```
27
28     <div class="input-group">
29         <label for="zip-code">Zip Code:</label>
30         <input type="text" id="zip-code" required="required">
31     </div>
32
33     <div class="input-group last">
34         <label for="country">Country:</label>
35         <select id="country" required="required">
36             <option>Afghanistan</option>
37             <!-- rest of countries go here -->
38             <option>Zimbabwe</option>
39         </select>
40     </div>
41
42     <div class="input-group">
43         <input type="submit" value="Submit" id="address-details-form-submit-button">
44     </div>
45
46
47     </form>
48 </body>
49 </html>
```

(the above code snippet online)

If you save and load this page again, you will not see any difference until you hit the submit button with one (or more) of the fields not filled in.

See how this works on Chrome browser.

The screenshot shows a web browser window titled "A basic input example". The address bar indicates the URL is "localhost:63342/techcareerbooster_course_content/courses/01-full-s...". The page content is a form titled "Enter your address details:". It contains fields for First Name, Last Name, Address, Zip Code, and Country, with "Afghanistan" selected. A "Submit" button is present. A validation message "Please fill out this field." with an exclamation mark icon is displayed above the "Address" field, indicating it is a required field.

First Name is Flagged As Required

As you can see, when user is trying to submit the form without having filled in the first name, the form is not submitted and a warning balloon is popped up exactly below the input field.

Important: This functionality is not supported by all browsers. In JavaScript lessons you will learn how you can accomplish the same functionality using JavaScript and, eventually, having this feature working on all browsers.

Sign Up Form

We continue our journey to HTML forms with another example. A sign up form. We will build the following form:

The screenshot shows a 'Sign Up Form' window. At the top, there are standard OS X window controls (red, yellow, green buttons) and a title bar with the text 'Sign Up Form'. Below the title bar is a toolbar with various icons. The main content area is titled 'Sign Up'. It contains four text input fields labeled 'First Name:', 'Last Name:', 'Email (*):', and 'Password (*):'. Below these fields is a checkbox labeled 'Receive newsletter'. At the bottom is a large red button labeled 'Sign Up'.

A Sign Up Form

There are some properties to this form that worth noting here:

1. The labels for the required fields (email and password) have an extra (*) to indicate that the corresponding inputs are required.
2. If you submit the form without having a proper email set in the email field, you will see that there is a balloon prompting you to enter a correct email address.
3. If you start typing on the password box, you will see that it does not display the characters that you type. It displays black bullets instead.
4. There is a new UI control, the checkbox, which is used to ask the user their confirmation to receive a newsletter or not.

Let's see this form in action:

The Sign Up Form in Action

In order to implement the above form we will need the following:

For first name and last name the things that we need are already known:

```
1 <div class="input-group">
2   <label for="first-name">First Name:</label>
3   <input type="text" id="first-name" autofocus="autofocus"/>
4 </div>
5
6 <div class="input-group">
7   <label for="last-name">Last Name:</label>
8   <input type="text" id="last-name"/>
9 </div>
```

(the above code snippet online)

input with type email

The email case is exactly the same as the 2 previous ones, except from the fact that we are using the type attribute with value `email`. This gives the correct semantics to the input and also allows the browser to display a warning balloon when the actual value that we have typed in does not look to be an email address. The browser does a validation check on the value to make sure that the value includes the character @.

Hence, our email input will be:

```
1 <div class="input-group">
2   <label for="email">Email (*):</label>
3   <input type="email" id="email" required="required"/>
4 </div>
```

(the above code snippet online)

input with type password

The password case is also an `input` element, but it has its own value for the `type` attribute. The value is `password`. This asks the browser to make sure that it hides the actual input and it displays black bullets for each one of the characters typed in.

The password input will be:

```
1 <div class="input-group">
2   <label for="password">Password (*):</label>
3   <input type="password" id="password" required="required"/>
4 </div>
```

(the above code snippet online)

input with type checkbox

For the check box functionality, we are using the `input` tag with the `type` attribute set to `checkbox`. This creates a rectangle on which the user can click and mark it as checked.

Important: If you set the required attribute on a checkbox, then the user will be asked to check that box and switch it on. This might be the desired behaviour, but not on our example sign up form case for the newsletter.

Hence, the checkbox input will be:

```
1 <div class="input-group">
2   <input type="checkbox" id="newsletter"/>
3   <label for="newsletter">Receive newsletter</label>
4 </div>
```

(the above code snippet online)

Having said all the above, we can now list the whole HTML content for our Sign Up form:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Sign Up Form</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10    <h2>Sign Up</h2>
11    <form>
12      <div class="input-group">
13        <label for="first-name">First Name:</label>
14        <input type="text" id="first-name" autofocus="autofocus"/>
15      </div>
16
17      <div class="input-group">
18        <label for="last-name">Last Name:</label>
19        <input type="text" id="last-name"/>
20      </div>
21
22      <div class="input-group">
23        <label for="email">Email (*):</label>
24        <input type="email" id="email" required="required"/>
25      </div>
26
```

```
27      <div class="input-group">
28          <label for="password">Password (*):</label>
29          <input type="password" id="password" required="required"/>
30      </div>
31
32      <div class="input-group">
33          <input type="checkbox" id="newsletter"/>
34          <label for="newsletter">Receive newsletter</label>
35      </div>
36
37      <div class="input-group">
38          <input type="submit" value="Sign Up"/>
39      </div>
40  </form>
41
42  </body>
43
44 </html>
```

(the above code snippet online)

and the corresponding CSS file is:

```
1  * {
2      box-sizing: border-box;
3  }
4
5  html {
6      font-size: 62.5%;
7  }
8
9  body {
10     font-size: 1.6rem;
11     margin-top: 1.6em;
12     text-align: center;
13 }
14
15 .input-group {
16     margin-bottom: 1.0rem;
17 }
18
19 label {
20     display: inline-block;
21     width: 13rem;
22     text-align: right;
23 }
```

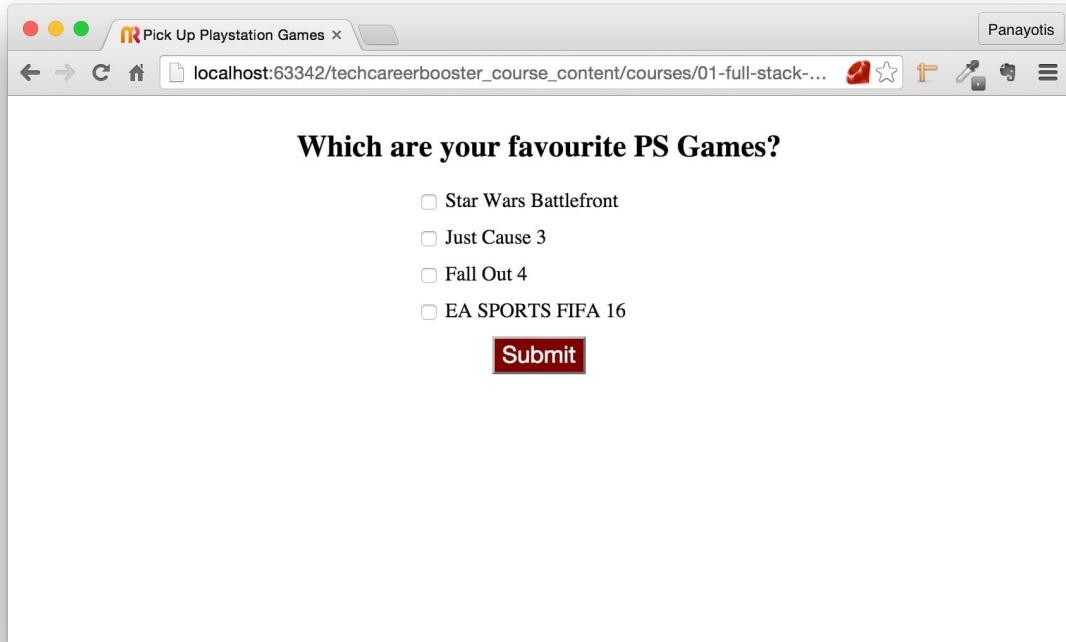
```
25 form input[type='submit'] {  
26   font-size: 1.8rem;  
27   background-color: Maroon;  
28   color: White;  
29 }  
30  
31 form input[type='submit']:hover {  
32   background-color: Black;  
33   cursor: pointer;  
34 }
```

(the above code snippet online)

If you save the above files and load the page on your browser, you will have the sign up form ready.

Multiple Checkboxes

The checkbox input is quite intuitive when we want to ask user an on / off choice, yes / no, true / false. Let's see another example. We want to ask the user to select one or more of the PlayStation games that he likes most.



Multiple Checkboxes

The user is allowed to pick up any number of options. This is a way to offer to user a multiple-select set of options.

Let's see the HTML mark up for that:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Pick Up Playstation Games</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Which are your favourite PS Games?</h2>
11         <form>
12             <div class="input-group">
13                 <input type="checkbox" id="star-wars-battlefront"/>
14                 <label for="star-wars-battlefront">Star Wars Battlefront</label>
15             </div>
16
17             <div class="input-group">
18                 <input type="checkbox" id="just-cause-3"/>
19                 <label for="just-cause-3">Just Cause 3</label>
20             </div>
21
22             <div class="input-group">
23                 <input type="checkbox" id="fall-out-4"/>
24                 <label for="fall-out-4">Fall Out 4</label>
25             </div>
26
27             <div class="input-group">
28                 <input type="checkbox" id="ea-sports-fifa-16"/>
29                 <label for="ea-sports-fifa-16">EA SPORTS FIFA 16</label>
30             </div>
31
32             <div class="input-group">
33                 <input type="submit" value="Submit"/>
34             </div>
35         </form>
36
37     </body>
38
39 </html>
```

(the above code snippet online)

This is pretty straightforward, if one remembers what we have learned about the checkboxes. Nothing new. The CSS that accompanies this HTML content is the following:

```
1 * {
2     box-sizing: border-box;
3 }
4
5 html {
6     font-size: 62.5%;
7 }
8
9 body {
10    font-size: 1.6rem;
11    margin-top: 1.6em;
12    text-align: center;
13 }
14
15 .input-group {
16    margin-bottom: 1.0rem;
17 }
18
19 label {
20    display: inline-block;
21    width: 170px;
22    text-align: left;
23 }
24
25 form input[type='submit'] {
26    font-size: 1.8rem;
27    background-color: Maroon;
28    color: White;
29 }
30
31 form input[type='submit']:hover {
32    background-color: Black;
33    cursor: pointer;
34 }
```

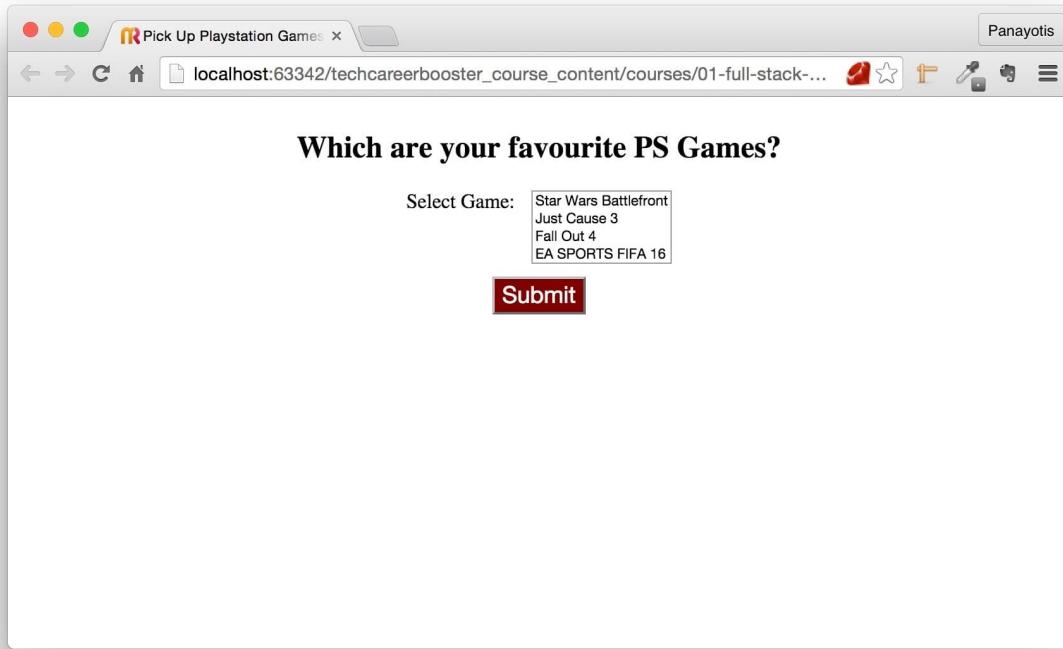
(the above code snippet online)

Nothing new here too.

Save the above two files and load the HTML page on your browser. You will finally have the page with the checkboxes for the user to pick up their favourite PS Games.

Multiple Select Box

A very popular alternative to multiple checkboxes is the multiple select box. It is constructed with the `select` HTML tag that has the attribute ‘multiple’. Let’s see that. We are going to build the above HTML page again, using a multiple select box. This is going to be the final result:



Multiple Select Box

As you can see above, the multiple select box presents to user more than 1 option at once. User is able to select more than 1. See how it works here:

This is how the multiple select box works

The HTML is pretty straight forward. The only new thing here is the `multiple="multiple"` attribute on the `select` element.

Note: The `multiple` is another example of a boolean HTML element attribute (like `autofocus` and `required`).

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Pick Up Playstation Games - Multi Select Box</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Which are your favourite PS Games?</h2>
11         <form>
12             <div class="input-group">
13                 <label for="star-wars-battlefront">Select Game:</label>
```

```
14     <select id="star-wars-battlefront" multiple="multiple">
15         <option>Star Wars Battlefront</option>
16         <option>Just Cause 3</option>
17         <option>Fall Out 4</option>
18         <option>EA SPORTS FIFA 16</option>
19     </select>
20 </div>
21
22 <div class="input-group">
23     <input type="submit" value="Submit"/>
24 </div>
25 </form>
26
27 </body>
28
29 </html>
```

(the above code snippet online)

In order to achieve the result presented on the video, the CSS that you should use is the following:

```
1 * {
2     box-sizing: border-box;
3 }
4
5 html {
6     font-size: 62.5%;
7 }
8
9 body {
10     font-size: 1.6rem;
11     margin-top: 1.6em;
12     text-align: center;
13 }
14
15 .input-group {
16     margin-bottom: 1.0rem;
17 }
18
19 label {
20     display: inline-block;
21     width: 95px;
22     text-align: left;
23     vertical-align: top;
24 }
25
26 form input[type='submit'] {
```

```
27     font-size: 1.8rem;
28     background-color: Maroon;
29     color: White;
30 }
31
32 form input[type='submit']:hover {
33     background-color: Black;
34     cursor: pointer;
35 }
```

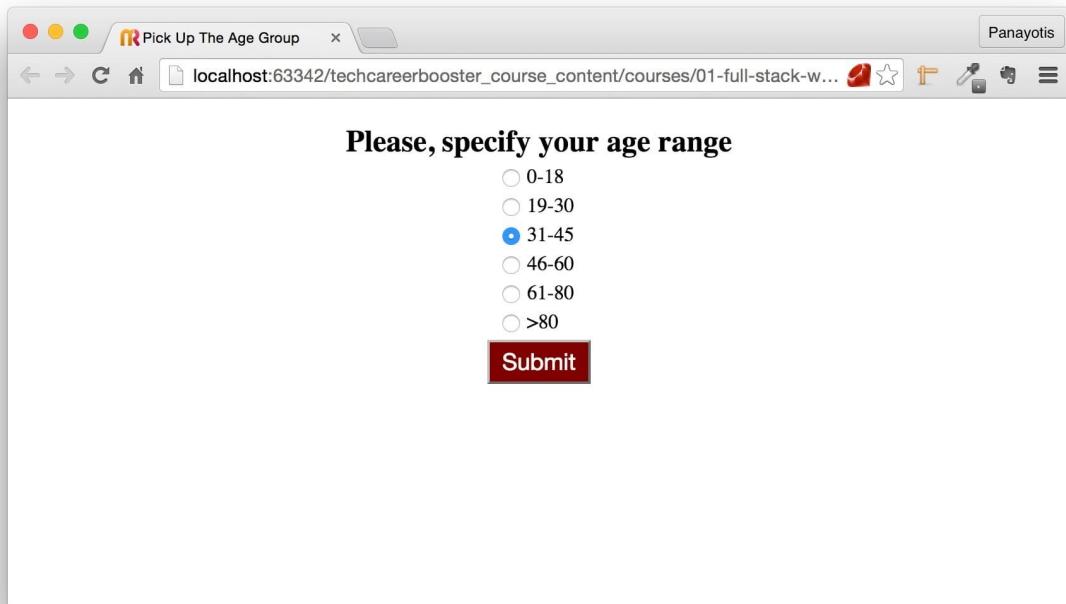
(the above code snippet online)

There is only one thing that you should note here. It is the `vertical-align: top;` property for the `label` tag. This will allow the label to appear at the top of its container, which is the `<div class="input-group">`. If you do not do that, it will be printed at the bottom, aligned with the bottom edge of the multiple select box.

input of type radio

Another way to present user with a list of choices and ask to select one, is using the `input` elements that have `type` equal to `radio`. The point here is that the radio buttons, as we call them, are used to allow the user to pick up only 1 option. So they function more like the way a single select box functions.

Let's see the following example:



Specify Age Range Using Radio Buttons

As you can see above, the user is presented with a list of options, each one having a radio button, which the user can click. If the user clicks on an option, other selected option is deselected.

You can see the user working with this form in the following video:

Selecting Age Range Using Radio Buttons

The radio buttons are very simple to implement. As we said, they are input tag elements that have the type set to radio. So, for our example page, the first age range can be specified as follows:

```
1 <div class="input-group">
2   <input type="radio" id="age-range-0-to-18">
3   <label for="age-range-0-to-18">0-18</label>
4 </div>
```

(the above code snippet online)

As you can see above, there is a corresponding label that is bound to the radio button. This is useful because, except from the fact that the label explains the option, so that the users know what they are selecting, it also gives the users the option to click on the label (and not only on the radio button itself) in order to select it.

Let's create our HTML page with all the necessary radio buttons to represent the form with the age ranges:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Pick Up The Age Group</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10    <h2>Please, specify your age range</h2>
11    <form>
12      <div class="input-group">
13        <input type="radio" id="age-range-0-to-18">
14        <label for="age-range-0-to-18">0-18</label>
15      </div>
16
17      <div class="input-group">
18        <input type="radio" id="age-range-19-to-30">
19        <label for="age-range-19-to-30">19-30</label>
20      </div>
21
22      <div class="input-group">
23        <input type="radio" id="age-range-31-to-45">
```

```
24      <label for="age-range-31-to-45">31-45</label>
25  </div>
26
27  <div class="input-group">
28    <input type="radio" id="age-range-46-to-60">
29    <label for="age-range-46-to-60">46-60</label>
30  </div>
31
32  <div class="input-group">
33    <input type="radio" id="age-range-61-to-80">
34    <label for="age-range-61-to-80">61-80</label>
35  </div>
36
37  <div class="input-group">
38    <input type="radio" id="age-range-above-80">
39    <label for="age-range-above-80">&gt;80</label>
40  </div>
41
42  <div class="input-group last">
43    <input type="submit" value="Submit"/>
44  </div>
45 </form>
46
47 </body>
48
49 </html>
```

(the above code snippet online)

And the CSS file, let it be like this:

```
1  * {
2    box-sizing: border-box;
3    padding: 0;
4    margin: 0;
5  }
6
7  html {
8    font-size: 62.5%;
9  }
10
11 body {
12   font-size: 1.6rem;
13   margin-top: 20px;
14   text-align: center;
15 }
16
```

```
17 h2 {
18     margin-bottom: 5px;
19 }
20
21 .input-group {
22     margin-bottom: 5px;
23 }
24
25 .input-group.last {
26     margin-bottom: 20px;
27 }
28
29 label {
30     display: inline-block;
31     width: 41px;
32     padding-left: 3px;
33     text-align: left;
34 }
35 form input[type='submit'] {
36     font-size: 1.8rem;
37     background-color: Maroon;
38     color: White;
39     padding: 5px 10px;
40 }
41
42 form input[type='submit']:hover {
43     background-color: Black;
44     cursor: pointer;
45 }
```

(the above code snippet online)

If you save those two files and reload your page, you will see the page displayed correctly, as expected.

But there is a problem. When you click on a radio button, it does not deselect the previously one selected. And allows all the options to be switched on. Which is not the requirement here and the correct way for the radio buttons to work. As we said earlier, only 1 radio button should be selected at any time (or none).

The problem occurs because we have not told the browser that all these radio buttons should function together as a group. If we had a way to tell that to the browser, then browser would have been smart enough to switch off a radio button when switch on another.

In order to tell to the browser, that a set of radio buttons belong to the same group, you have to use the attribute `name` and give it the same value for all `input` radio elements belonging to the same group. Let's try with our page. We will set the attribute `name` on all `input` radio elements and we will set it to be with the value `age-range`.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Pick Up The Age Group</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Please, specify your age range</h2>
11         <form>
12             <div class="input-group">
13                 <input type="radio" id="age-range-0-to-18" name="age-range">
14                 <label for="age-range-0-to-18">0-18</label>
15             </div>
16
17             <div class="input-group">
18                 <input type="radio" id="age-range-19-to-30" name="age-range">
19                 <label for="age-range-19-to-30">19-30</label>
20             </div>
21
22             <div class="input-group">
23                 <input type="radio" id="age-range-31-to-45" name="age-range">
24                 <label for="age-range-31-to-45">31-45</label>
25             </div>
26
27             <div class="input-group">
28                 <input type="radio" id="age-range-46-to-60" name="age-range">
29                 <label for="age-range-46-to-60">46-60</label>
30             </div>
31
32             <div class="input-group">
33                 <input type="radio" id="age-range-61-to-80" name="age-range">
34                 <label for="age-range-61-to-80">61-80</label>
35             </div>
36
37             <div class="input-group">
38                 <input type="radio" id="age-range-above-80" name="age-range">
39                 <label for="age-range-above-80">&gt;80</label>
40             </div>
41
42             <div class="input-group last">
43                 <input type="submit" value="Submit"/>
44             </div>
45         </form>
46
```

```
47    </body>
48
49 </html>
```

(the above code snippet online)

If you save the above code and reload your page, you will see that, now, the radio buttons work as expected. When the user switches on one option, the previously selected one is automatically switched off.

checked attribute

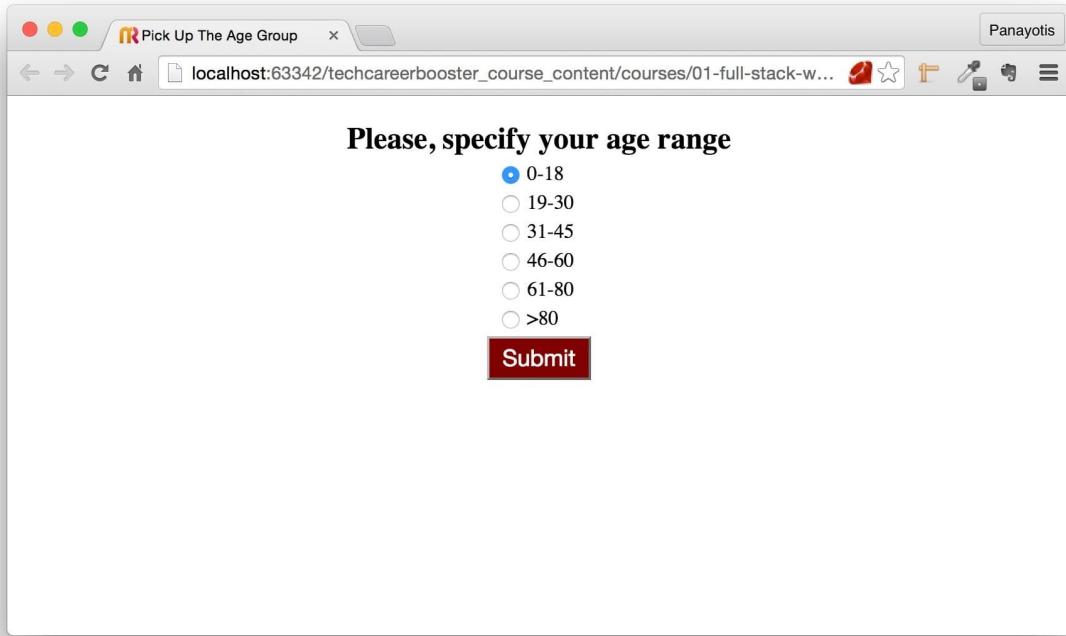
Both for checkboxes and for radio buttons, you may want an option to be switched on when the page first loads. And option can be switched on if it has the attribute `checked` (this is a boolean HTML element attribute). Try the following HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Pick Up The Age Group</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Please, specify your age range</h2>
11         <form>
12             <div class="input-group">
13                 <input type="radio" id="age-range-0-to-18" name="age-range" checked="checked">
14                 <label for="age-range-0-to-18">0-18</label>
15             </div>
16
17             <div class="input-group">
18                 <input type="radio" id="age-range-19-to-30" name="age-range">
19                 <label for="age-range-19-to-30">19-30</label>
20             </div>
21
22             <div class="input-group">
23                 <input type="radio" id="age-range-31-to-45" name="age-range">
24                 <label for="age-range-31-to-45">31-45</label>
25             </div>
26
27             <div class="input-group">
28                 <input type="radio" id="age-range-46-to-60" name="age-range">
29                 <label for="age-range-46-to-60">46-60</label>
30             </div>
```

```
31     </div>
32
33     <div class="input-group">
34         <input type="radio" id="age-range-61-to-80" name="age-range">
35         <label for="age-range-61-to-80">61-80</label>
36     </div>
37
38     <div class="input-group">
39         <input type="radio" id="age-range-above-80" name="age-range">
40         <label for="age-range-above-80">&gt;80</label>
41     </div>
42
43     <div class="input-group last">
44         <input type="submit" value="Submit"/>
45     </div>
46 </form>
47
48 </body>
49
50 </html>
```

(the above code snippet online)

Do you see the `checked="checked"` on the first radio button? If you save the above content and load the page on your browser, you will see that the first age range is preselected. Like that:

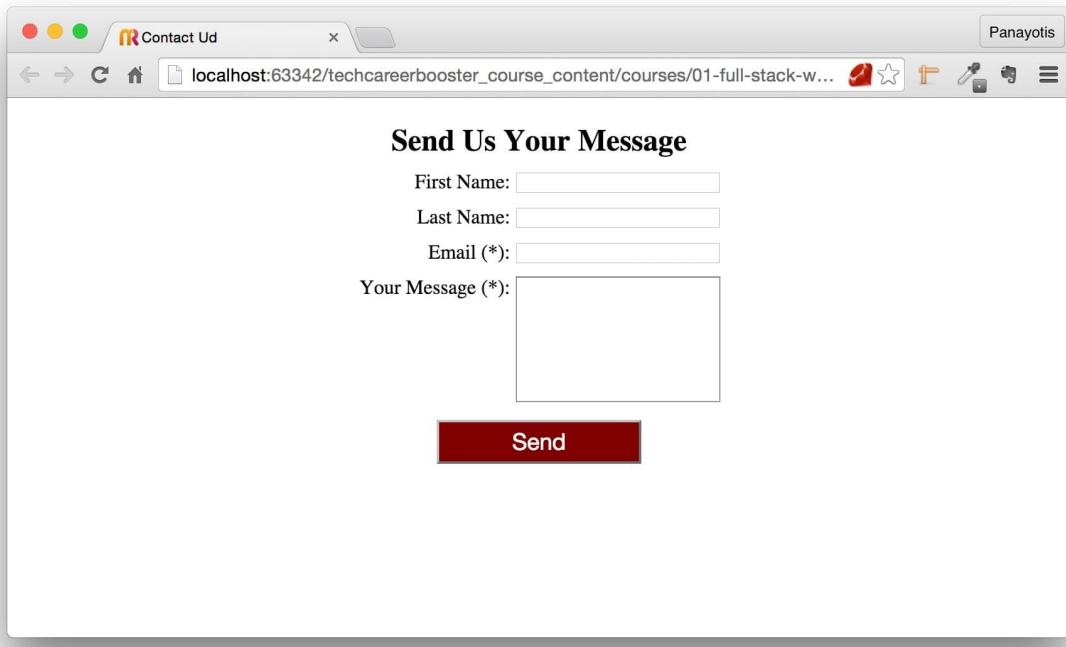


First Age Range is Preselected When Page Loads

The same technique, with `checked` attribute can bee applied to checkboxes too.

`textarea`

Let's move on to create a Contact Us form, in which we ask the user to type in their basic identity details and also give us a message that they would like to inform us about. Here it is how it would look like:



Contact Us Form

The properties of this form that we need to note here are:

1. First name and last name are not mandatory
2. The email is mandatory
3. The message is mandatory. Note that we do not use a one-line input control for the message.
We want the user to be able to write multiple lines in his message.

The key to allow multiple lines on the input control for the message is the HTML element `textarea`. The `textarea` is an HTML element that has an opening (`<textarea>`) and a closing (`</textarea>`) tag. When used, it displays a UI control on the page that allows the user to enter multiple lines.

Having said that, this is the HTML fragment that will allow us to present such a UI control for the message input:

```
1 <div class="input-group">
2   <label for="message">Your Message (*):</label>
3   <textarea id="message" required="required"></textarea>
4 </div>
```

(the above code snippet online)

So, `textarea`, as you can see is very simple.

Let's see the code of our HTML Contact Us form page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Contact Us</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Send Us Your Message</h2>
11         <form>
12             <div class="input-group">
13                 <label for="first-name">First Name:</label>
14                 <input type="text" id="first-name" autofocus="autofocus"/>
15             </div>
16
17             <div class="input-group">
18                 <label for="last-name">Last Name:</label>
19                 <input type="text" id="last-name"/>
20             </div>
21
22             <div class="input-group">
23                 <label for="email">Email (*):</label>
24                 <input type="email" id="email" required="required"/>
25             </div>
26
27             <div class="input-group">
28                 <label for="message">Your Message (*):</label>
29                 <textarea id="message" required="required"></textarea>
30             </div>
31
32             <div class="input-group">
33                 <input type="submit" value="Send"/>
34             </div>
35         </form>
36
37     </body>
38
39 </html>
```

(the above code snippet online)

You can use the following CSS file content too:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14     text-align: center;
15 }
16
17 h2 {
18     margin-bottom: 10px;
19 }
20
21 .input-group {
22     margin-bottom: 10px;
23 }
24
25 label {
26     display: inline-block;
27     width: 121px;
28     text-align: right;
29     vertical-align: top;
30 }
31
32 input, textarea {
33     width: 163px;
34 }
35
36 textarea {
37     height: 100px;
38     resize: none;
39 }
40
41 form input[type='submit'] {
42     font-size: 1.8rem;
43     background-color: Maroon;
44     color: White;
45     padding: 5px 10px;
46 }
```

```
47  
48 form input[type='submit']:hover {  
49     background-color: Black;  
50     cursor: pointer;  
51 }
```

(the above code snippet online)

If you save the above two files and load the HTML page on your browser, you will see the form functioning as expected.

Something that might worth noting here is the property `resize` that has been used on the `textarea` CSS rule. This now has been set to have the value `none`. This fixes the size of the `textarea` UI control presented to the user. If you want the user to be able to resize the area that he has available to type in his message, you can remove this property completely.

Finally, the height of the `textarea` is defined using the `height` CSS property. Another way you can define the height of a `textarea` is to use the attribute `rows` on the `textarea` element. It takes the number of lines/rows the user can type in, before start scrolling and a vertical scroll bar appears on the right hand side of the `textarea`. Hence:

```
1 <textarea rows="5"></textarea>
```

(the above code snippet online)

will render the `textarea` height such that the user is able to type in 5 lines of text and all being visible. The 6th line will make the text scroll and a vertical scroll bar will appear on the right.

Similar to `rows` attribute, one can define the width of a `textarea` using the attribute `cols`. However, this might not be that handy, because we usually prefer the width to be a relative size of the container of the `textarea`.

Other than that, there is nothing special with the CSS rules for this form.

placeholder attribute

Sometimes, you may have seen forms like this:

The screenshot shows a web browser window with the title bar 'Contact Ud'. The address bar displays 'localhost:63342/techcareerbooster_course_content/courses/01-full-stack-w...'. The main content area has a heading 'Send Us Your Message'. Below it are four input fields: 'First Name' (highlighted with a blue border), 'Last Name', 'Email (required)', and 'Your Message (required)'. A large red 'Send' button is at the bottom.

Contact Us Form Without Labels

This form above, does not use labels, but it has a text inside the input element that explains to the user what he has to type in. It works like a label. But, the behaviour here is that when the user starts typing in, the text inside the input disappears. The prompt text appears again only if the user deletes the words that he has typed in. Watch the following video to understand what we mean:

Contact Us Form Without Labels

This feature of the input controls is achieved with the attribute `placeholder`. It takes as value the text prompt that we want to appear inside the input control. Let's change our HTML contact us form to work with placeholders and without labels:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Contact Ud</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Send Us Your Message</h2>
11         <form>
12             <div class="input-group">
13                 <input type="text" id="first-name" autofocus="autofocus" placeholder="First Name"/>
```

```
15     </div>
16
17     <div class="input-group">
18         <input type="text" id="last-name" placeholder="Last Name"/>
19     </div>
20
21     <div class="input-group">
22         <input type="email" id="email" required="required" placeholder="Email (re\
23 quired)"/>
24     </div>
25
26     <div class="input-group">
27         <textarea id="message" required="required" placeholder="Your Message (req\
28 uired)"></textarea>
29     </div>
30
31     <div class="input-group">
32         <input type="submit" value="Send"/>
33     </div>
34 </form>
35
36 </body>
37
38 </html>
```

(the above code snippet online)

As you can see above, we have removed the labels and we have added the attribute `placeholder`. Save the above HTML code and reload the page. You will see the contact us form working as demonstrated in the video above.

Important: You need to take into account that, if the form is already filled in, and the input controls do not have labels, but only placeholders, then the user might not know which input control corresponds to what. Look at the following form:

The screenshot shows a web browser window with a title bar 'Contact Us' and a user 'Panayotis'. The address bar shows 'localhost:63342/techcareerbooster_course_content/courses/01-full-stack-w...'. The main content is a form titled 'Send Us Your Message'. It contains three input fields: 'John' (placeholder), 'Smith' (placeholder), and 'john_smith@gmail.com' (value). Below these is a text area with the following content:
Hi,
Can you please come back to
me on 677 7778777? I have a
problem with my invoice.
Thanks in advance,
John Smith
A red 'Send' button is at the bottom.

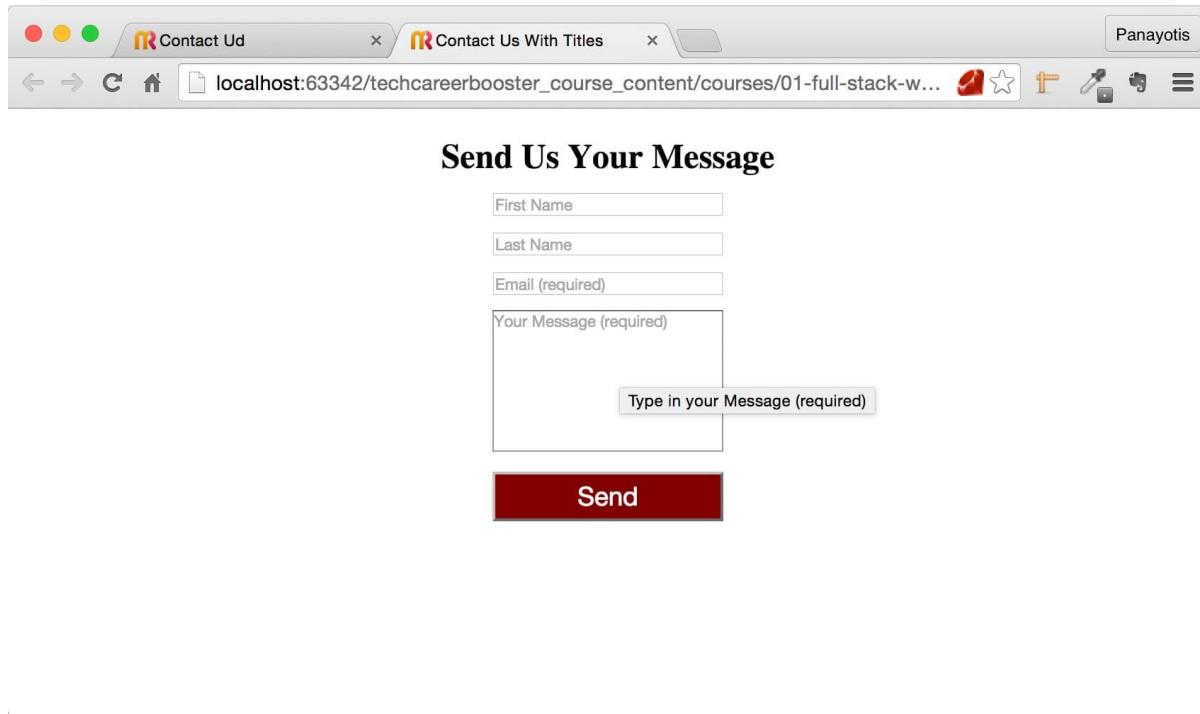
Contact Form without Labels Already Filled In

Can you tell which piece of information is being asked for the first input control? Is it the first name or the last name? Or something else? What about the third input control? Is it the email or the username? Which piece of information is required? The answers to these questions can be inferred only by looking at the values that the input controls already have. And, even in that case, we are not 100% sure about it. So, you need to think about it twice if you are to remove the labels from your input controls and rely only on the placeholders. One good technique that you might consider is to use both a label and a placeholder. The placeholder is usually more descriptive than the label giving some extra instructions about how the input information should be given by the user.

title

Another way we can help user fill in the information of an input control is by the use of the attribute **title**. The **title** attribute takes a text, a phrase, that is presented to the user when the user moves his mouse over the input control.

Look at the following example presented here:



Contact Us Form With Titles - Mouse over Message

On the above example, we are using titles and the screenshot is when the user moves his mouse over the message input control.

You can also watch this video to see how the mouse over works on this contact us form.

How the title attribute works on the contact us form

It's very simple to enhance our label-less form with titles. Let's do that:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Contact Us With Titles</title>
6          <link rel="stylesheet" href="stylesheets/title.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Send Us Your Message</h2>
11         <form>
12             <div class="input-group">
13                 <input type="text" id="first-name" autofocus="autofocus" placeholder="First Name" title="Type in your First Name"/>
14             </div>
15
16             <div class="input-group">
17                 <input type="text" id="last-name" placeholder="Last Name" title="Type in your Last Name"/>
18             </div>
19         </form>
```

```

20      </div>
21
22      <div class="input-group">
23          <input type="email" id="email" required="required" placeholder="Email (re\
quired)" title="Type in your Email (required)"/>
24      </div>
25
26
27      <div class="input-group">
28          <textarea id="message" required="required" placeholder="Your Message (req\
uired)" title="Type in your Message (required)"/></textarea>
29      </div>
30
31
32      <div class="input-group">
33          <input type="submit" value="Send"/>
34      </div>
35  </form>
36
37  </body>
38
39 </html>

```

(the above code snippet online)

Save the above HTML file and load it on your browser. You will see that the result is one presented on the previous video. If you move your mouse over the input controls, you can see the title popping up.

Note: The `title` attribute can be used on any visual HTML element. And has the same functionality. If the user moves his mouse over the HTML element, the `title` attribute value will pop up at that point, usually explaining to the user about the corresponding element.

autocomplete attribute

Browser is set up to remember the values that you have previously typed in on same / similar input controls. With this list of values in its memory, browser has the ability to suggest to you a list of values that you can choose from while filling in an input control. For example, if you have previously filled in a form that asks for the email with the value “john_smith@gmail.com”, browser will remember that, and next time you fill in the same form email input (or other form that asks for email) will suggest to you the value “john_smith@gmail.com” so that you can select it, instead of typing it end-to-end. This feature is called *autocomplete* and it is really helpful because it saves user from keystrokes.

You can see that feature in action here:

Autocomplete Feature

By default all the input controls have this feature on. If you want to switch it off, you have to use the attribute `autocomplete` with the value `off`. This attribute can be set to the `form` level. In

that case all the input controls of that form inherit the same level of autocompletion. You can also set it on individual input elements.

In the following example, we set autocompletion off for the whole form except for the email field, on which we switch it on (with the value `on`):

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Contact Us With Titles</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Send Us Your Message</h2>
11         <form autocomplete="off">
12             <div class="input-group">
13                 <input type="text" id="first-name" autofocus="autofocus" placeholder="First Name" title="Type in your First Name"/>
14             </div>
15
16             <div class="input-group">
17                 <input type="text" id="last-name" placeholder="Last Name" title="Type in your Last Name"/>
18             </div>
19
20             <div class="input-group">
21                 <input type="email" id="email" required="required" placeholder="Email (required)" title="Type in your Email (required)" autocomplete="on"/>
22             </div>
23
24             <div class="input-group">
25                 <textarea id="message" required="required" placeholder="Your Message (required)" title="Type in your Message (required)"/></textarea>
26             </div>
27
28             <div class="input-group">
29                 <input type="submit" value="Send"/>
30             </div>
31
32         </form>
33
34     </body>
35
36
37 </html>
```

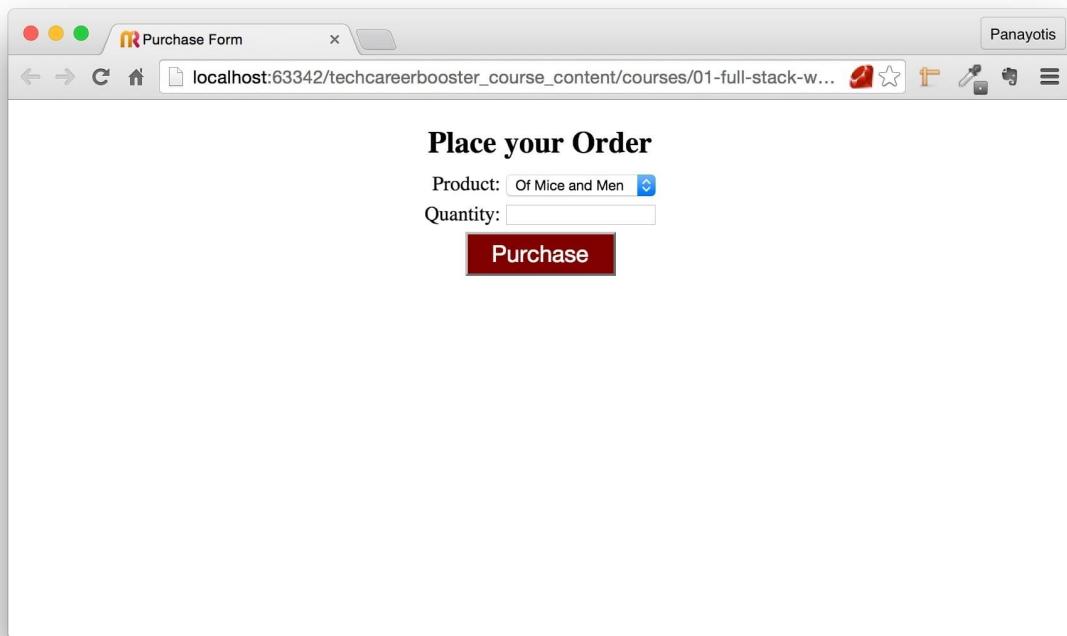
(the above code snippet online)

If you save and reload this page on your browser, when trying to fill in the first name and last name, you will not get any autocompletion help. You will only get it for the email input control.

Contact Form with Autocompletion Only on Email

input with type number

There are cases in which we want to ask the user to give us a number. For example, we have a checkout form and there is an input control for the buyer to key in the quantity of the product that he is purchasing. Look at the following form:



The screenshot shows a web browser window with the title "Purchase Form". The URL in the address bar is "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-w...". The main content of the page is a form titled "Place your Order". It contains a dropdown menu labeled "Product" with the option "Of Mice and Men" selected. Below it is a numeric input field labeled "Quantity" with the value "1". At the bottom of the form is a red rectangular button labeled "Purchase". The browser interface includes standard controls like back, forward, and search, along with a tab labeled "Panayotis" and various icons for sharing and saving.

Form with Quantity Being a Number

As you can see above, the quantity does not have any difference on how the input control is drawn. The difference will be obvious when you try to key in the quantity. You will see that the input control changes to have a small handle on the right that allows you to increase or decrease the number inside the box, by clicking on the corresponding edges of the handle. You will also experience the following:

1. You can use the up arrow key to increase the number inside the quantity field.
2. You can use the down arrow key to decrease the number inside the quantity field.
3. You can only type in numbers. If you try to type in a character that is not a number, that character will not be accepted.

Let's see that in action:

Using the Quantity Field

In order to achieve the above functionality, you need to use the type attribute with the value number. Let's see the HTML code of the above page.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Purchase Form</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Place your Order</h2>
11         <form>
12
13             <div class="input-group">
14                 <label for="product">Product:</label>
15                 <select id="product" autofocus="autofocus">
16                     <option>Of Mice and Men</option>
17                     <option>Game of Thrones</option>
18                     <option>Star Wars II</option>
19                 </select>
20             </div>
21
22             <div class="input-group">
23                 <label for="quantity">Quantity: </label>
24                 <input type="number" id="quantity" required="required"/>
25             </div>
26
27             <div class="input-group">
28                 <input type="submit" value="Purchase"/>
29             </div>
30
31         </form>
32     </body>
33
34 </html>
```

(the above code snippet online)

As you can see above, for the quantity field, we are using an `input` HTML element with value `number` on its `type` attribute. Other than that, there is not anything new.

The CSS of this page is:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14     text-align: center;
15 }
16
17 h2 {
18     margin-bottom: 10px;
19 }
20
21 .input-group {
22     margin-bottom: 5px;
23 }
24
25 .input-group.last {
26     margin-bottom: 10px;
27 }
28
29 label {
30     display: inline-block;
31     width: 61px;
32     text-align: right;
33 }
34
35 input, select {
36     width: 120px;
37 }
38
39 form input[type='submit'] {
40     font-size: 1.8rem;
41     background-color: Maroon;
42     color: White;
43     padding: 5px 10px;
44 }
45
46 form input[type='submit']:hover {
```

```

47     background-color: Black;
48     cursor: pointer;
49 }
```

(the above code snippet online)

Save the above files and load your page on your browser. You will see how the form behaves exactly as described above.

min and max attributes

There is a problem with the quantity field on the previous purchase form. Although the user is allowed to enter only numbers, the problem is that he can set any number that he likes. Even negative integers. We can use the `min` and `max` attributes and specify the minimum and maximum values that the user can enter respectively.

In the example of our purchase form, we would like to assign the value `1` to the attribute `min` and the value `10` to the attribute `max`. Hence, limiting the user to values between `1` and `10`. Let's do that:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Purchase Form</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Place your Order</h2>
11         <form>
12
13             <div class="input-group">
14                 <label for="product">Product:</label>
15                 <select id="product" autofocus="autofocus">
16                     <option>Of Mice and Men</option>
17                     <option>Game of Thrones</option>
18                     <option>Star Wars II</option>
19                 </select>
20             </div>
21
22             <div class="input-group">
23                 <label for="quantity">Quantity: </label>
24                 <input type="number" id="quantity" required="required" min="1" max="10"/>
25             </div>
26
27             <div class="input-group">
```

```
28      <input type="submit" value="Purchase"/>
29  </div>
30
31  </form>
32
33
34  </body>
35
36 </html>
```

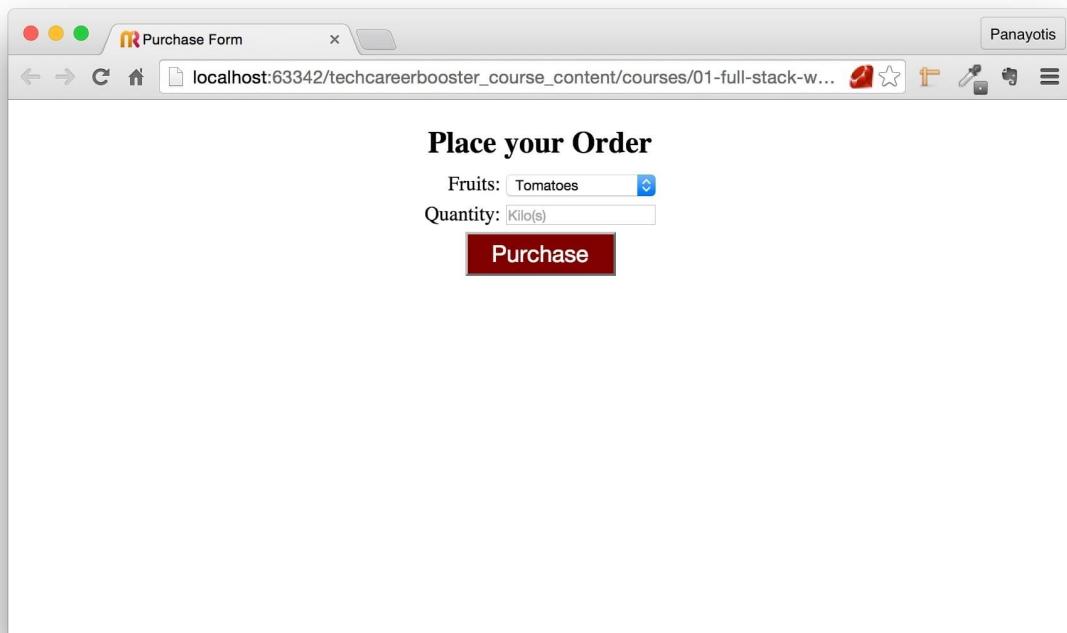
(the above code snippet online)

If you save the above file and reload the page, you will see that trying to set a number minimum from 1 or above 10 will not allow the user to submit the form. Watch this video to see that in action:

How the Quantity is now limited according to min and max attributes

step attribute

Let's see the following purchase form now, in which the quantity represents kilos:



Now Quantity Represents Kilos

So, the HTML content now has as follows:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Purchase Form</title>
6          <link rel="stylesheet" href="stylesheets/purchase.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Place your Order</h2>
11         <form>
12
13             <div class="input-group">
14                 <label for="fruits">Fruits:</label>
15                 <select id="fruits" autofocus="autofocus">
16                     <option>Tomatoes</option>
17                     <option>Potatoes</option>
18                     <option>Apples</option>
19                     <option>Oranges</option>
20                 </select>
21             </div>
22
23             <div class="input-group">
24                 <label for="quantity">Quantity: </label>
25                 <input type="number" id="quantity" required="required" min="1" max="10" p\
laceholder="Kilo(s)"/>
26             </div>
27
28
29             <div class="input-group">
30                 <input type="submit" value="Purchase"/>
31             </div>
32
33         </form>
34
35
36     </body>
37
38 </html>
```

(the above code snippet online)

There is nothing different (if compared to the previous HTML page) other than the `placeholder` attribute, which explains that the quantity represents kilos. The quantity can still be between “1” and “10” (thanks to `min` and `max` attributes).

If you save this file and load the page on your browser, you will see that there is a problem. You cannot set a value to quantity that is not an integer. For example, you cannot purchase 2.5 kilos of potatoes. Watch this video here to understand what we mean:

We cannot set the value to sub-multiples (aliquots) of 1

How can we allow the `number` type `inputs` to accept decimal / float numbers? This is done with the help of the attribute `step`. This specifies the distance from one valid number to the next valid number allowed to be keyed in in the input `number` control. The default value is "1". That's why, only positive, negative or 0 integers are allowed. If you set the value of `step` to "0.1" for example, you will be able to set numbers like "1.2", "1.4", e.t.c.

Let's do that for our fruits purchase form:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Purchase Form</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Place your Order</h2>
11         <form>
12
13             <div class="input-group">
14                 <label for="fruits">Fruits:</label>
15                 <select id="fruits" autofocus="autofocus">
16                     <option>Tomatoes</option>
17                     <option>Potatoes</option>
18                     <option>Apples</option>
19                     <option>Oranges</option>
20                 </select>
21             </div>
22
23             <div class="input-group">
24                 <label for="quantity">Quantity: </label>
25                 <input type="number" id="quantity" required="required" min="1" max="10" s \
26 step="0.1" placeholder="Kilo(s)"/>
27             </div>
28
29             <div class="input-group">
30                 <input type="submit" value="Purchase"/>
31             </div>
32         </form>
33
34     </body>
35
36 </html>
```

(the above code snippet online)

We only added the `step="0.1"` attribute value pair on the `quantity` input control. If you load the page on your browser, you will now be able to specify the 2.5 kilos quantity and any quantity from 1 to 10 with step 0.1.

Let's see that in action:

0.1 Step is used to allow for the value 2.5

However, if you try to enter the value “2.25” (2 1/4 kilos) you will fail again. Because “0.25” is not a sub-multiple of “0.1”. You will be warned that the nearest value to “2.25” that you can enter is either “2.3” or “2.2”.

If you want to allow any decimal number to be entered, you can set the value of the `step` attribute to any. Let's do that on our form:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Purchase Form</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Place your Order</h2>
11         <form>
12
13             <div class="input-group">
14                 <label for="fruits">Fruits:</label>
15                 <select id="fruits" autofocus="autofocus">
16                     <option>Tomatoes</option>
17                     <option>Potatoes</option>
18                     <option>Apples</option>
19                     <option>Oranges</option>
20                 </select>
21             </div>
22
23             <div class="input-group">
24                 <label for="quantity">Quantity: </label>
25                 <input type="number" id="quantity" required="required" min="1" max="10" s\
26 step="any" placeholder="Kilo(s)"/>
27             </div>
28
29             <div class="input-group">
30                 <input type="submit" value="Purchase"/>
31             </div>
32         </form>
33     </body>
34 </html>
```

(the above code snippet online)

If you save the above file and load your page again, you will be able to give any value to the quantity field, as long as it is in the range “1”, “10”.

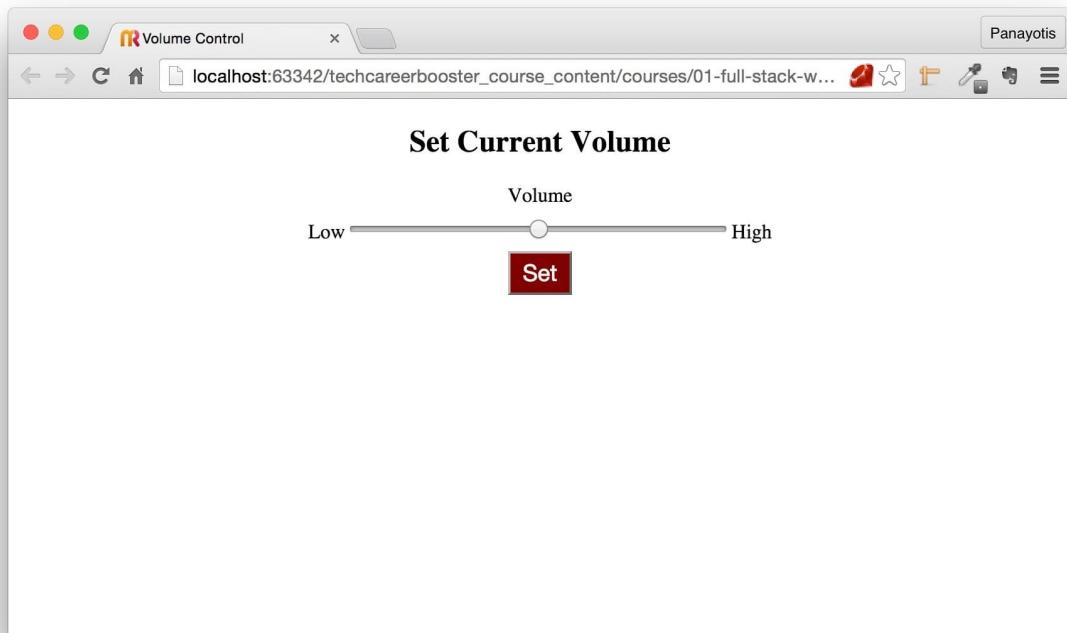
Let's see that in action again:

We can now enter any value to Quantity

input with type range

There is another alternative UI control that can help us collect information related to a range of numbers. Instead of using the type with value number we can use the type with value range. The input controls with type equal to range they are usually been drawn as a slider.

Let's suppose that we are implementing a radio listener and we need to allow the user to set the volume control. Like this:



Form that is using range input

The HTML for this page is very simple:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Volume Control</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9          <h2>Set Current Volume</h2>
10
11     <form>
12         <label for="volume">Volume</label><br/>
13         Low <input type="range" id="volume" min="0" max="50" value="25"/> High<br/>
14         <input type="submit" value="Set"/>
15     </form>
16     </body>
17 </html>
```

(the above code snippet online)

As you can see above, the `input` is of type `range`. It also has a `min` and `max` value. Also, the starting, default value is 25 and is specified by setting the `value` attribute with name `value`.

Save the above file and use this CSS:

```
1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7  html {
8      font-size: 62.5%;
9  }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14     text-align: center;
15 }
16
17 h2 {
18     margin-bottom: 20px;
19 }
20
21 input[type="range"] {
22     width: 300px;
```

```

23     margin-bottom: 10px;
24 }
25
26 label {
27     display: inline-block;
28     margin-bottom: 10px;
29 }
30
31 form input[type='submit'] {
32     font-size: 1.8rem;
33     background-color: Maroon;
34     color: White;
35     padding: 5px 10px;
36 }
37
38 form input[type='submit']:hover {
39     background-color: Black;
40     cursor: pointer;
41 }
```

(the above code snippet online)

to see the form in action.

value attribute

In the purchase form with the products and the quantity, there is a small problem with the quantity field that, when the form is loaded, the quantity does not have any prepopulated value. We would like the value of quantity to be 1 when the form loads. How can we do that?

We can do that if we use the attribute `value` and set its value to 1. Let's see that:

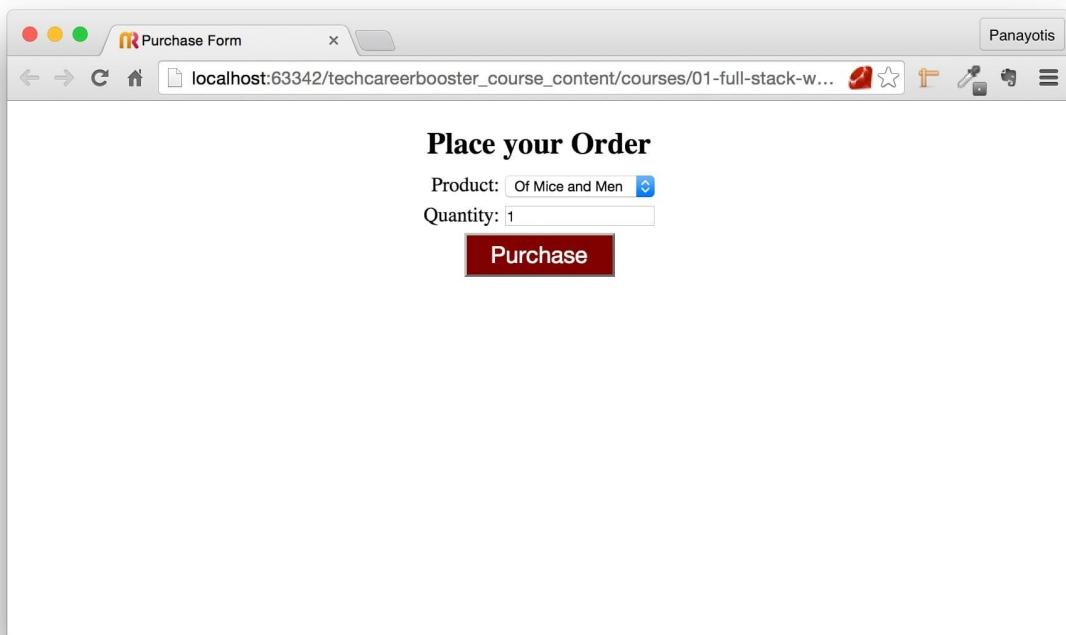
```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Purchase Form</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10    <h2>Place your Order</h2>
11    <form>
12
13      <div class="input-group">
14        <label for="product">Product:</label>
15        <select id="product" autofocus="autofocus">
```

```
16      <option>Of Mice and Men</option>
17      <option>Game of Thrones</option>
18      <option>Star Wars II</option>
19    </select>
20  </div>
21
22  <div class="input-group">
23    <label for="quantity">Quantity: </label>
24    <input type="number" id="quantity" required="required" min="1" max="10" v\
25 alue="1"/>
26  </div>
27
28  <div class="input-group">
29    <input type="submit" value="Purchase"/>
30  </div>
31 </form>
32
33 </body>
34 </html>
```

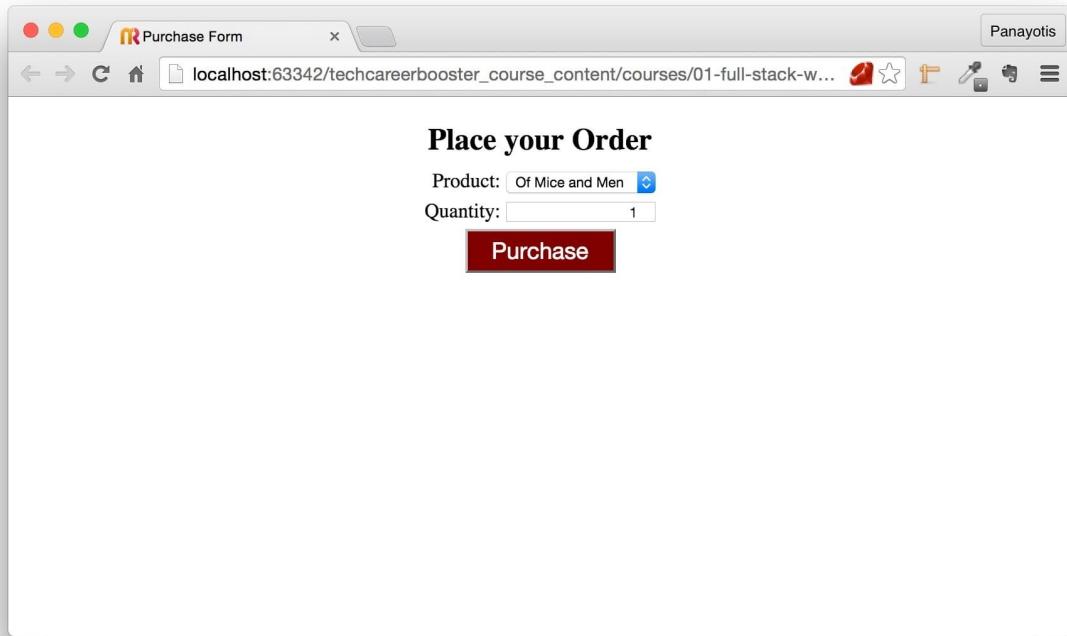
(the above code snippet online)

Do you see the `value="1"` on the quantity input element? If you save this and load the page, you will see that the quantity is pre-populated with the value 1:



Quantity Prepopulated with value 1

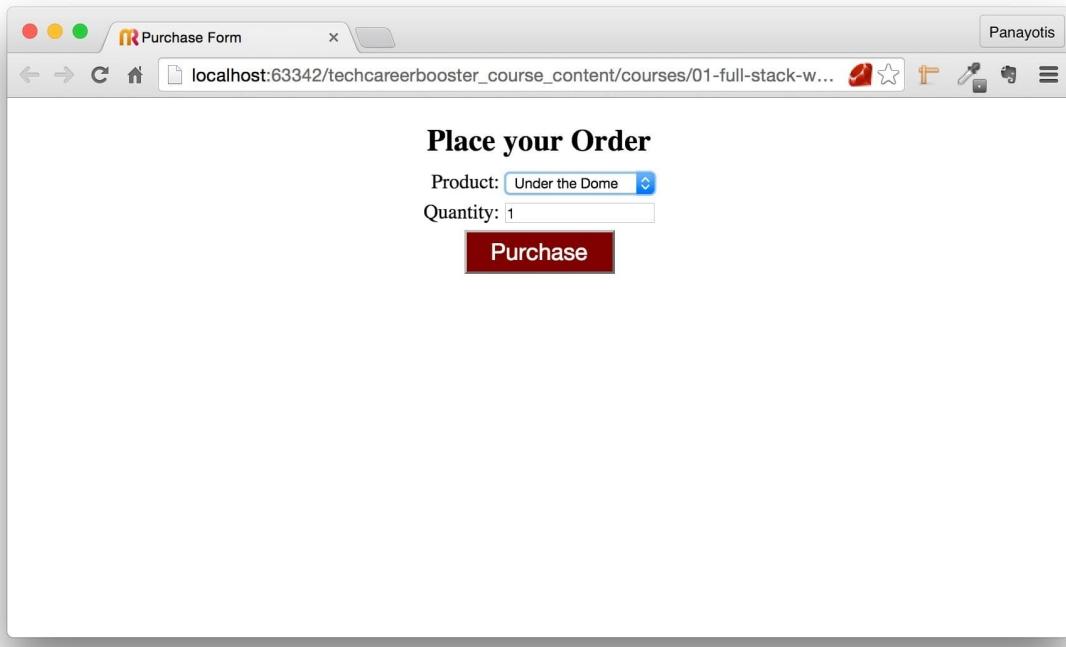
Question: How would you make the value inside the quantity box appear right aligned rather than left as it is now? How would you do this?



Quantity Value Right Aligned

Grouping Selection Options

When we have a long list of options, we usually decide to present them in alphabetical order, so that it is easier for the users to select the option they prefer. Another extra facility that we have in long list of options, is the grouping of options by a category. Let's suppose that we want to group the products on our purchase form by their category:



Grouping Of Products

Watch the following video to see this form in action:

Working on the Form that Groups Products

Since we have already learned about the `select` box, adding the grouping feature on a `select` box is easy. We only have to group options inside an `optgroup` HTML element. The `optgroup` encloses the `option` elements that are part of the group. Also, it has a `label` attribute that has the value that plays the role of the name of the group.

Hence, the following is the HTML page that groups the products in two groups/categories. The TV Series and the Movies:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Purchase Form</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Place your Order</h2>
11         <form>
12
13             <div class="input-group">
14                 <label for="product">Product:</label>
```

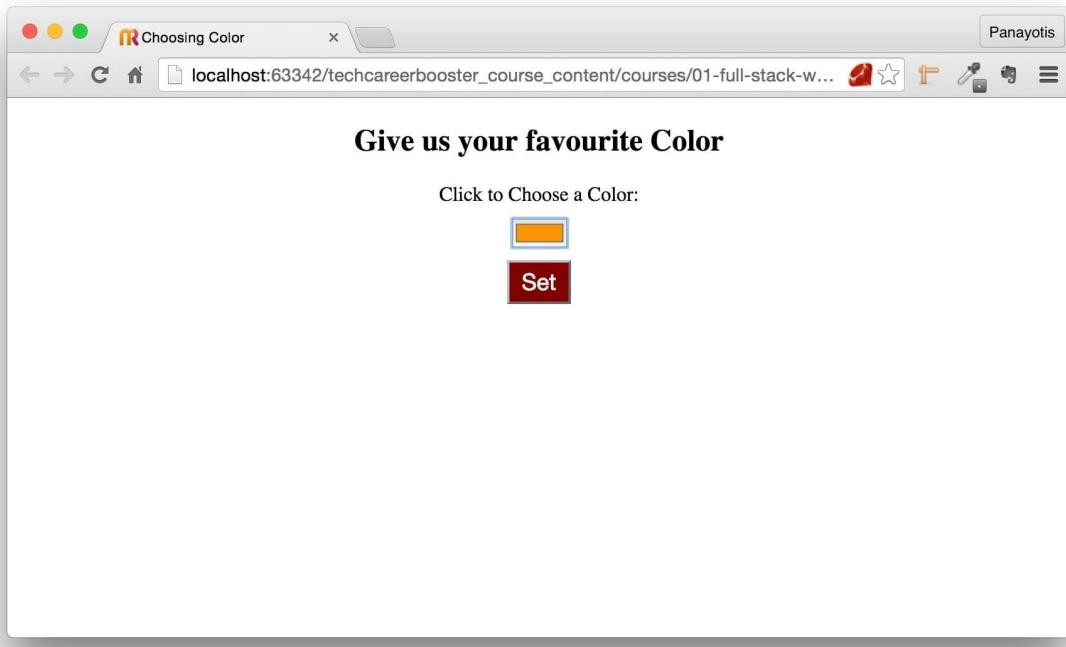
```
15      <select id="product" autofocus="autofocus">
16          <optgroup label="TV Series">
17              <option>Under the Dome</option>
18              <option>Game of Thrones</option>
19              <option>Dexter</option>
20          </optgroup>
21          <optgroup label="Movies">
22              <option>Star Wars I</option>
23              <option>The Beauty and The Beast</option>
24              <option>James Bond - Skyfall</option>
25          </optgroup>
26      </select>
27  </div>
28
29  <div class="input-group">
30      <label for="quantity">Quantity: </label>
31      <input type="number" id="quantity" required="required" min="1" max="10" value="1"/>
32  </div>
33
34
35  <div class="input-group">
36      <input type="submit" value="Purchase"/>
37  </div>
38 </form>
39
40 </body>
41
42 </html>
```

(the above code snippet online)

Do you see the `<optgroup> . . . </optgroup>` above and how it is being used? When you save this page and load it on your browser, you will see the drop down list box presenting its options in two groups.

input of type color

Sometimes we want to ask the user to specify a color. There is a handy input of type `color` that can be used for this case. See the following form:



Form with input of type color

Do you see that there is a colored box in the middle? This can be clicked by the user and a color picker window will pop up to allow the user to select the color he wants. See this form in action here:

How user is using the color picker

In order to implement this simple form, use the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Choosing Color</title>
6     <link rel="stylesheet" href="stylesheets/color.css" type="text/css">
7   </head>
8   <body>
9     <h2>Give us your favourite Color</h2>
10    <form>
11      <label for="color">Click to Choose a Color:</label><br/>
12      <input type="color" id="color"/><br/>
13      <input type="submit" value="Set"/>
14    </form>
15  </body>
16 </html>
```

(the above code snippet online)

with the following CSS:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14     text-align: center;
15 }
16
17 h2 {
18     margin-bottom: 20px;
19 }
20
21 input[type="color"] {
22     margin-bottom: 10px;
23 }
24
25 label {
26     display: inline-block;
27     margin-bottom: 10px;
28 }
29
30 form input[type='submit'] {
31     font-size: 1.8rem;
32     background-color: Maroon;
33     color: White;
34     padding: 5px 10px;
35 }
36
37 form input[type='submit']:hover {
38     background-color: Black;
39     cursor: pointer;
40 }
```

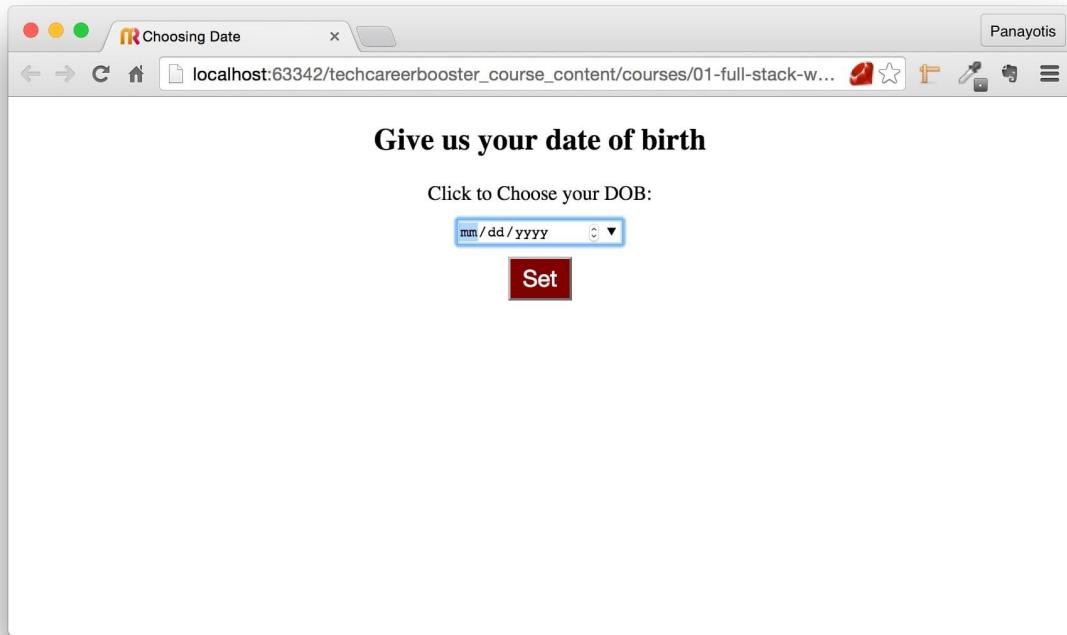
(the above code snippet online)

As you can see, the `input` of type `color` is very simple.

input of type date

Note: This may not be supported on all browsers. We are working here with Chrome.

We will often come across the need to ask our users for a date. For example, the following form asks for the date of birth:



Asking for Date of Birth

See this form in action:

How the date picker is being used by the user

Things are very simple again. You only have to use an `input` of type `date`. The rest of the job is done by your browser. Your browser will display the correct input control to ease the user when specifying their date of birth.

Here is the HTML code for this page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Choosing Date</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9          <h2>Give us your date of birth</h2>
```

```
10   <form>
11     <label for="dob">Click to Choose your DOB:</label><br/>
12     <input type="date" id="dob" autofocus="autofocus"/><br/>
13     <input type="submit" value="Set"/>
14   </form>
15 </body>
16 </html>
```

(the above code snippet online)

And the corresponding CSS:

```
1  * {
2    box-sizing: border-box;
3    padding: 0;
4    margin: 0;
5  }
6
7 html {
8   font-size: 62.5%;
9 }
10
11 body {
12   font-size: 1.6rem;
13   margin-top: 20px;
14   text-align: center;
15 }
16
17 h2 {
18   margin-bottom: 20px;
19 }
20
21 input[type="date"] {
22   margin-bottom: 10px;
23 }
24
25 label {
26   display: inline-block;
27   margin-bottom: 10px;
28 }
29
30 form input[type='submit'] {
31   font-size: 1.8rem;
32   background-color: Maroon;
33   color: White;
34   padding: 5px 10px;
35 }
```

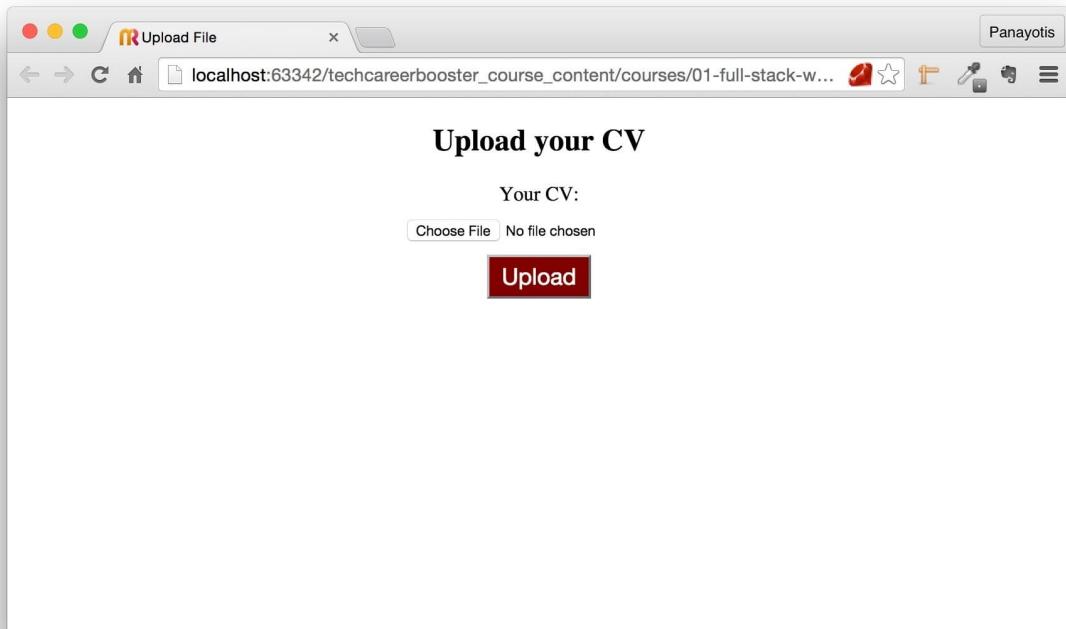
```
36
37 form input[type='submit']:hover {
38     background-color: Black;
39     cursor: pointer;
40 }
```

(the above code snippet online)

As you can see the input of type date is enough.

input of type file

Sometimes you want to ask the user to send a file, a document. For example, when we want the user to upload his CV. Like in the form below:



Form to Upload a File

As you can see above, there is a button `Choose File` which allows the user to click and attach a file to the form. Then when the user clicks on `Upload` submit button, the form will be submitted alongside the file attached.

Here is the HTML code for such a form:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Upload File</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10         <h2>Upload your CV</h2>
11         <form>
12             <label for="cv">Your CV:</label><br/>
13             <input type="file" id="cv"/><br/>
14             <input type="submit" value="Upload"/>
15         </form>
16
17     </body>
18
19 </html>
```

(the above code snippet online)

It is very simple. The `input` is of type `file`. Use the following CSS code:

```
1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7  html {
8      font-size: 62.5%;
9  }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14     text-align: center;
15 }
16
17 h2 {
18     margin-bottom: 20px;
19 }
20
21 input[type="file"] {
22     margin-bottom: 10px;
23 }
```

```
24
25 label {
26     display: inline-block;
27     margin-bottom: 10px;
28 }
29
30 form input[type='submit'] {
31     font-size: 1.8rem;
32     background-color: Maroon;
33     color: White;
34     padding: 5px 10px;
35 }
36
37 form input[type='submit']::hover {
38     background-color: Black;
39     cursor: pointer;
40 }
```

(the above code snippet online)

Save all files and load your HTML page. You will see the form as given above in the screenshot. Click on the button Choose File and you will be prompted to choose a file from your local machine.

Grouping Form Fields

We continue this first encounter with forms using a way to visually group fields of a form. Look at the following example:

The screenshot shows a web browser window titled "Checkout" with the URL "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-w...". The page content is titled "Place your Order". It features two **fieldset** elements. The first **fieldset** is labeled "Product Details" and contains a **label** for "Product:" followed by a **select** element with an **autofocus** attribute, and a **label** for "Quantity:" followed by a **input** element with a value of "1". The second **fieldset** is labeled "Shipping Address" and contains three **label** elements: "Address:", "Zip Code:", and "Country:", each followed by its respective **input** element. At the bottom of the form is a large red **button** with the text "Purchase".

Checkout Form With Grouping of Fields

As you can see above, we have grouped the fields of the form into 2 groups. Product Details and Shipping Address. This is done with the help of two HTML tags, the **fieldset** and the **legend**. The **fieldset** is actually used to group the fields and the **legend** is used to add the title on top left. We open the **fieldset** with the opening tag **<fieldset>** and then we write, first, the **legend** and then the form fields that are part of the group.

Let's see the HTML code for the above page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Checkout</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <h2>Place your Order</h2>
12         <form>
13             <fieldset>
14                 <legend>Product Details</legend>
15                 <div class="input-group">
16                     <label for="product">Product:</label>
17                     <select id="product" autofocus="autofocus">
```

```
18      <option>Product...</option>
19      <option>Game Of Thrones</option>
20      <option>Of Mice And Men</option>
21      <option>James Bond</option>
22      <option>Star Wars</option>
23  </select>
24 </div>
25
26 <div class="input-group">
27   <label for="quantity">Quantity:</label>
28   <input type="number" id="quantity" min="1" max="10" value="1"/>
29 </div>
30
31 </fieldset>
32
33 <fieldset>
34   <legend>Shipping Address</legend>
35   <div class="input-group">
36     <label for="address">Address:</label>
37     <input type="text" id="address" placeholder="Address"/>
38   </div>
39
40   <div class="input-group">
41     <label for="zip-code">Zip Code:</label>
42     <input type="text" id="zip-code" placeholder="Zip Code"/>
43   </div>
44   <div class="input-group">
45     <label for="country">Country:</label>
46     <select id="country">
47       <option>Country...</option>
48       <option>Afghanistan</option>
49       <!-- ... rest of the countries go here ... -->
50       <option>Zimbabwe</option>
51     </select>
52   </div>
53 </fieldset>
54
55 <div class="input-group">
56   <input type="submit" value="Purchase"/>
57 </div>
58 </form>
59
60 </body>
61
62 </html>
```

(the above code snippet online)

Do you see how the `<fieldset>....</fieldset>` has been used to group the fields of the form into 2 groups? Do you see the `<legend>....</legend>` which is exactly below the opening `<fieldset>` tag?

Note that `<fieldset>` is a block level element and hence, when used, no other element exists left or right to it.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Create a Web page with a form like this:

Task With Form

2. Here are some things that you need to take care of and some hints to help you develop this page:

1. We remind you that `form` is a block level element.
2. `fieldset` is also a block level element.
3. You may not need divs to divide your page into areas. Since `fieldset` is a block level element, you can use only fieldsets to divide your page.
4. The required fields are:
 1. departure city
 2. destination city
 3. departure date
 4. email
 5. mobile
 6. number of tickets
 7. first passenger first name
 8. first passenger last name
 9. first passenger passport
 10. first passenger date of birth
5. Make sure you use the appropriate `type` value when defining your `input` elements.
6. Number of tickets should take one of the two values: 1, 2. And the initial value is 1.
7. When page loads the focus should be on the departure city field.
8. The optional message area cannot be resized. Its height is such that one can type in 5 rows before scrolling is enabled.
9. Use `title` attribute for all the input elements to further explain what needs to be entered in each field.
10. For the drop downs with the cities, you can use the following list:
 1. Boston
 2. Los Angeles
 3. New York
 4. Phoenix
 5. San Diego
 6. Santa Barbara
 7. Tijuana

Chapter 28 - Advanced CSS Selectors

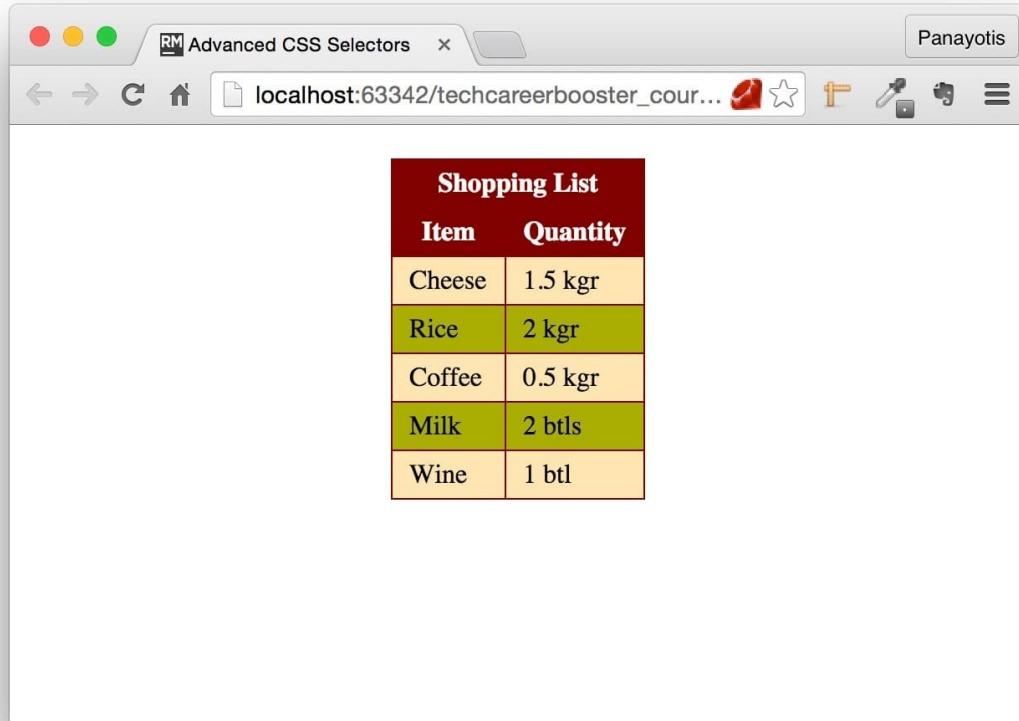
Summary

We have learned a lot of CSS selectors until now, but this chapter has the intention to teach some of the more advanced topics on this subject.

We will deal with three categories of CSS selectors:

1. The pseudo-classes
2. The pseudo-elements
3. Other advanced CSS selectors

The pseudo-classes will allow us to style our elements based on their state or other characteristic of them. Hence, you will be able, for example to style the even rows of a table with a different color from the odd ones.

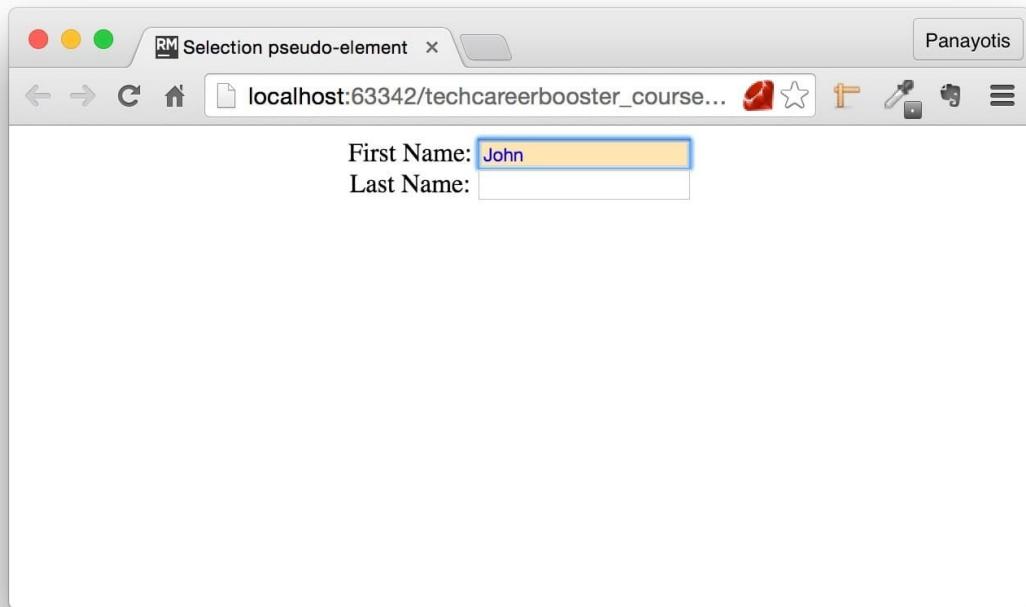


A screenshot of a web browser window titled "Advanced CSS Selectors". The address bar shows "localhost:63342/techcareerbooster_cour...". The main content area displays a table with a dark red header and alternating green and yellow rows. The table is titled "Shopping List" and has columns "Item" and "Quantity". The data is as follows:

| Item | Quantity |
|--------|----------|
| Cheese | 1.5 kgr |
| Rice | 2 kgr |
| Coffee | 0.5 kgr |
| Milk | 2 btls |
| Wine | 1 btl |

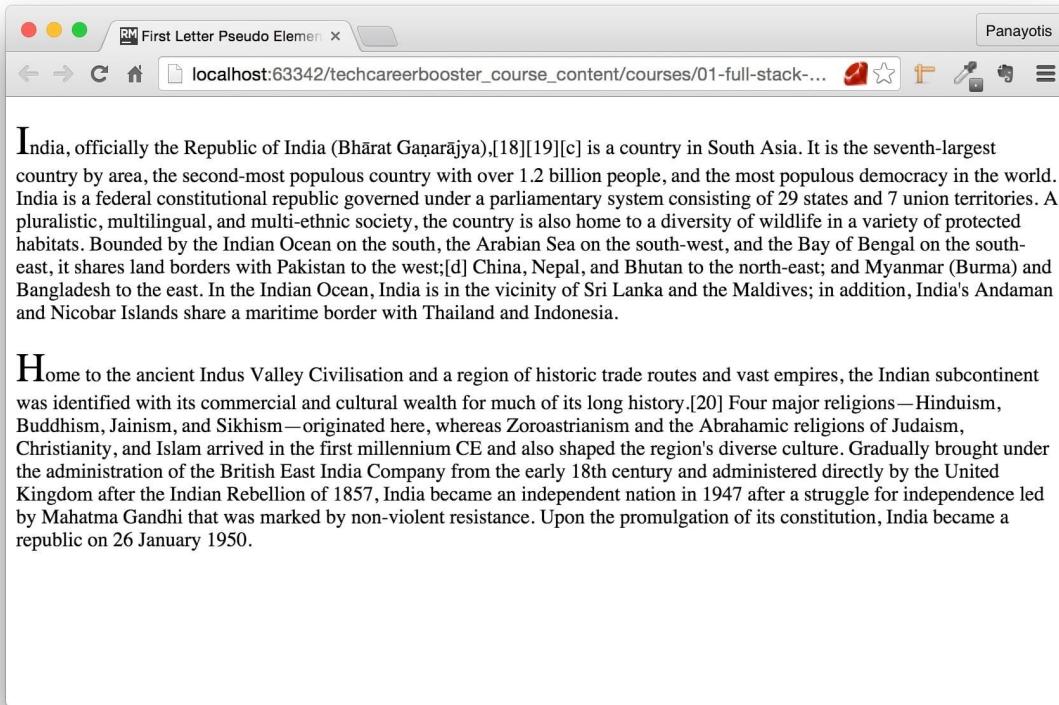
Table With Different Style on Even vs Odd Rows

You will also be able to change the style of an input element based on its focus state:



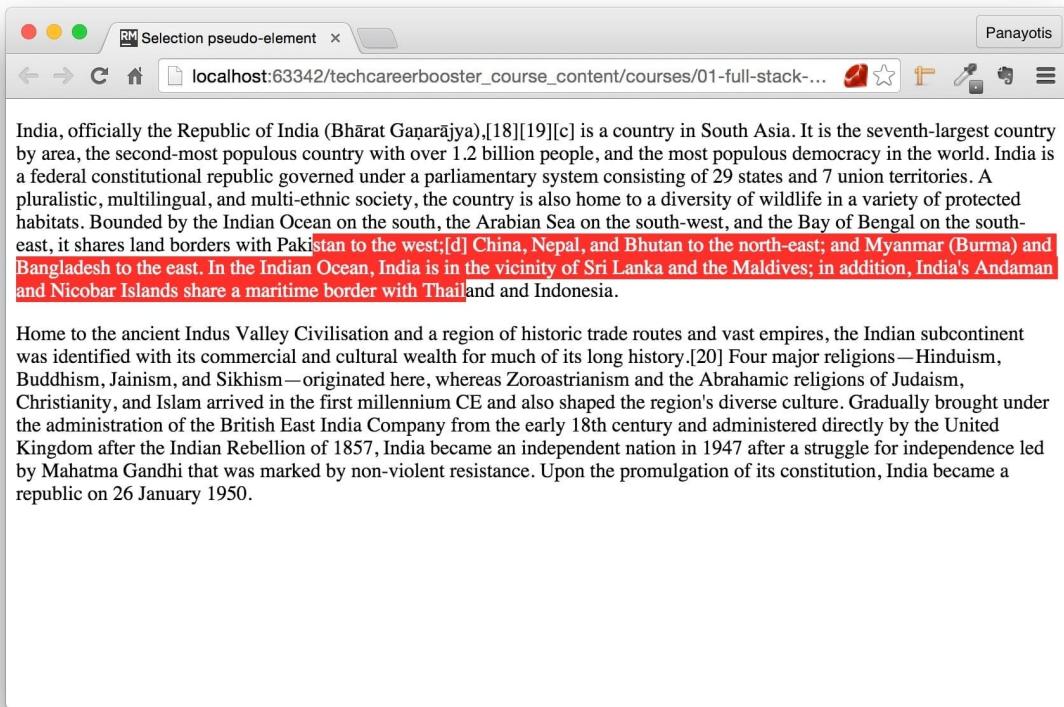
Blue Text on Focused Field

The pseudo-elements allow us to refer to elements that do not really exist at the source of the HTML page as distinct elements. For example, we can dynamically define an element before the content of an anchor element. Or we can refer to the first letter of a paragraph and style it differently:



First Letter Of Paragraph with distinct style

Or you can define the properties of selected parts of the page like here:



Selected part Styled with Red and White

Finally, you will learn about other useful CSS selectors like:

1. element > element
2. element + element
3. [attribute]
4. [attribute=value] or [attribute="value"] or [attribute='value']
5. [attribute~=value] or [attribute~=="value"] or [attribute~='value']
6. [attribute*=value] or [attribute*="value"] or [attribute*='value']
7. [attribute|=value] or [attribute|="value"] or [attribute|='value']
8. [attribute^=value] or [attribute^="value" or [attribute^='value']]
9. [attribute\$=value] or [attribute\$="value"] or [attribute\$='value']

Learning Goals

1. Learn about pseudo-classes.
 1. anchor a pseudo-classes.
 2. :first-child
 3. :last-child
 4. :first-of-type
 5. :last-of-type
 6. :nth-child(n)

7. :nth-of-type(n)
8. :not(selector)
9. :optional
10. :required
11. :focus
2. Learn about pseudo-elements
 1. ::first-line
 2. ::first-letter
 3. ::selection
 4. ::before and ::after
3. Learn about other advanced CSS selectors, like:
 1. element > element
 2. element + element
 3. [attribute]
 4. [attribute=value] or [attribute="value"] or [attribute='value']
 5. [attribute~=value] or [attribute~="value"] or [attribute~='value']
 6. [attribute*=value] or [attribute*="value"] or [attribute*='value']
 7. [attribute|=value] or [attribute|="value"] or [attribute|='value']
 8. [attribute^=value] or [attribute^="value" or [attribute^='value']]
 9. [attribute\$=value] or [attribute\$="value"] or [attribute\$='value']

Pseudo-classes

We have already worked with selectors like this:

```
1 a:link {
2   color: Blue;
3 }
```

(the above code snippet online)

which is the unvisited link selector. The `:link` part of the selector defines a state or a characteristic of the actual element, of the anchor `a` element. It is a link that has not been visited. This kind of element attribution to specify a particular state or characteristic of the element is called a *pseudo-class*.

About the states of an anchor a

There are some pseudo-classes that are used for anchor `a` elements. These are the following:

1. `:link`, is for an unvisited anchor
2. `:visited`, is for a visited anchor
3. `:hover`, is for an anchor when the mouse is over it
4. `:active`, is for an anchor that is selected, i.e. clicked but mouse button has not been released.

Important: `:hover` styling must be defined AFTER `:link` and `:visited`. And `:active` must come after `:hover`. So, if you remember the acronym LVHA, then you will never make a mistake in the order.

:first-child and :first-of-type. :last-child and :last-of-type

The pseudo-class :first-child specifies that the element needs to be the first child of a containing element. Let's see the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Advanced CSS Selectors</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9
10     <div id="wrapper">
11         <p>
12             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod t\
13             empor incididunt ut labore et
14                 dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation u\
15             llamco laboris nisi ut aliquip ex ea
16                 commodo consequat. Duis aute irure dolor in reprehenderit in voluptate ve\
17             lit esse cillum dolore eu fugiat nulla
18                 pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa q\
19             ui officia deserunt mollit anim id est laborum.
20         </p>
21
22         <p>
23             Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantiu\
24             m doloremque laudantium, totam rem aperiam, eaque ipsa
25                 quae ab illo inventore veritatis et quasi architecto beatae vitae dicta s\
26             unt explicabo. Nemo enim ipsam voluptatem quia voluptas
27                 sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eo\
28             s qui ratione voluptatem sequi nesciunt. Neque porro
29                 quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisc\
30             i velit, sed quia non numquam eius modi tempora
31                 incident ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim a\
32             d minima veniam, quis nostrum exercitationem
33                 ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequ\
34             atur? Quis autem vel eum iure reprehenderit
35                 qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illu\
36             m qui dolorem eum fugiat quo voluptas nulla pariatur?
37         </p>
38     </div>
39 </body>
40 </html>
```

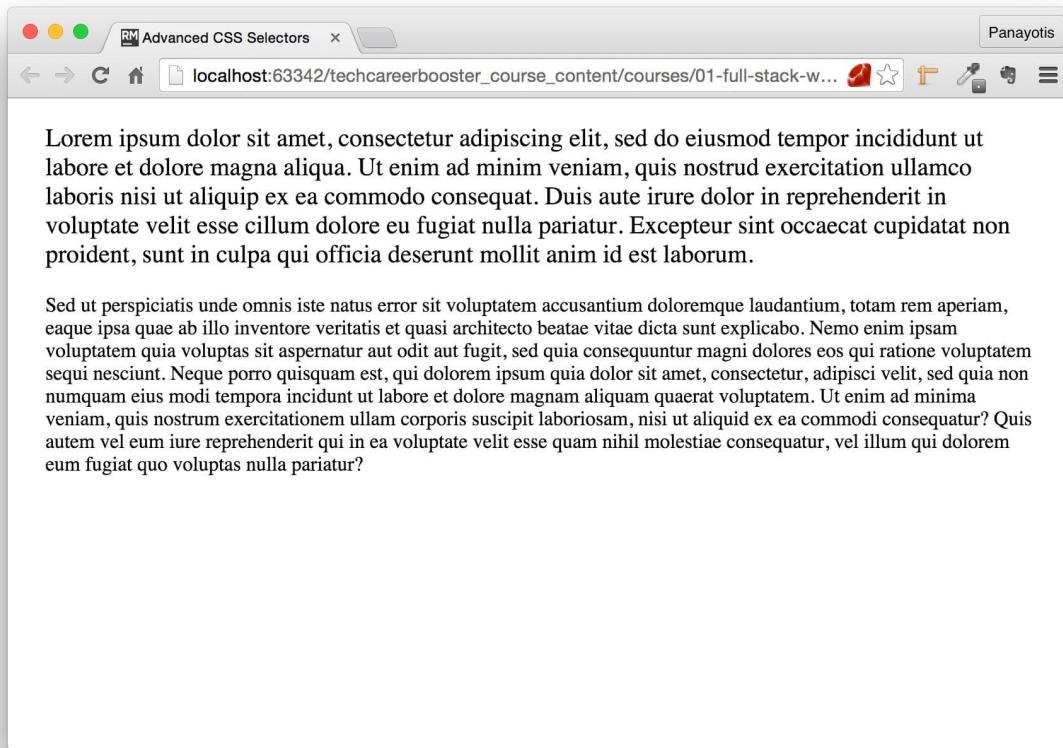
(the above code snippet online)

As you can see above, there are 2 paragraphs in this document. What if we wanted to set the styling for the first paragraph, but not for the second? We can apply the following CSS:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14 }
15
16 #wrapper {
17     margin: 0 30px;
18 }
19
20 p {
21     margin-bottom: 20px;
22 }
23
24 p:first-child {
25     font-size: 2rem;
26 }
```

(the above code snippet online)

Do you see the `p:first-child`. This is the way to select the first child `p` element. The first paragraph has a little bit larger font. Here is the final result:



Selecting First Paragraph Child

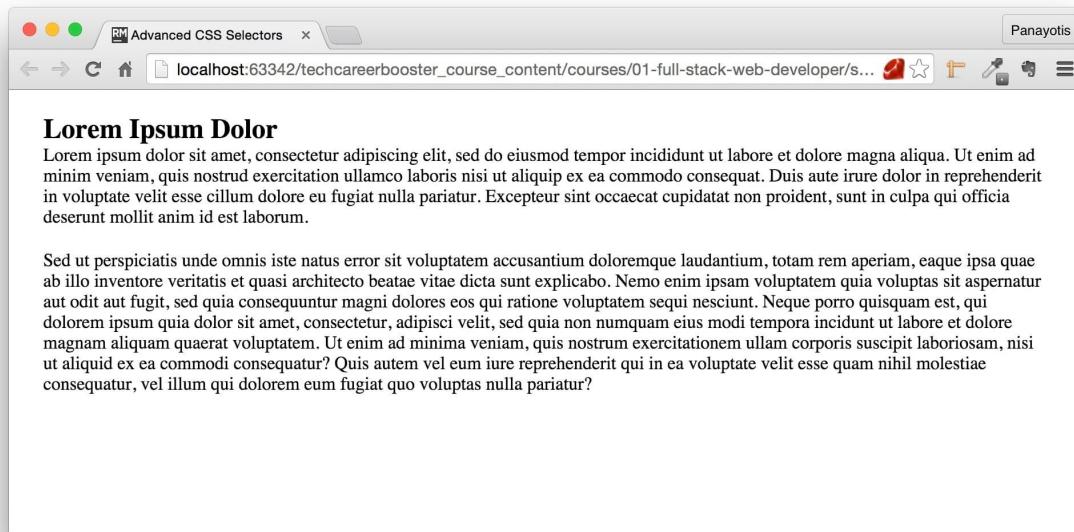
However, let's see this example here:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Advanced CSS Selectors</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8      <body>
9
10         <div id="wrapper">
11             <h2>Lorem Ipsum Dolor</h2>
12
13             <p>
14                 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod t\
15                 empor incididunt ut labore et
16                     dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation u\
17                     llamco laboris nisi ut aliquip ex ea
18                         commodo consequat. Duis aute irure dolor in reprehenderit in voluptate ve\
19                         lit esse cillum dolore eu fugiat nulla
```

```
20      pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa q\
21  ui officia deserunt mollit anim id est laborum.
22      </p>
23
24      <p>
25          Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantiu\
26  m doloremque laudantium, totam rem aperiam, eaque ipsa
27          quae ab illo inventore veritatis et quasi architecto beatae vitae dicta s\
28  unt explicabo. Nemo enim ipsam voluptatem quia voluptas
29          sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eo\
30  s qui ratione voluptatem sequi nesciunt. Neque porro
31          quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisc\
32  i velit, sed quia non numquam eius modi tempora
33          incident ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim a\
34  d minima veniam, quis nostrum exercitationem
35          ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequ\
36  atur? Quis autem vel eum iure reprehenderit
37          qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illu\
38  m qui dolorem eum fugiat quo voluptas nulla pariatur?
39      </p>
40
41      </div>
42  </body>
43 </html>
```

(the above code snippet online)

If you save the above and reload it on your browser, you will see this:



First paragraph is not changed

This is because, now, the `p` is *not the first child* of its containing element. There is a solution to this and it is called `:first-of-type`. `:first-of-type` matches the first child that has the type of the element attributed. Hence `p:first-of-type` will select the first `p` element that is child of a containing element. Hence, `p` can be in any child position, first, second, third or any larger, but it has to be the first of type `p`.

Let's change the CSS:

```

1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7  html {
8      font-size: 62.5%;
9  }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14 }
15
16 #wrapper {
17     margin: 0 30px;
18 }
19
20 p {

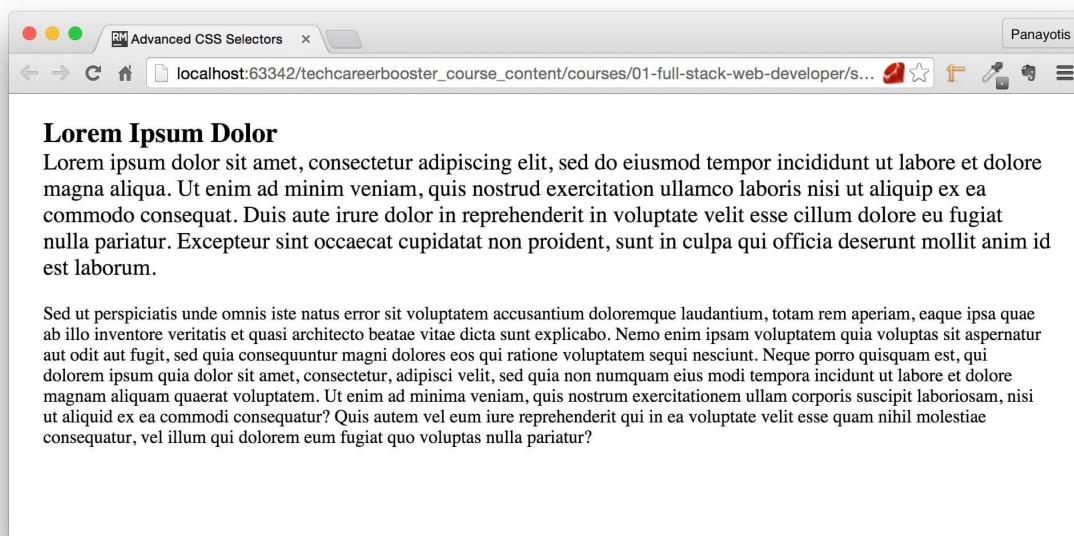
```

```

21     margin-bottom: 20px;
22 }
23
24 p:first-of-type {
25   font-size: 2rem;
26 }
```

(the above code snippet online)

and reload the last page. You will see this:



First paragraph is selected using :first-of-type

As you can see above, the first paragraph is now styled with larger font. It has been selected with the `:first-of-type` pseudo-class.

Now that you have learned how `:first-child` and `:first-of-type` work, you can easily guess what the `:last-child` and `:last-of-type` do too.

`:nth-child(n)` and `:nth-of-type(n)`

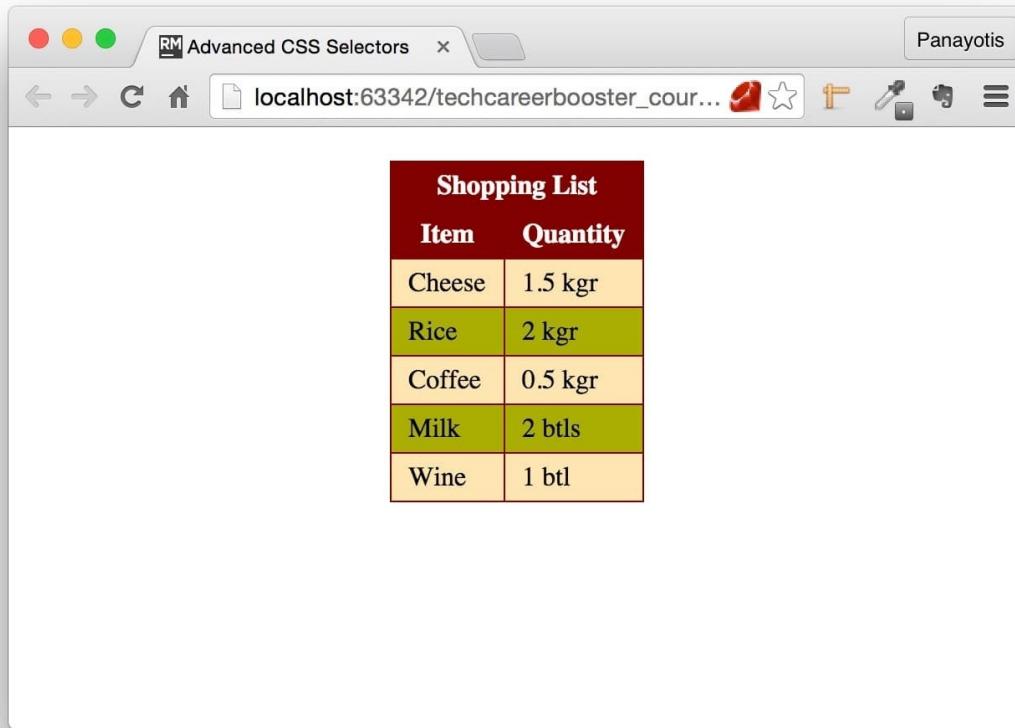
These are generalizations of the previous ones. So, instead of being limited to `first` and `last` you can now use the `:nth-child(n)` and `:nth-of-type(n)` to specify any child position. So,

1. if you want to specify the `p` elements that are on 2nd child position of their parent: `:nth-child(2)`
2. If you want to specify the `p` elements that are on 2nd position amongst the `p` children of their parent: `:nth-of-type(2)`

However, these pseudo-classes here can be even more generic, because the value `n` inside the parentheses can also be:

1. a keyword or
2. a formula

Let's have a look at the following page:



A screenshot of a web browser window titled "Advanced CSS Selectors". The address bar shows "localhost:63342/techcareerbooster_cour...". The main content area displays a table with a dark red header and alternating green and orange rows. The table has two columns: "Item" and "Quantity". The data is as follows:

| Shopping List | |
|---------------|----------|
| Item | Quantity |
| Cheese | 1.5 kgr |
| Rice | 2 kgr |
| Coffee | 0.5 kgr |
| Milk | 2 btls |
| Wine | 1 btl |

Table With Different Style on Even vs Odd Rows

As you can see above, the table body, has different styling on odd vs even rows. Here is how we can achieve that:

The HTML code is the following:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Advanced CSS Selectors</title>
6      <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7    </head>
8    <body>
9
10   <table>
11
```

```

12     <thead>
13         <tr><th colspan="2">Shopping List</th></tr>
14         <tr><th>Item</th><th>Quantity</th></tr>
15     </thead>
16
17     <tbody>
18         <tr><td>Cheese</td><td>1.5 kgr</td></tr>
19         <tr><td>Rice</td><td>2 kgr</td></tr>
20         <tr><td>Coffee</td><td>0.5 kgr</td></tr>
21         <tr><td>Milk</td><td>2 btl</td></tr>
22         <tr><td>Wine</td><td>1 btl</td></tr>
23     </tbody>
24
25     </table>
26 </body>
27 </html>

```

(the above code snippet online)

which is pretty straight forward. Nothing new here. Note that there is not styling attributes on the HTML markup for the table. All the styling will take place inside the CSS file. So, read carefully the following:

```

1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7  html {
8      font-size: 62.5%;
9  }
10
11 body {
12     font-size: 1.6rem;
13     margin-top: 20px;
14 }
15
16 table {
17     /* Next rule will make the borders of table cells to collapse and appear uniformly */
18     border-collapse: collapse;
19     /* Next rule will make the table appear in the horizontal middle of the page \
20     */
21     margin: 0 auto;
22 }
23
24

```

```

25 table ,
26 table th,
27 table td {
28     /* This is necessary to give a border around the table and in between the cel\
ls */
29     border: 1px solid maroon;
30 }
31
32
33 th, td {
34     /* This is necessary to allow for some space around the text inside the table\
cells */
35     padding: 5px 10px;
36 }
37
38
39 th {
40     background-color: maroon;
41     color: white;
42 }
43
44 /* This is the new stuff here. The first CSS selector selects the odd rows ...*/
45 tbody tr:nth-child(odd) {
46     background-color: #FFE4B5;
47 }
48
49 /* ...and the 2nd CSS selector, selects the even rows */
50 tbody tr:nth-child(even) {
51     background-color: #aaad21;
52 }
```

(the above code snippet online)

Please, read the explanations provided inside the CSS file. The new stuff is here:

```

1 tbody tr:nth-child(odd) {
2     background-color: #FFE4B5;
3 }
4
5 tbody tr:nth-child(even) {
6     background-color: #aaad21;
7 }
```

(the above code snippet online)

Do you see how we select the odd and the even rows inside the tbody element? We are using the pseudo-class :nth-child giving as input argument to the parentheses the odd and even keywords respectively.

Besides the two keywords above, you can also use a formula inside the `:nth-child` parentheses. The formula is of the format: `a n + b` where `a` is a cycle size, `n` is the counter that starts from 0 and `b` is an offset value.

Here are some example:

1. `2n`, takes the values 0, 2, 4, 6... e.t.c. Hence it selects all the even positions.
2. `3n`, takes the values 0, 3, 6, 9... e.t.c. Hence it selects all the positions that are multiple of 3.
3. `2n + 1`, takes the values 1, 3, 5... e.t.c. Hence it selects all the odd positions.

Some other useful pseudo-classes

Some other useful pseudo-classes are the following:

`:not(selector)`

This one selects every element that does not match the selector inside the parentheses. For example: `p:not(.lead)` selects all the paragraphs that do not have the class `lead`.

`:optional and :required`

Selects all the elements that do not have the attribute `required` set. For example: `input:optional` will select all the `input` elements that do not have `required="required"`.

On the other hand, the `:required` pseudo-class does the oposite. Selects all the elements that have the `required="required"` attribute.

`:focus`

This pseudo-class is matching the focus state of an input element. So, you can change the way an input element is styled when the focus is on that element.

Let's see the following HTML page:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Selection pseudo-element</title>
6          <link rel="stylesheet" href="stylesheets/focus.css" type="text/css"/>
7      </head>
8
9      <body>
10
11         <form>
12             <div class="input-group">
13                 <label for="first-name">First Name:</label>
14                 <input id="first-name" type="text" autofocus="autofocus"/>

```

```
15     </div>
16
17     <div class="input-group">
18         <label for="last-name">Last Name:</label>
19         <input id="last-name" type="text"/>
20     </div>
21 </form>
22
23 </body>
24 </html>
```

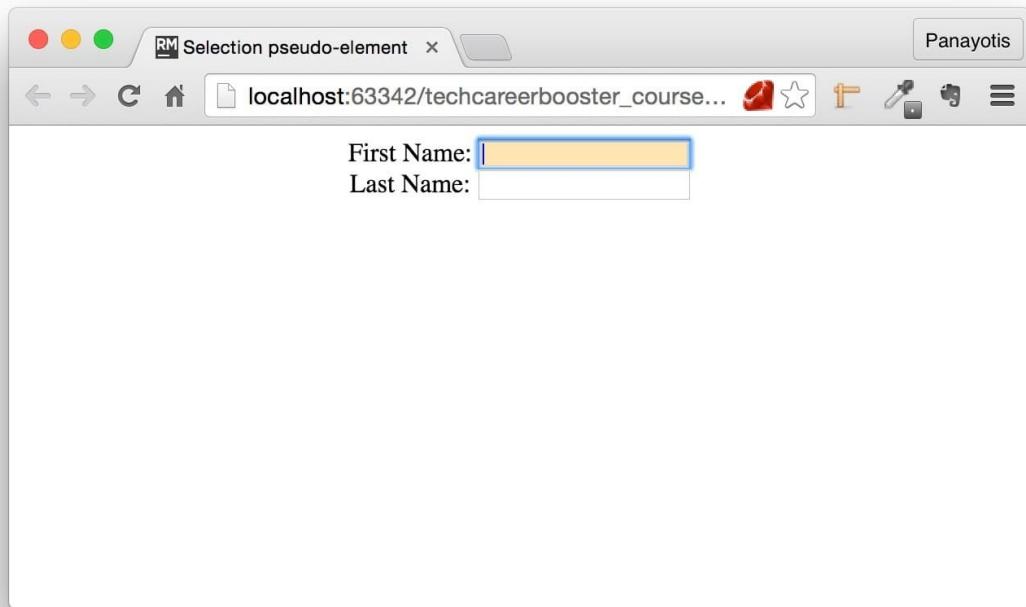
(the above code snippet online)

And the accompanying CSS:

```
1 body {
2     text-align: center;
3 }
4
5 input[type=text]:focus {
6     background-color: #FFE4B5;
7     color: blue;
8 }
```

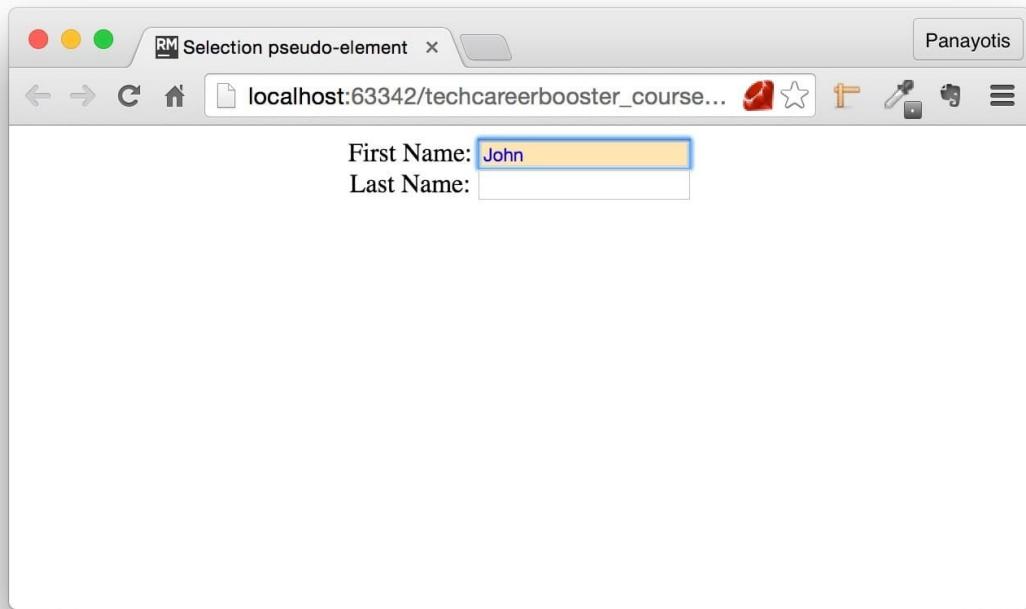
(the above code snippet online)

If you save the two files and load the HTML page on your browser, you will see this:



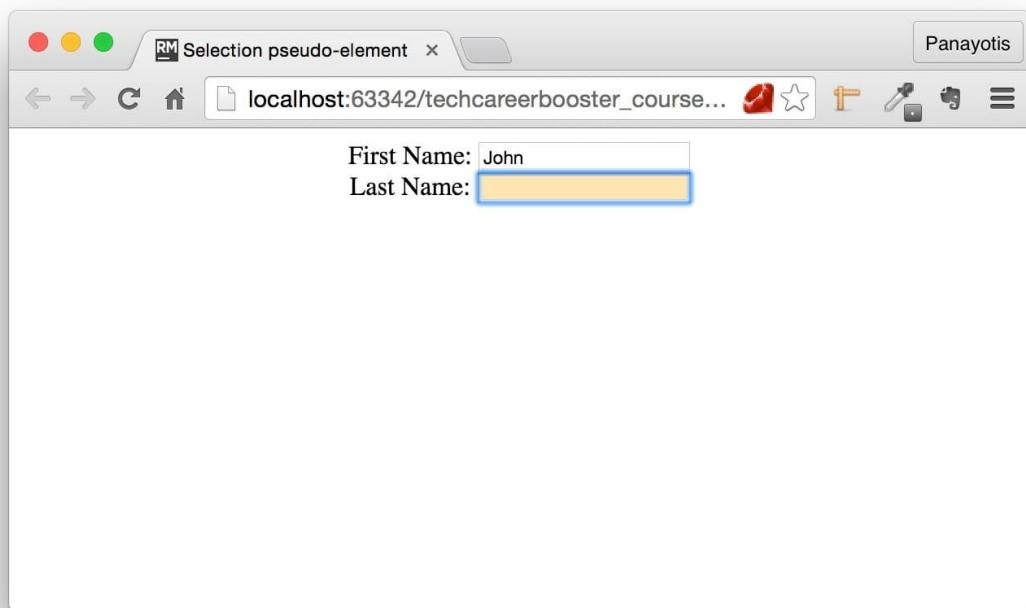
Focused Input Element with Special Styling

As you can see above, the focused input element has background color #FFE4B5. Also, if you type in some text you will see that it is blue:



Blue Text on Focused Field

whereas when the focus is out to another field the text and background are with default colors:



Non-Focused Element has Default Colors

Pseudo-elements

Except from the pseudo-classes, which are used to select elements with specific state or characteristic, we can select specific parts of an element using pseudo-elements.

Note: The pseudo-elements are specified like the pseudo-classes but with double colon (::), instead of single colon (:). This is part of the CSS3 specification. Older specifications used to have single colon for both pseudo-classes and pseudo-elements.

::first-line

It is used to add special style to the first line of a text. Let's see the following HTML page:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>First Line Pseudo Element</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css"/>
7      </head>
8
9      <body>
10
11         <p>India, officially the Republic of India (Bhārat Gaṇarājya),[18][19][c] is \
12 a country in South Asia. It is the seventh-largest country by area,
13     the second-most populous country with over 1.2 billion people, and the most\
14 populous democracy in the world. India is a federal constitutional
15     republic governed under a parliamentary system consisting of 29 states and \
16 7 union territories. A pluralistic, multilingual, and multi-ethnic society,
17     the country is also home to a diversity of wildlife in a variety of protect\
18 ed habitats. Bounded by the Indian Ocean on the south, the Arabian Sea on the
19     south-west, and the Bay of Bengal on the south-east, it shares land borders\
20 with Pakistan to the west;[d] China, Nepal, and Bhutan to the north-east;
21     and Myanmar (Burma) and Bangladesh to the east. In the Indian Ocean, India \
22 is in the vicinity of Sri Lanka and the Maldives; in addition, India's
23     Andaman and Nicobar Islands share a maritime border with Thailand and Indon\
24 esia.</p>
25
26         <p>
27             Home to the ancient Indus Valley Civilisation and a region of historic trad\
28 e routes and vast empires, the Indian subcontinent was identified with its
29     commercial and cultural wealth for much of its long history.[20] Four major\
30 religions—Hinduism, Buddhism, Jainism, and Sikhism—originated here, whereas
31     Zoroastrianism and the Abrahamic religions of Judaism, Christianity, and Is\
32 lam arrived in the first millennium CE and also shaped the region's diverse cultu\
33 re.
```

```
34     Gradually brought under the administration of the British East India Company  
35     from the early 18th century and administered directly by the United Kingdom after  
36     the Indian Rebellion of 1857, India became an independent nation in 1947 after  
37     a struggle for independence led by Mahatma Gandhi that was marked by non-violent  
38     resistance. Upon the promulgation of its constitution, India became a republic  
39     on 26 January 1950.  
40     </p>  
41  
42     </body>  
43  
44     </html>
```

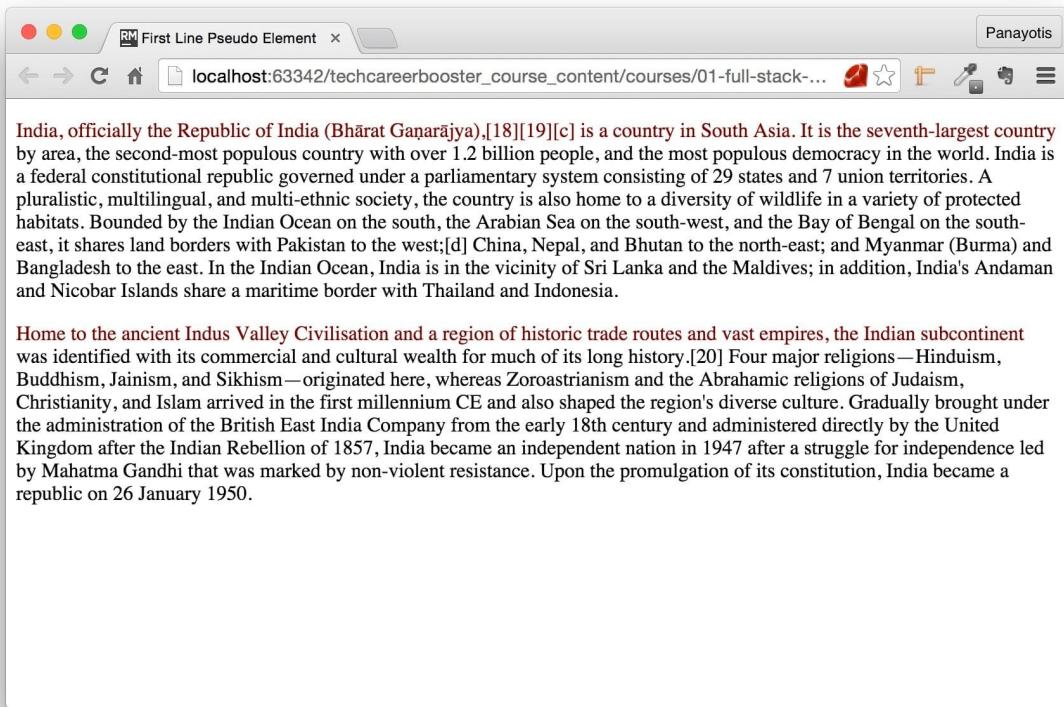
(the above code snippet online)

If you apply the CSS rules here:

```
1 p::first-line {  
2     color: Maroon;  
3 }
```

(the above code snippet online)

and load the page on your browser, you will see this:



First Line Of Paragraph with its own style

Do you see that the first line has been styled with its own color? This is thanks to the `::first-line` pseudo-element.

Note that the `::first-line` pseudo-element is only applied to block elements. Moreover, you cannot set every style property. The properties that can be set are:

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

`::first-letter`

The `::first-letter` pseudo-element is used to style the first letter of a text.

Let's see the following HTML code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>First Letter Pseudo Element</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css"/>
7      </head>
8
9      <body>
10
11         <p>India, officially the Republic of India (Bhārat Gaṇarājya),[18][19][c] is \
12 a country in South Asia. It is the seventh-largest country by area,
13     the second-most populous country with over 1.2 billion people, and the most\
14 populous democracy in the world. India is a federal constitutional
15     republic governed under a parliamentary system consisting of 29 states and \
16 7 union territories. A pluralistic, multilingual, and multi-ethnic society,
17     the country is also home to a diversity of wildlife in a variety of protect\
18 ed habitats. Bounded by the Indian Ocean on the south, the Arabian Sea on the
19     south-west, and the Bay of Bengal on the south-east, it shares land borders\
20 with Pakistan to the west;[d] China, Nepal, and Bhutan to the north-east;
21     and Myanmar (Burma) and Bangladesh to the east. In the Indian Ocean, India \
22 is in the vicinity of Sri Lanka and the Maldives; in addition, India's
23     Andaman and Nicobar Islands share a maritime border with Thailand and Indon\
24 esia.</p>
25
26         <p>
27             Home to the ancient Indus Valley Civilisation and a region of historic trad\
28 e routes and vast empires, the Indian subcontinent was identified with its
29     commercial and cultural wealth for much of its long history.[20] Four major\
30 religions—Hinduism, Buddhism, Jainism, and Sikhism—originated here, whereas
31     Zoroastrianism and the Abrahamic religions of Judaism, Christianity, and Isl\
32 am arrived in the first millennium CE and also shaped the region's diverse cultu\
33 re.
34
35             Gradually brought under the administration of the British East India Compan\
36 y from the early 18th century and administered directly by the United Kingdom aft\
37 er
38
39             the Indian Rebellion of 1857, India became an independent nation in 1947 af\
40 ter a struggle for independence led by Mahatma Gandhi that was marked by non-viol\
41 ent
42
43             resistance. Upon the promulgation of its constitution, India became a repub\
44 lic on 26 January 1950.
45         </p>
46
47     </body>
48 </html>
```

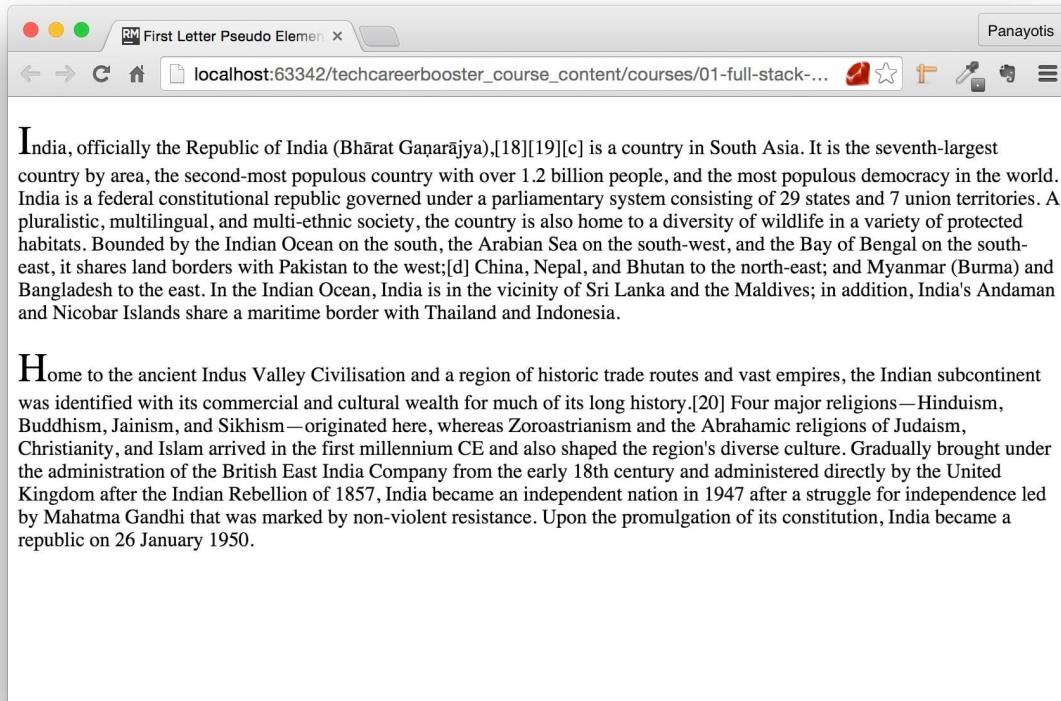
(the above code snippet online)

with the following CSS:

```
1 p::first-letter {  
2     font-size: xx-large;  
3 }
```

(the above code snippet online)

If you save the above files and load the HTML page on your browser, you will see this:



First Letter Of Paragraph with distinct style

Do you see the first letter of each paragraph with its own style? Large font? This has been achieved with the help of the `::first-letter` pseudo-element.

Note that the `::first-letter` pseudo-element is only applied to block elements. Moreover, you cannot set every style property. The properties that can be set are:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties

- text-decoration
- vertical-align (only if “float” is “none”)
- text-transform
- line-height
- float
- clear

`::selection`

This is used to match the portion of the page content that has been selected by the user.

Look at the following HTML page:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>First Letter Pseudo Element</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css"/>
7      </head>
8
9      <body>
10
11         <p>India, officially the Republic of India (Bhārat Gaṇarājya),[18][19][c] is \
12 a country in South Asia. It is the seventh-largest country by area,
13 the second-most populous country with over 1.2 billion people, and the most\
14 populous democracy in the world. India is a federal constitutional
15 republic governed under a parliamentary system consisting of 29 states and \
16 7 union territories. A pluralistic, multilingual, and multi-ethnic society,
17 the country is also home to a diversity of wildlife in a variety of protect\
18 ed habitats. Bounded by the Indian Ocean on the south, the Arabian Sea on the
19 south-west, and the Bay of Bengal on the south-east, it shares land borders\
20 with Pakistan to the west;[d] China, Nepal, and Bhutan to the north-east;
21 and Myanmar (Burma) and Bangladesh to the east. In the Indian Ocean, India \
22 is in the vicinity of Sri Lanka and the Maldives; in addition, India's
23 Andaman and Nicobar Islands share a maritime border with Thailand and Indon\
24 esia.</p>
25
26         <p>
27             Home to the ancient Indus Valley Civilisation and a region of historic trad\
28 e routes and vast empires, the Indian subcontinent was identified with its
29 commercial and cultural wealth for much of its long history.[20] Four major\
30 religions—Hinduism, Buddhism, Jainism, and Sikhism—originated here, whereas
31 Zoroastrianism and the Abrahamic religions of Judaism, Christianity, and Is\
32 lam arrived in the first millennium CE and also shaped the region's diverse cultu\
33 re.
34             Gradually brought under the administration of the British East India Compan\

```

```
35 y from the early 18th century and administered directly by the United Kingdom aft\
36 er
37     the Indian Rebellion of 1857, India became an independent nation in 1947 af\
38 ter a struggle for independence led by Mahatma Gandhi that was marked by non-viol\
39 ent
40     resistance. Upon the promulgation of its constitution, India became a repub\
41 lic on 26 January 1950.
42     </p>
43
44 </body>
45 </html>
```

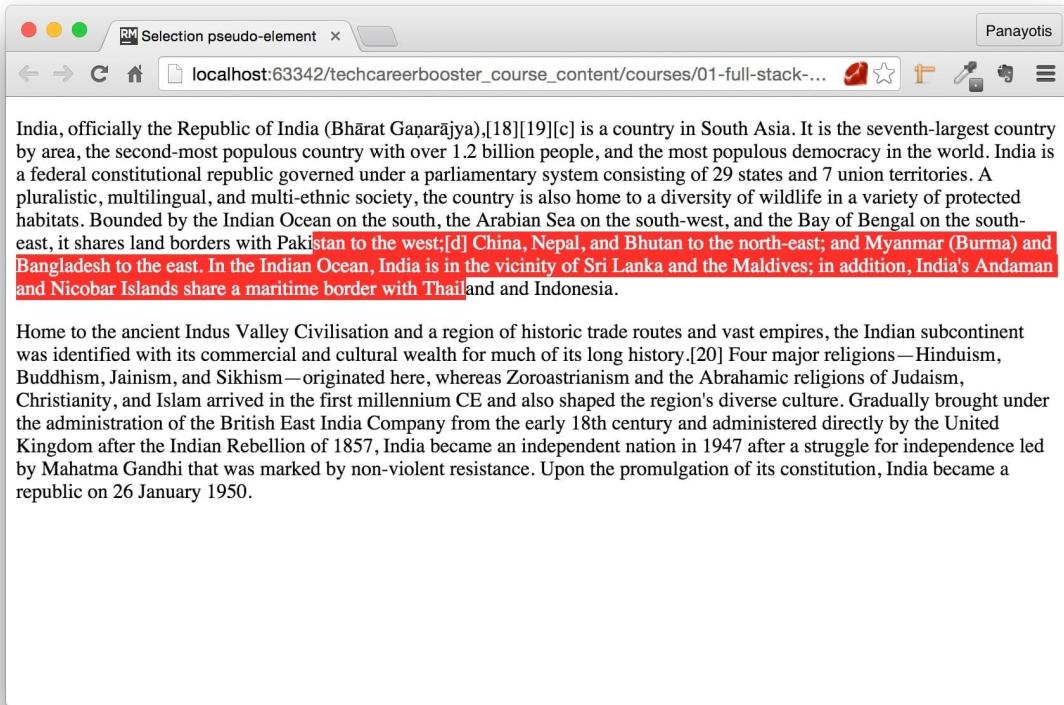
(the above code snippet online)

with this CSS here:

```
1 ::selection {
2     background-color: red;
3     color: white;
4 }
```

(the above code snippet online)

Save the files and load the page on your browser. If you select part of the text, you will see the background color to be red and the color of text to be white.



Selected part Styled with ::selection pseudo-element

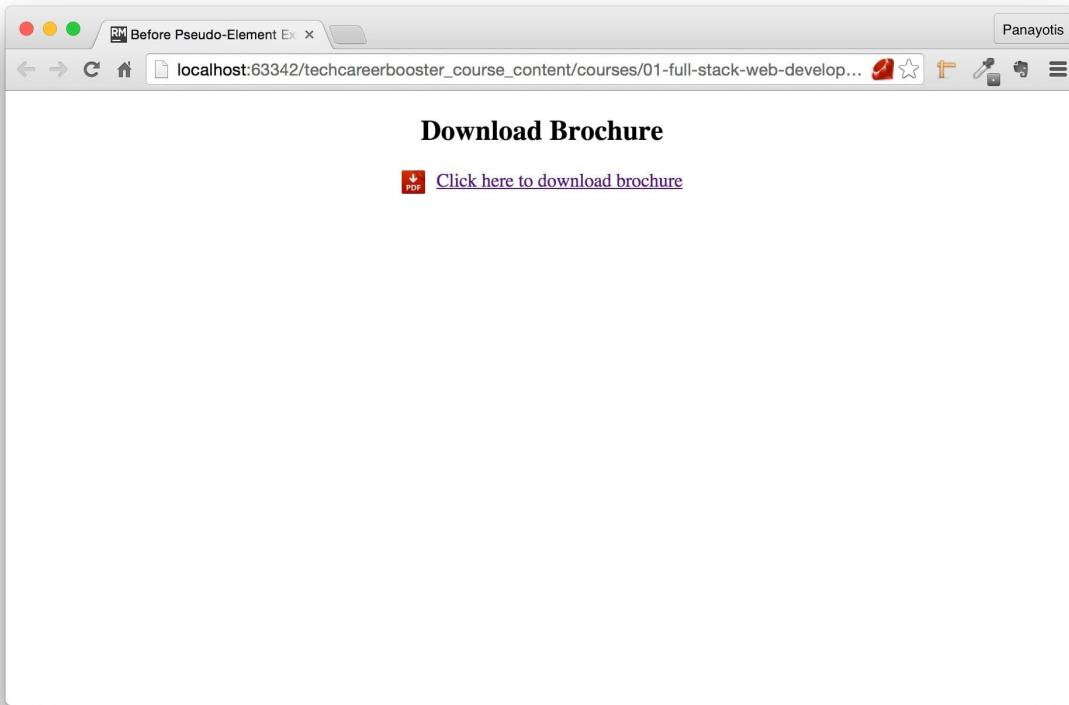
Note that not all CSS properties can be used in ::selection pseudo-element. The ones that can be used are:

- color
- background
- cursor
- outline

::before and ::after

These pseudo-elements are used to add content exactly before or after, respectively, the content of the element they are attached to.

Let's see the following page:



Icon Left to the Anchor Content

As you can see above, there is an icon left to the anchor content. Let's see how we can implement this. Here is the HTML code:

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <title>Before Pseudo-Element Example</title>
7      <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
8  </head>
9
10 <body>
11     <h2>Download Brochure</h2>
12     <a href="https://manuals.info.apple.com/MANUALS/1000/MA1565/en_US/iphone_user\_
13 _guide.pdf">Click here to download brochure</a>
14 </body>
15
16 </html>
```

(the above code snippet online)

This is very simple, as you can see. And if you read the content carefully, you will see that there is no `img` element left to the phrase "Click here to download brochure".

Let's add the following CSS:

```

1 body {
2     text-align: center;
3 }
4
5 a::before {
6     content: "";
7
8     display: inline-block;
9     height: 20px;
10    width: 20px;
11    margin-right: 10px;
12    vertical-align: middle;
13
14    background-image: url(..../images/pdf-download.png);
15    background-repeat: no-repeat;
16    background-attachment: scroll;
17    background-position: center;
18    background-size: 100% 100%;
19 }
```

(the above code snippet online)

Here, we are telling the browser to create an `inline-block` element dynamically and put it before the content of the anchor. The selection is done with `a::before`. The `::before` rules need to specify the content that they will be appending to the existing element content. This is done with the `content` property. In our example, we have the `content` being an empty string. This means that we do not want to actually append anything that has content. It will be a block with 20 x 20 dimensions and it will have a background image. For the background image, we are using the properties that we already know.

What are the advantages of this approach. There is only one real advantage. We do not have to repeat an `img` element for every anchor element in our page. This means that we can quickly add a second and a third and even more download link elements, pointing to PDF documents. They will all automatically go to have a PDF icon left to them, that the user can click to carry out the download.

Can we achieve the same feature using other methods? Yes we can. But we need to know this technique too, because it is found in many Web sites and applications nowadays.

Please, take into account the following about the `::before` and `::after` pseudo-elements:

1. The `content` property is mandatory. It can be a string, it can be a `url` to an image (e.g. `content: url(..../images/foo.png)`). It can be a reference to an attribute of the element attached. E.g. `content: attr(href)` would attach the value of the attribute `href` at the beginning (if `::before`) or at the end of the content (if `::after`).
2. The content that is dynamically attached is not visible in the source code of the HTML page.

3. The content attached is by default an inline element. But you can use the `display` property to turn it to a block element or an inline-block element.
4. Don't put critical information in such elements. Firstly, they are not accessible by assistive technologies, because they are not part of the HTML source page. Also, they may not be supported by all browsers.

Other CSS Selectors

We will close this chapter by listing some of the CSS selectors that we believe you need to be aware of:

- `element > element`.
Example: `div > p`, selects all `p` elements that have a `div` element as parent.
- `element + element`.
Example: `div + p`, selects all `p` elements that are placed immediately after a `div` element.
- `[attribute]`.
Example: `[data-first-name]`, selects all elements that have an attribute with name `data-first-name`.
- `[attribute=value]` or `[attribute="value"]` or `[attribute='value']`.
Example: `[type="submit"]`, selects all elements that have the attribute `type` with value `submit`.
Another example is this `input[type="submit"]` in which we select all the `input` elements with attribute `type` having the value `submit`.
- `[attribute~=value]` or `[attribute~="value"]` or `[attribute~='value']`.
Example `a[href~="www.google.com"]` which selects all the anchor `a` elements that have an `href` attribute that includes the word `www.google.com`.
- `[attribute*=value]` or `[attribute*="value"]` or `[attribute*='value']`.
Same as the `~=` operator.
- `[attribute|=value]` or `[attribute|="value"]` or `[attribute|='value']`.
Example `a[href|="https"]` which selects all the anchor `a` elements that have an `href` attribute that starts with `https`.
- `[attribute^=value]` or `[attribute^="value"]` or `[attribute^='value']`.
Same as the `|=` operator.
- `[attribute$=value]` or `[attribute$="value"]` or `[attribute$='value']`.
Example `a[href$="pdf"]` which selects all the anchor `a` elements that have an `href` attribute that ends with `pdf`.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. You need to implement the following HTML page containing a table with countries population

The screenshot shows a web browser window with the title "Task with Advanced Select" and the URL "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-develop...". The browser interface includes standard controls like back, forward, and search, along with a tab labeled "Panayotis". Below the header, the main content area displays a table titled "Global Population". The table has a yellow header row with four columns labeled "Years" (2011, 2012, 2013, 2014). The body of the table contains 10 rows, each representing a country with its name in red and its population in black. The data is as follows:

| Country Name | 2011 | 2012 | 2013 | 2014 |
|----------------|------------|------------|------------|------------|
| Afghanistan | 28,809,167 | 29,726,803 | 30,682,500 | 31,627,506 |
| Bahamas, The | 366,711 | 372,388 | 377,841 | 383,054 |
| Cabo Verde | 495,159 | 500,870 | 507,258 | 513,906 |
| Denmark | 5,570,572 | 5,591,572 | 5,614,932 | 5,639,565 |
| Ecuador | 15,177,280 | 15,419,493 | 15,661,312 | 15,902,916 |
| Faeroe Islands | 48,492 | 48,393 | 48,292 | 48,221 |
| Gabon | 1,577,298 | 1,613,489 | 1,650,351 | 1,687,673 |
| Haiti | 10,144,890 | 10,288,828 | 10,431,249 | 10,572,029 |
| Iceland | 319,014 | 320,716 | 323,764 | 327,589 |
| Jamaica | 2,699,838 | 2,707,805 | 2,714,734 | 2,721,252 |

Table for Task with Advanced CSS Selectors

2. You need to take care of the following:
 1. You need to use the font “Montserrat:400,700” from Google Fonts site.
 2. “Global Population” header is an `h2` header.
 3. You can take the countries and their population from here: [Population, total](#). Use any country that you like, but make sure that you have at least 10 rows of countries in your table.
 4. It may help you to separate your table element to two sections: `thead` and `tbody`
 5. Make sure that the table borders are styled using CSS rules. Think about how you can make the border collapsing to take place.
 6. The border color should be maroon.
 7. The background color of the head rows of the table should be `#FFE4B5`.
 8. The background color of the odd rows of the body of the table should be `#000096`. The color should be white.
 9. The background color of the even rows of the body of the table should be `lightgray`. The color should be black.

10. Make sure that the “Years” inside the first row of the head of the table is aligned exactly as depicted on the image above.
11. The content of the cells with the country names is left aligned.
12. The content of the cells with the country population numbers is right aligned.
13. At the left of each country name, there is a circle, a bullet with color red. You need to make sure that this is added automatically even if you add new rows to the table. Here are some hints for that:
 1. you will need to use a ::before pseudo-element.
 2. the content of the pseudo-element is actually empty.
 3. note that you can create a circle with a bullet shape using a inline-block that has border and background color red and also has border-radius equal to 50%.
 4. the width and height of the block that is the content of the pseudo-element should be 5px.
3. Upload your work on Netlify.

Chapter 29 - HTML5

Summary

HTML5 is now the standard that is supported by all the major browsers. However, HTML5 is a living standard and new features are added quite often. In this chapter we are going to learn about the semantic elements of HTML5, which we should be using in order to tell browser and other developers reading our code what each area of our page is about.

We will implement a page like this:

The screenshot shows a web page titled "Welcome to our e-shop!" with a navigation bar at the top. Below the title, there's a section titled "Products" featuring four items. Each item includes a small image, a title, and a detailed description. To the right, there's a sidebar with a newsletter sign-up form.

- Product 1 :** 
Product 1 : Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Product 2 :** 
Product 2 : Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Product 3 :** 
Product 3 : Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
- Product 4 :** 
Product 4 : Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

At the bottom, there are links for "About", "Join Us", and "Contact Us".

Page that uses HTML5 Semantic Elements

using HTML5 semantic elements.

Then we will talk about HTML5 Multimedia support. We will learn how we can embed video and audio resources inside our HTML pages. You will be requested to build a page like this:

Enjoy our Videos and Music

Videos

Gregor Salto - Para Voz Feat. Curio Capoeira | Official Video | Capoeira Music Video Capoeira (/kəpu'ərə/; Portuguese pronunciation: [kapu'eɾɐ]) is a Brazilian martial art that combines elements of dance.^{[1][2]} Acrobatics^[3] and music, and is sometimes referred to as a game.^{[5][6]} It was developed in Brazil mainly by West African descendants with native Brazilian influences, probably beginning in the 16th century. It is known for quick and complex moves, using mainly power, speed, and leverage for a wide variety of kicks, spins, and highly mobile techniques. The most widely accepted origin of the word capoeira comes from the Tupi words *ka* ("jungle") e *pôer* ("it was"), referring to the areas of low vegetation in the Brazilian interior where fugitive slaves would hide. Practitioners of the art are called capoeiristas.

Forever Tango - A Evaristo Carriego Tango is a dance that has influences from European and African culture.^[5] Dances from the candombe ceremonies of former slave peoples helped shape the modern day Tango. The dance originated in lower-class districts of Buenos Aires and Montevideo. The music derived from the fusion of various forms of music from Europe.^[6] The word "tango" seems to have first been used in connection with the dance in the 1890s, possibly related to the Latin word "tangere" but more likely related to the African slave word "tango" (drum or dance place).

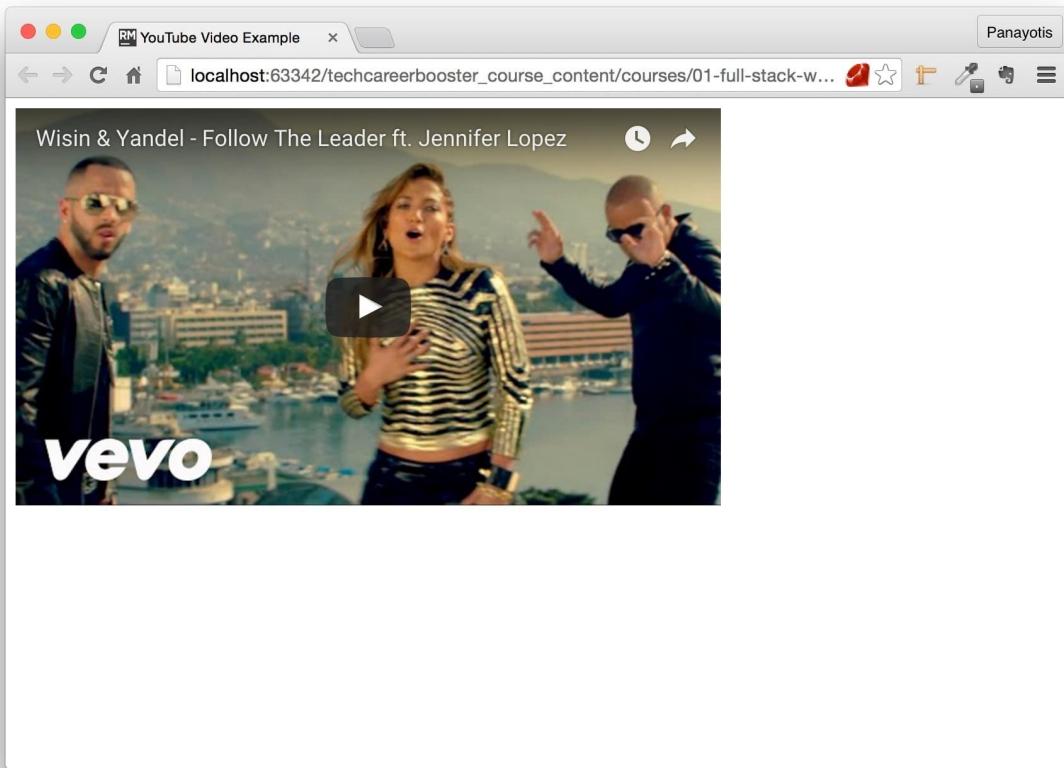
Tchaikovsky - Swan Lake - four little swans Pyotr Ilyich Tchaikovsky (/'pjɔ:tɔ:r i:'li:tʃ tʃaɪkɔfski:/^[1] Russian: Пётр Ильич Чайковский;^[a 1] tr. Pyotr Ilyich Chaikovsky; 25 April/7 May 1840 – 25 October/6 November 1893),^[a 2] often anglicized as Peter Ilyich Tchaikovsky /'pi:tər .../, was a Russian composer of the late-Romantic period, some of whose works are among the most popular music in the classical repertoire. He was the first Russian composer whose music made a lasting impression internationally, bolstered by his appearances as a guest conductor in Europe and the United States. Tchaikovsky was honored in 1884 by Emperor Alexander III, and awarded a lifetime pension.

Carlos Santana -- Black Magic Woman [Official Live Video] HQ Carlos Santana About this sound audio (help·info) (born July 20, 1947) is a Mexican and American musician who first became famous in the late 1960s and early 1970s with his band, Santana, which pioneered a fusion of rock and Latin American

About Us Work Newsletter

Task with Video and Audio Tags

Finally, we will see how we can embed YouTube videos inside our HTML pages:



Example of YouTube video embedded into an HTML page

Learning Goals

1. Learn an overview of the HTML standard history.
2. Learn about the semantic HTML elements, like
 - header
 - footer
 - section
 - nav
 - and more
3. Learn about the HTML5 Multimedia capabilities with `video` and `audio` tags.
4. Learn how to embed YouTube videos inside your HTML pages.

HTML5

HTML has been invented in 1990's. From 1991 till 1998 has gone from version 1 to version 4. HTML 4 is the version that has been widely used until today. In parallel, XHTML has been introduced. It has been recommended by W3C in 2000. The XHTML syntax was strict and the developers were forced to write valid and “well-formed” code. For example, all the elements need to have a closing tag even if they didn't have any content. In 2004, W3C decided to close

the development of HTML (version 4 at that time) in favor of XHTML. Then WHATWG (Web Hypertext Application Technology Working Group) decided to further develop HTML but keep it compatible with older versions of it. Hence in 2008, the first public draft of HTML5 was released.

WHATWG released HTML5 in 2012 as a **living standard**. Living standard means that the specification is continuously under the improvement process. New features are added but old ones are not removed. In 2014, W3C, which had announced their support to WHATWG, released the HTML5 recommendation. [Here is the link to that recommendation](#), just in case you want to have a look.

Hence, today, we are talking about HTML5 which is backwards compatible with HTML4 but includes a lot of new features.

We have already used a lot of the HTML5 features since we started this course, but we really didn't mention that explicitly. In this chapter we are going to talk about HTML5 explicitly and discuss about its main new features that it brings when compared to HTML4.

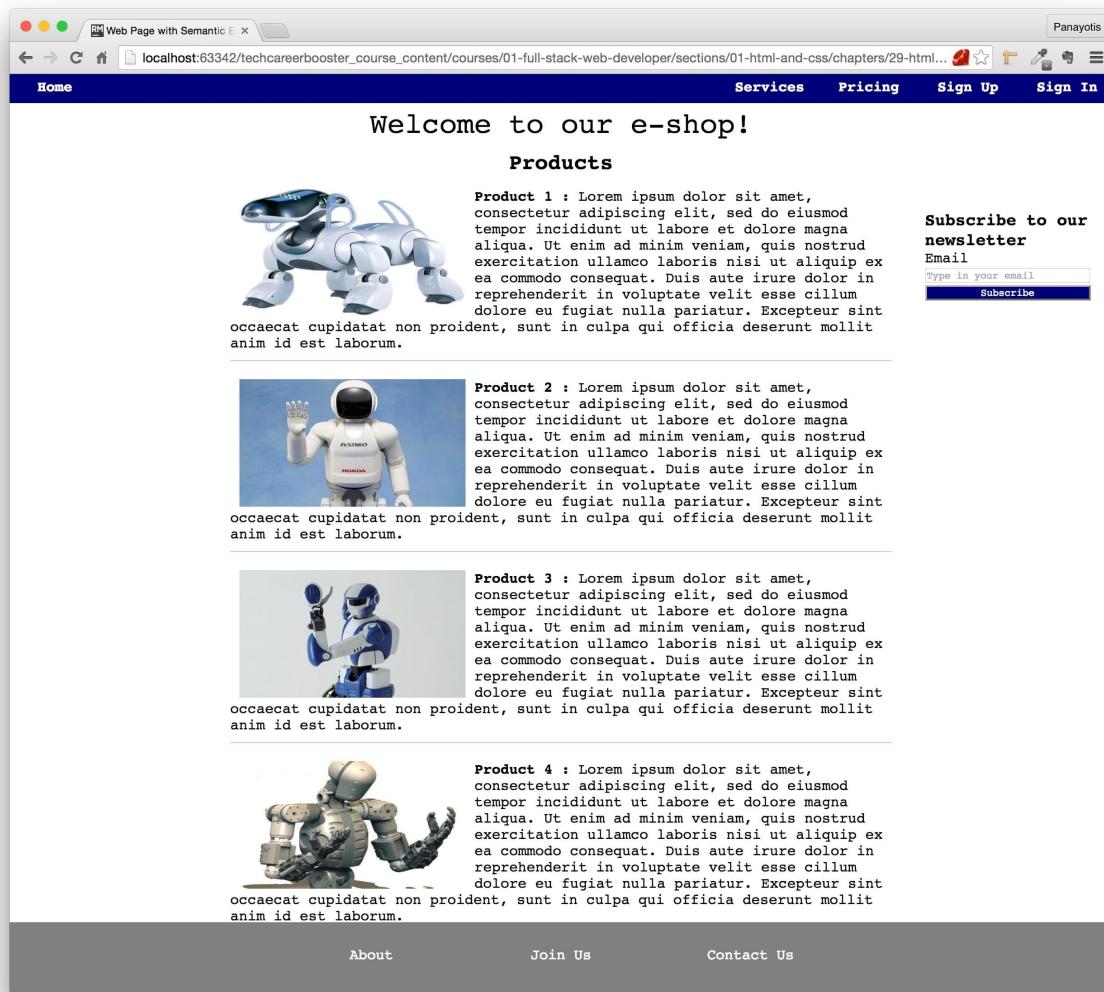
Semantic Elements

HTML5 defines a series of HTML elements that have semantic names. They are block-level elements and they should be used instead of just divs when appropriate. Semantic elements are here to make sure that when one reads the HTML code can easier understand what the elements' content is supposed to be about.

Here are some of the semantic HTML5 elements:

1. header
2. section
3. footer
4. aside
5. nav
6. article
7. main
8. figure

Here is an example HTML page that builds its layout using some of the above HTML5 elements:



Page that uses HTML5 Semantic Elements

There is no visual element on this page that would tell us it is an HTML5 page. If you want to tell whether this page is HTML5 or not, you would have to look into the source code of it.

Let's try to build the above HTML page using HTML5 elements.

Here are the properties of this page that we need to pay attention to:

1. It uses a font family fetched from Google Fonts. It is “Source Code Pro” which is a monospace font family.
2. The top navigation bar stays on fixed position, even if we scroll the content of the page.
3. The top navigation bar items change color to yellow when the mouse hovers over them.
4. The right-hand side, with the newsletter subscription form, stays on the same position even if we scroll the content.
5. The bottom navigation bar stays on fixed position, even if we scroll the content of the page.
6. The bottom navigation bar items change color to black when we hover over them.

You can watch this short video that demonstrates the above:

Usage of the HTML5 page”>Usage of the HTML5 page

Please, note again, that visually, the HTML5 page, that uses semantic elements like the above, does not differ to anything if compared to a page that uses HTML4 simple `div` elements. The difference is that the actual HTML markup with semantic elements is more self-descriptive and tries to describe its meaning both to the browser and to the developers that read the code.

Having said that, let's see how the above can be constructed with HTML5 semantic elements:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Web Page with Semantic Elements</title>
6          <link href='https://fonts.googleapis.com/css?family=Source+Code+Pro:400,500,6\00,700,900,300,200' rel='stylesheet' type='text/css'>
7          <link rel="stylesheet" href="stylesheets/main.css" type="text/css"/>
8      </head>
9
10
11     <body>
12         <nav>
13             <ul class="pull-left">
14                 <li><a href="#">Home</a></li>
15             </ul>
16
17             <ul class="pull-right">
18                 <li><a href="#">Services</a></li>
19                 <li><a href="#">Pricing</a></li>
20                 <li><a href="#">Sign Up</a></li>
21                 <li><a href="#">Sign In</a></li>
22             </ul>
23         </nav>
24
25         <header>
26             Welcome to our e-shop!
27         </header>
28
29         <section>
30             <h2>Products</h2>
31             <ul>
32
33                 <li>
34                      <strong>Product 1 :</strong> Lorem ipsum dolor sit amet, consec\etur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco labo\ris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehender\
```

```
39 it in
40     voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur \
41 sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit a\
42 nim id est laborum.
43     </li>
44
45     <li>
46          <strong>Product 2 :</strong> Lorem ipsum dolor sit amet, consec\
48 etur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
49         aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco labo\
50 ris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehender\
51 it in
52     voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur \
53 sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit a\
54 nim id est laborum.
55
56     </li>
57
58     <li>
59          <strong>Product 3 :</strong> Lorem ipsum dolor sit amet, consec\
61 etur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
62         aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco labo\
63 ris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehender\
64 it in
65     voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur \
66 sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit a\
67 nim id est laborum.
68     </li>
69
70     <li>
71          <strong>Product 4 :</strong> Lorem ipsum dolor sit amet, consec\
73 etur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
74         aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco labo\
75 ris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehender\
76 it in
77     voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur \
78 sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit a\
79 nim id est laborum.
80     </li>
81     </ul>
82     </section>
83
84     <aside>
```

```

85      <h3>Subscribe to our newsletter</h3>
86      <form>
87          <div class="input-group">
88              <label for="newsletter-email">Email</label>
89              <input type="email" id="newsletter-email" placeholder="Type in your ema\
90      il"/>
91          </div>
92          <div class="input-group">
93              <input type="submit" value="Subscribe"/>
94          </div>
95      </form>
96  </aside>
97
98  <footer>
99      <ul>
100         <li><a href="#">About</a></li>
101         <li><a href="#">Join Us</a></li>
102         <li><a href="#">Contact Us</a></li>
103     </ul>
104  </footer>
105 </body>
106 </html>

```

(the above code snippet online)

As you can see, the code above is very simple and basically, does not have anything new to you, other than the semantic elements. Instead of the semantic elements you could have equally used `div` elements.

Let's pay some attention to the following:

1. The top navigation bar is composed of two `ul` elements. It is very common to create navigation bar items lists using `ul` elements. It may seem to be a little bit unintuitive to you at the beginning, because you are used to have lists being with bullets and usually vertical. But we will show how we can convert them to nice menu items lists. Note that the first list is composed of the Home menu item only. This is because we want the left side of the bar to have only the Home item. The second list contains the rest of the items.
2. The main content of the page starts with a `header` element.
3. Next, below the `header` element we have the main content of the page, which is the list of products. The list of products is contained inside a `section` element.
4. For the list of products we are using, again, an unsorted list, a `ul` element. This is another example to demonstrate that lists (`ol` and `ul`) do not necessarily have to be the boring top down lists, with the simple bullets.
5. The subscription to newsletter part of the page is included inside an `aside` element. Works like a side note, something that is not related to the actual content of the particular page. This element contains a form with elements that we have already learned about in the forms chapter.

6. The bottom navigation bar is a `footer` element. We use this semantic element because this part of the page stays at the bottom as a page footer. Again, here, we are using a `ul` element. With similar to top navigation bar techniques, we convert this list to something that can work as a list of menu items. We will explain, later on, how we have managed to do that.

Let's start with a very basic `main.css` file:

```
1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7  html {
8      font-size: 62.5%;
9  }
10
11 html,
12 input {
13     font-family: "Source Code Pro", monospace;
14 }
15
16 input[type=submit] {
17     -webkit-appearance: button;
18     -moz-appearance: button;
19     -o-appearance: button;
20     -ms-appearance: button;
21     appearance: button;
22 }
23
24 body {
25     font-size: 1.6rem;
26     margin-top: 3.6rem;
27     margin-bottom: 8.0rem;
28 }
29
30 h2 {
31     text-align: center;
32 }
33
34 input[type=email] {
35     width: 100%;
36 }
37
38 input[type=submit] {
39     width: 100%;
```

```
40     background-color: #000078;  
41     color: White;  
42 }
```

(the above code snippet online)

There is nothing new to you on this CSS, except from the fact that you need to precisely tell the browser to treat the `input` of type `submit` as a button:

```
1 input[type=submit] {  
2     -webkit-appearance: button;  
3     -moz-appearance: button;  
4     -o-appearance: button;  
5     -ms-appearance: button;  
6     appearance: button;  
7 }
```

(the above code snippet online)

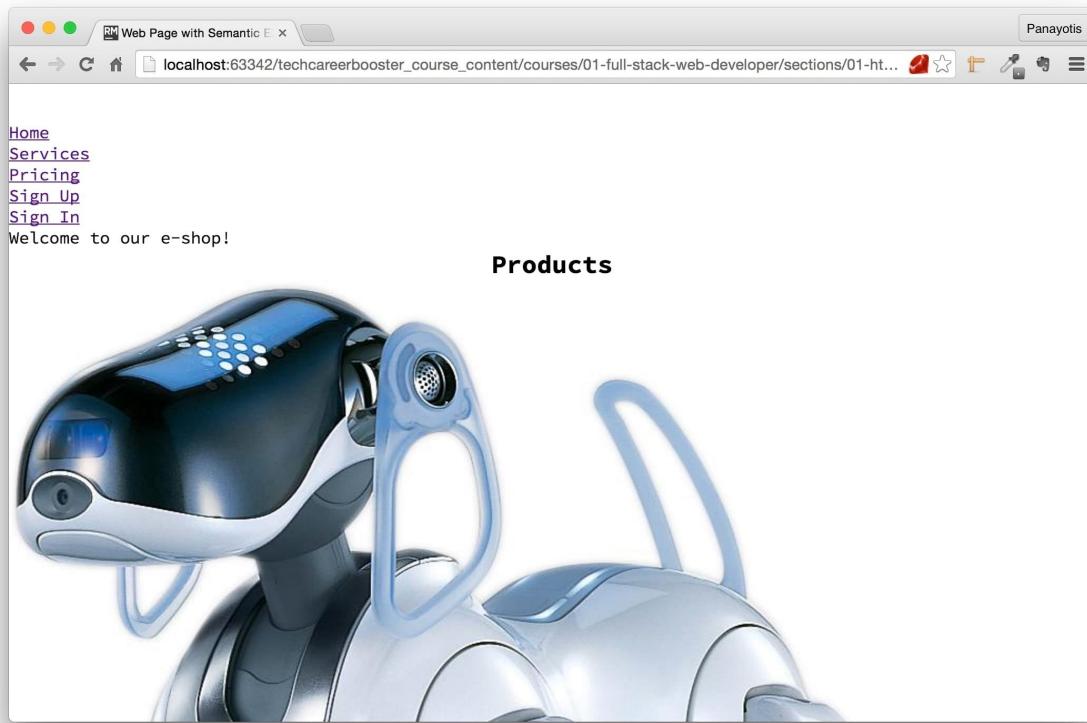
If you do not do that, there are combinations of browsers and operating systems that will not display the button with the font family that we want, but they will use a system font.

Note: In the above CSS rules, we are using the so-called vendor-specific prefixes:

1. `-webkit-` for Webkit based browsers (e.g. Chrome, Safari).
2. `-moz-` for Mozilla based browsers (e.g. Firefox).
3. `-o-` for Opera based browsers.
4. `-ms-` for Microsoft based browsers.

and finally, we are specifying a generic name for the same property. This is because this property might not be supported under the same name on all versions and brands of popular browsers.

If you save the above files and load the HTML page on your browser, then it will not be displaying the content properly:



Page Content is Not Displayed Correctly

Let's fix that. We will start with the navigation bar. We need to add the following CSS rules:

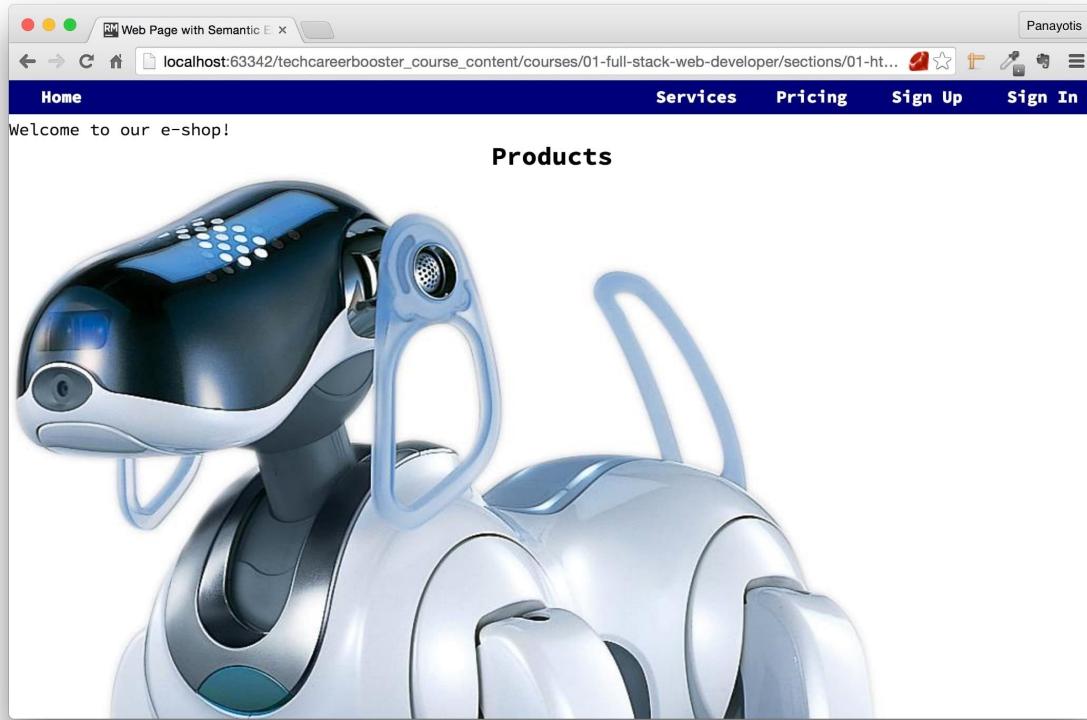
```
1 .pull-right {
2     float: right;
3 }
4
5 .pull-left {
6     float: left;
7 }
8
9 nav {
10    background-color: #000078;
11    color: white;
12    position: fixed;
13    top: 0;
14    left: 0;
15    width: 100%;
16    height: 3.2rem;
17 }
18
19 nav ul {
20     list-style-type: none;
21     padding: 5px 0;
```

```
22 }
23
24 nav ul li {
25   display: inline-block;
26   width: 100px;
27   text-align: center;
28 }
29
30 nav ul li a {
31   text-decoration: none;
32   color: White;
33   font-weight: 700;
34 }
35
36 nav ul li a:hover {
37   color: yellow;
38 }
```

(the above code snippet online)

1. The `pull-left` class is used to float the `Home` menu item to the left.
2. The `pull-right` class is used to float the rest of the menu items on the top navigation bar to the right.
3. The `nav` bar is styled so that it has fixed position.
4. The `nav ul` is styling the list of menu items. The important property here is the `list-style-type`, which takes the value `none`. This removes the default bullets that usually exist next to each `li` content.
5. The `nav ul li` selector makes the `li` items to be `inline-block`. This makes them appear in one line.
6. Then we style the anchors `a` inside each one of the `li` items.

If you save the above file and load the page on your browser, you will see this:



Top Navigation Bar has been styled

As you can see above, the top navigation bar has been styled correctly. Also, if you move your mouse over the menu items, you will see that their color changes to yellow.

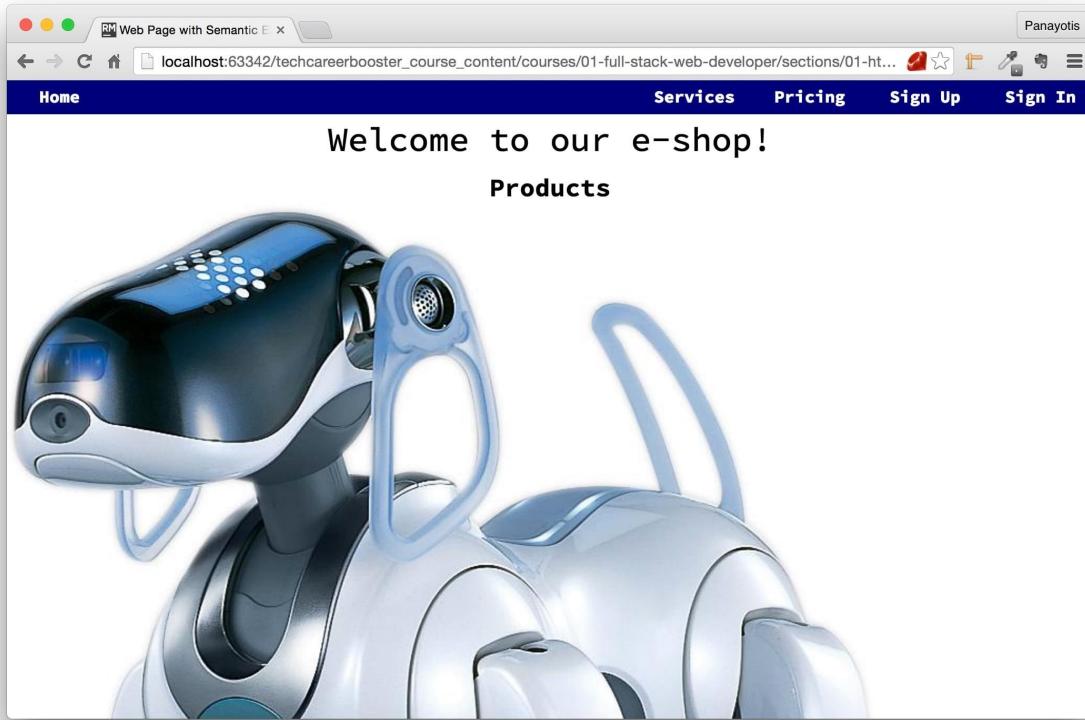
Let's now, quickly, fix the header content. Add the following to your CSS file:

```
1 header {  
2     text-align: center;  
3     margin-bottom: 1.0rem;  
4     font-size: 3.2rem;  
5     font-weight: 500;  
6 }
```

(the above code snippet online)

Pretty straightforward here. Look how we add the `font-weight`. We are using one of the weight integers that come together with our font-family.

If you save your CSS file and reload your page, you will see this:



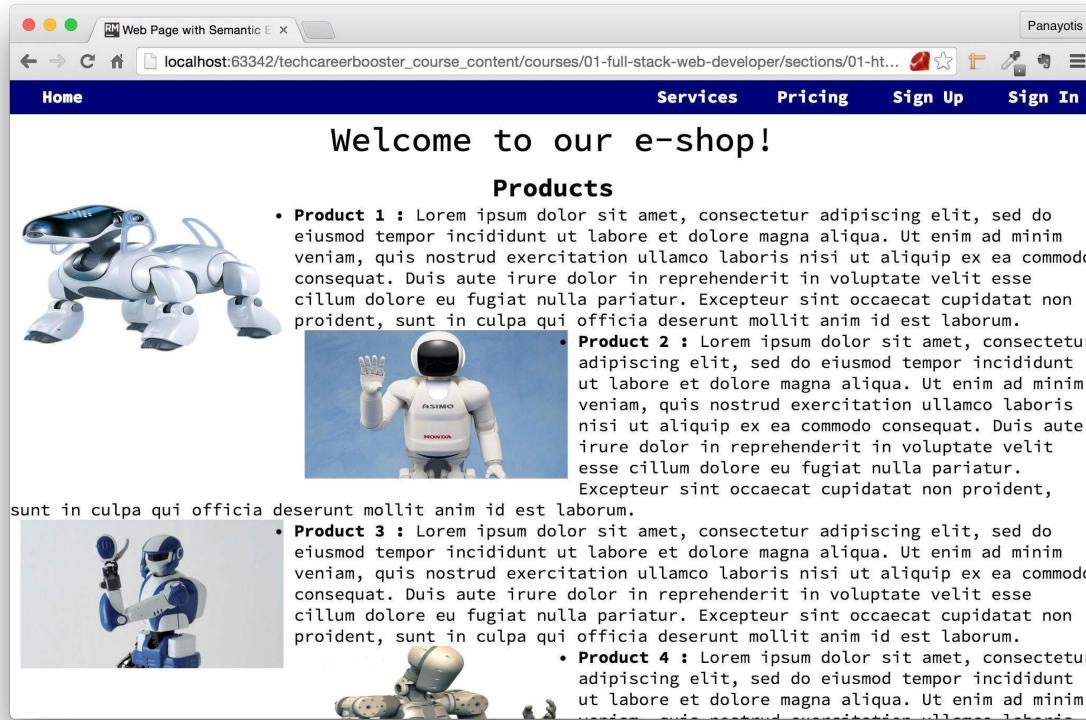
Header is styled correctly

Let's now take care of the section with the list of products. We will first make sure that the images are floated to the left, so that the text goes around it. Also, we make all the images of equal size. Let's do that:

```
1 .product-image {  
2     width: 250px;  
3     height: 141px;  
4     float: left;  
5     margin: 0 10px;  
6 }
```

(the above code snippet online)

If you amend your CSS file with the above rules and reload your page, you will see this:



Images Float Left and Same Size

As you can see above, the images float left and the text goes around each image. Also, all images are of the same size. Let's style the list of products now:

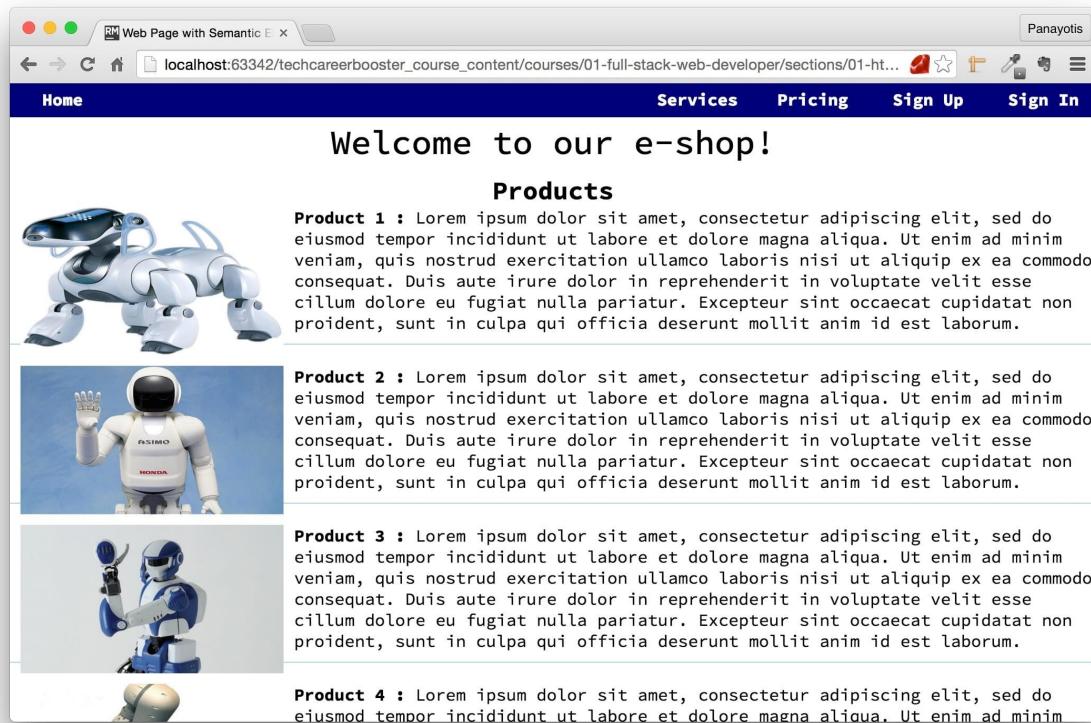
```

1 section ul {
2   list-style-type: none;
3 }
4
5 section ul li {
6   margin-bottom: 20px;
7   padding-bottom: 10px;
8   border-bottom: 1px solid lightblue;
9   min-height: 150px;
10}

```

(the above code snippet online)

As you can see above, we remove the default style type from the list (`list-style-type: none;`) and we also add some border bottom on each one of the list items. Moreover, we make sure that the `li` height is at least 150px, a little bit higher than the image height, to make sure that the image does not overflow the `li` height. If you save the above and reload your HTML page you will see this:



List of Products with Correct Style

As you can see above, the list of products seems to be ready. What is missing, is some margins to the left and right of the section with products. Let's do that too:

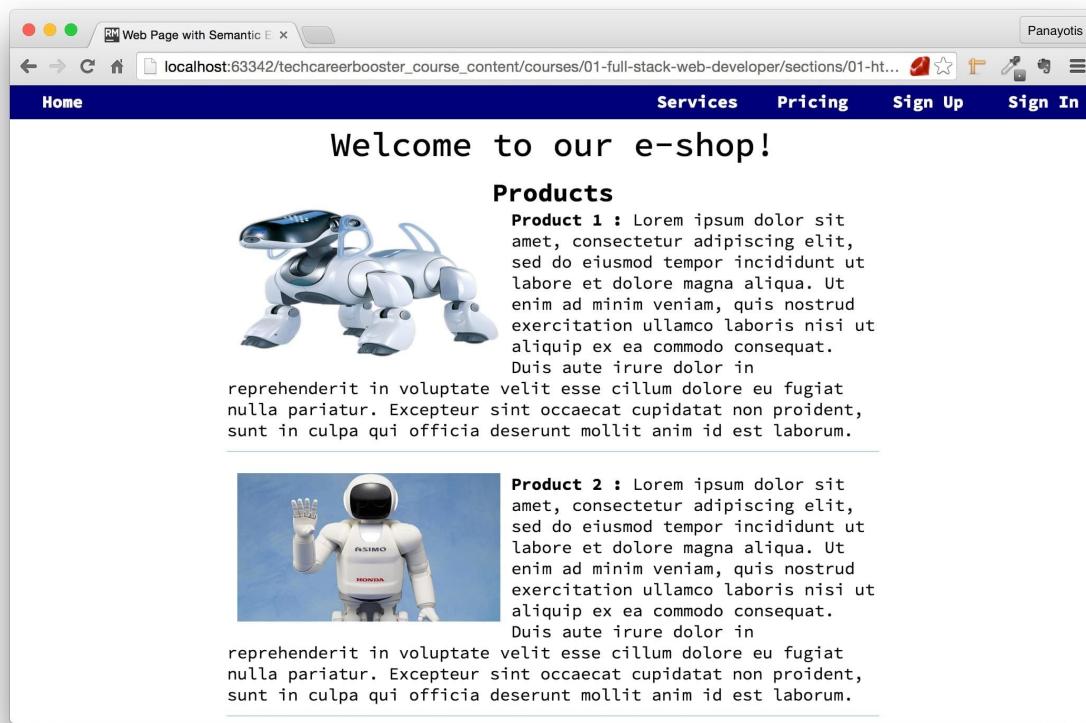
```

1  section {
2      width: 60%;
3      margin: 0 auto;
4 }
```

(the above code snippet online)

We limit the width of the section to 60% of the width of the browser window. We also make sure that the section appears in the center by setting the left and right margins to auto.

If you add the above to your CSS file and reload the page you will see this:

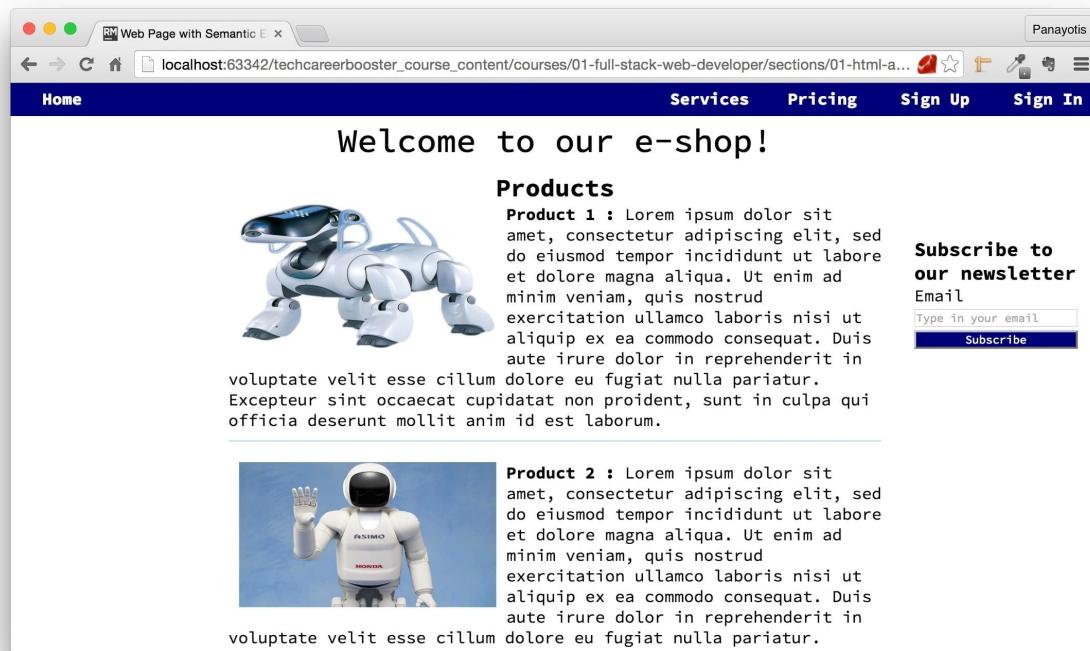


Section with products in center with margins left and right

Perfect. We now move to the other part of the page that has the newsletter subscription. As we said earlier, we want this to always be on a fixed position in relation to browser edges. Add the following CSS code to your CSS file:

```
1 aside {  
2     width: 15%;  
3     position: fixed;  
4     top: 150px;  
5     right: 2%;  
6 }
```

(the above code snippet online)



Subscription Form On Top Right Fixed Position

What remains to be styled is the bottom navigation bar, the footer. Its style is similar to the top navigation bar. Add the following to your CSS file:

```

1  footer {
2      position: fixed;
3      bottom: 0;
4      left: 0;
5      width: 100%;
6      height: 8.0rem;
7      background-color: gray;
8      color: white;
9      text-align: center;
10 }
11
12 footer ul {
13     list-style-type: none;
14     width: 60%;
15     margin: 2.7rem auto;
16 }
17
18 footer ul li {
19     display: inline-block;
20     width: 200px;
21 }
22

```

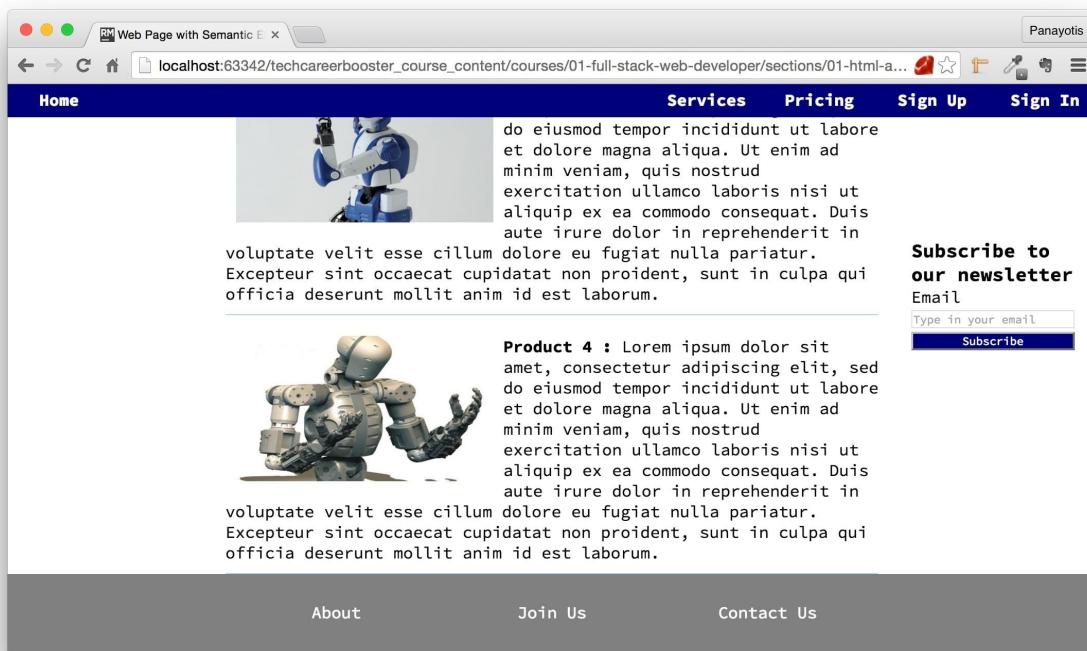
```

23 footer ul li a {
24     text-decoration: none;
25     color: White;
26 }
27
28 footer ul li a:hover {
29     color: Black;
30 }

```

(the above code snippet online)

If you amend your file and reload the page you will see this:



Footer is styled correctly

The page looks quite ready. There are some small CSS touches that will make it look better. For example, the margins below the headers. The final, complete CSS file should be like that:

```

1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7  html {
8      font-size: 62.5%;
9  }

```

```
10
11 html,
12 input {
13   font-family: "Source Code Pro", monospace;
14 }
15
16 input[type=submit] {
17   -webkit-appearance: button;
18   -moz-appearance: button;
19   -ms-appearance: button;
20   -o-appearance: button;
21   appearance: button;
22 }
23
24 body {
25   font-size: 1.6rem;
26   margin-top: 3.6rem;
27   margin-bottom: 8.0rem;
28 }
29
30 /* Utility classes */
31
32 .pull-right {
33   float: right;
34 }
35
36 .pull-left {
37   float: left;
38 }
39
40 /* general elements */
41 h2 {
42   text-align: center;
43 }
44
45 /* Navigation Bar Styling */
46
47 nav {
48   background-color: #000078;
49   color: white;
50   position: fixed;
51   top: 0;
52   left: 0;
53   width: 100%;
54   height: 3.2rem;
55 }
```

```
56
57 nav ul {
58   list-style-type: none;
59   padding: 5px 0;
60 }
61
62 nav ul li {
63   display: inline-block;
64   width: 100px;
65   text-align: center;
66 }
67
68 /* LVHA */
69
70 nav ul li a {
71   text-decoration: none;
72   color: White;
73   font-weight: 700;
74 }
75
76 nav ul li a:hover {
77   color: yellow;
78 }
79
80 /* --- end of navigation bar styling --- */
81
82 /* Header */
83 header {
84   text-align: center;
85   margin-bottom: 1.0rem;
86   font-size: 3.2rem;
87   font-weight: 500;
88 }
89
90 section {
91   width: 60%;
92   margin: 0 auto;
93 }
94
95 aside {
96   width: 15%;
97   position: fixed;
98   top: 150px;
99   right: 2%;
100 }
101
```

```
102 /* section with products */
103
104 section h2 {
105     margin-bottom: 15px;
106 }
107
108 section ul {
109     list-style-type: none;
110 }
111
112 section ul li {
113     margin-bottom: 20px;
114     padding-bottom: 10px;
115     border-bottom: 1px solid lightblue;
116     min-height: 150px;
117 }
118
119 .product-image {
120     width: 250px;
121     height: 141px;
122     float: left;
123     margin: 0 10px;
124 }
125
126 input[type=email] {
127     width: 100%;
128 }
129
130 input[type=submit] {
131     width: 100%;
132     background-color: #000078;
133     color: White;
134 }
135
136 /* footer */
137 footer {
138     position: fixed;
139     bottom: 0;
140     left: 0;
141     width: 100%;
142     height: 8.0rem;
143     background-color: gray;
144     color: white;
145     text-align: center;
146 }
147
```

```
148 footer ul {  
149     list-style-type: none;  
150     width: 60%;  
151     margin: 2.7rem auto;  
152 }  
153  
154 footer ul li {  
155     display: inline-block;  
156     width: 200px;  
157 }  
158  
159 footer ul li a {  
160     text-decoration: none;  
161     color: White;  
162 }  
163  
164 footer ul li a:hover {  
165     color: Black;  
166 }
```

(the above code snippet online)

If you save this file and load your page you will see the final version of the page.

This finishes our encounter with HTML5 semantic elements. We are proceeding to other aspects of HTML5 that you need to be aware of.

HTML5 Graphics

You can use two techniques to draw graphics with HTML5. The `svg` and the `canvas`. We will not deal with that now. These are advanced topics and we will talk about them in the JavaScript chapters.

HTML5 Multimedia

HTML5 defines some HTML elements that have a special use to display multimedia content: `video` and `audio`.

`video`

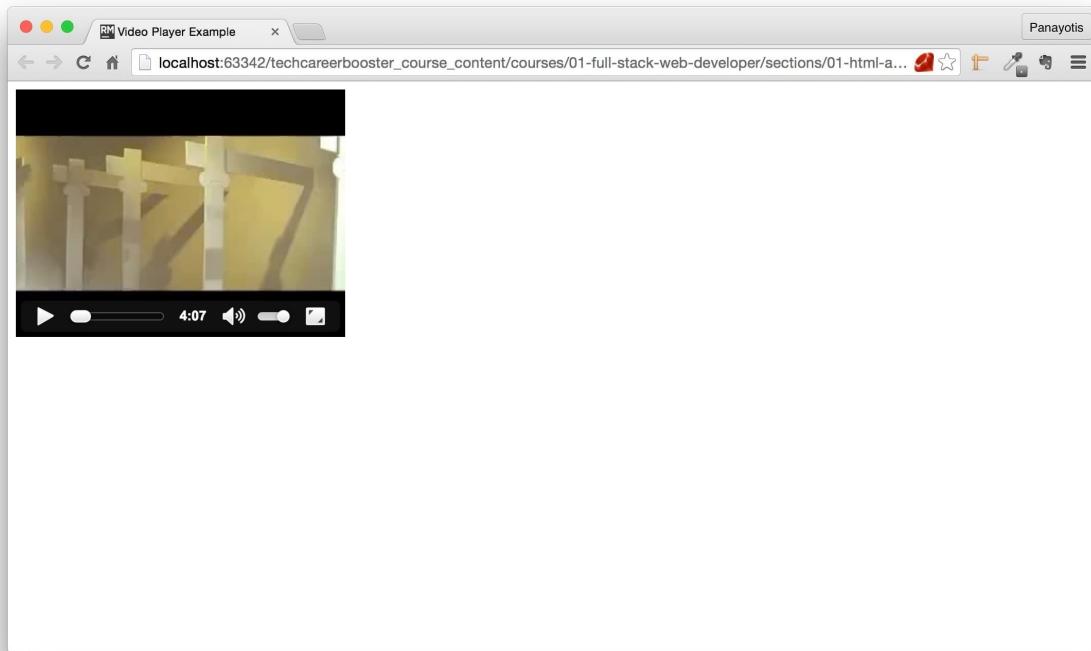
Before HTML5 there was not any standard way to show videos on a web page. The HTML5 `video` element specifies a standard way to embed a video in a web page.

See the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Video Player Example</title>
6      </head>
7
8      <body>
9          <video width="320" height="240" controls="controls">
10             <source src="http://tech-career-booster-courses.s3.amazonaws.com/01-full-stack-web-developer/sections/01-html-and-css/chapters/29-html5/assets/videos/animal\>
11 s-house-of-the-rising-sun.mp4"/>
12             Your browser does not support the video tag.
13         </video>
14     </body>
15 </html>
```

(the above code snippet online)

Save the above code as an HTML page and load it on your browser. You will see something like this:



Video Element Example

The video will start playing if you click on the play button. Pretty simple. Isn't it?

Here are some details that have to do with the `video` element:

1. the `controls="controls"` adds play, pause, resume e.t.c. play control buttons.

2. always include `width` and `height` attributes so that you control how the video player is drawn on your page.
3. `video` is an inline element.
4. Text that exists between the opening and closing tags, will appear only if the browser does not support the `video` tag.
5. You can include multiple times the `source` element. Usually we do that if we have the same video in different digital formats, e.g. AVI, MP4. The browser will pick up the first that will be able to play back. So, extra `source` elements pointing to different formats of the same video content work like a fallback source, in case the first source format is not supported.

audio

Like for video, before HTML5 there was not any standard way to display an audio player inside an HTML page and allow the user to listen to an audio file. With HTML5, the `<audio>` element can be used to give user access to audio files.

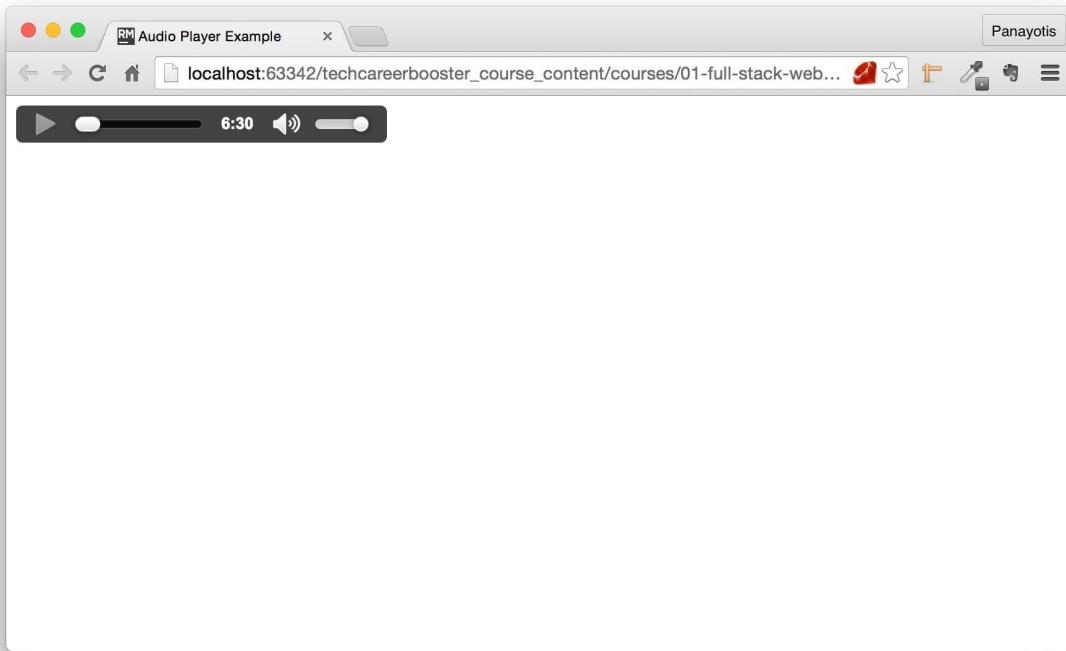
Let's see the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Audio Player Example</title>
6      </head>
7
8      <body>
9          <audio controls="controls">
10             <source src="http://tech-career-booster-courses.s3.amazonaws.com/01-full-st\
11 ack-web-developer/sections/01-html-and-css/chapters/29-html5/assets/audio/eagles-\
12 hotel-california-with-lyrics.mp3"/>
13             This browser does not support the audio HTML element.
14         </audio>
15     </body>
16 </html>
```

(the above code snippet online)

As you can see above, the `<audio>` tag is similar to `<video>`. It is very simple. You have to define the source of the audio. Again the `controls` attribute adds play controls on the player displayed inside the HTML page.

If you save the above page and then load it on your browser, you will see this:



A simple audio player as displayed by Chrome

As you can see the audio player has play controls and it also displays the duration of the audio play back.

Common to video and audio

There are some properties that are common to both `video` and `audio` tags. Here they are:

- `autoplay`. This is a boolean attribute. If it exists it will instruct browser to start playing back the video or audio as soon as it can. Here is how you should be using that:

```
1 <video controls="controls" autoplay="autoplay">
2   <source src="link-to-video.mp3">
3 </video>
```

(the above code snippet online)

Note that `autoplay` is by default switched off.

- `preload`. This is an attribute that controls whether the media will be loaded along side with the rest of the page content or not. It is very important and affects the user experience and performance of your page. It takes the following values:
 - `auto`. The entire media is loaded as soon as the rest of the page loads. This is the heaviest method. It might significantly increase the amount of time required for your page to load. Note that this is the default value. So, if `preload` is not present, then `auto` is assumed and the media loads alongside with the rest of the content of your page.

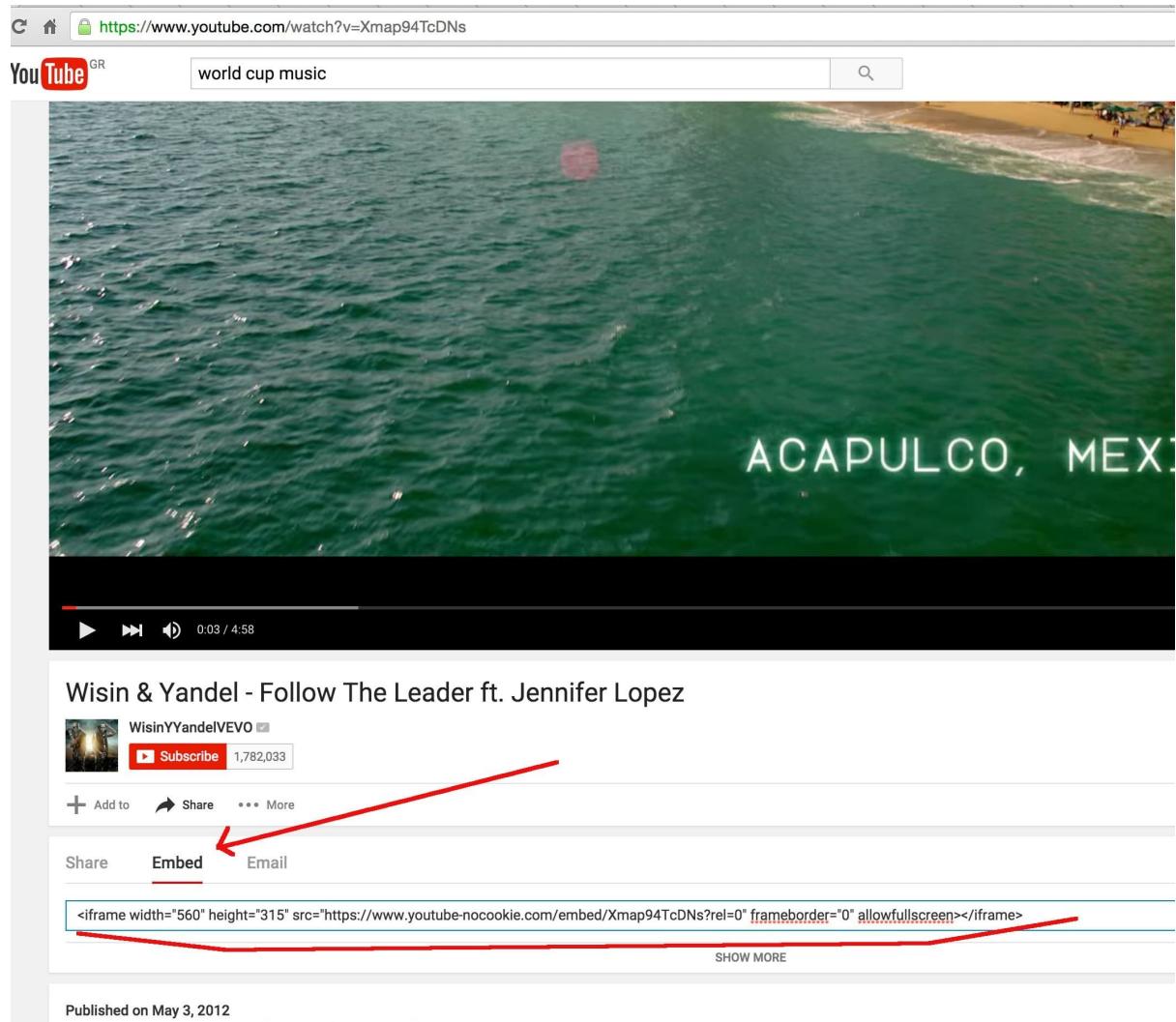
- `none`. None of the media content is preloaded. Neither the content nor any metadata (e.g. its duration).
- `metadata`. This is very useful because it is both quick and allows the browser to display enough information about the media that makes for a nice user experience. It does not download the media content. It downloads the media metadata, e.g. its duration.

Important: If your page requires a lot of media to be displayed at the same time, i.e. has a lot of audio or video tags, the best preloading method that you should use is `metadata`. If you do not use that, due to limitations in number of requests and downloads that a page can do at the same time, some of your media might not be accessible. This also means that you need to be careful with the `autoplay` property too. Since it basically preloads the content as if you had `preload="auto"`.

YouTube Videos

You may be wondering whether you can attach a YouTube video source to your HTML page using the `video` tag. The answer is no, because you do not actually have the real source, web address, of the video resource.

If you want to embed YouTube videos on your page, you will need to follow the “embed” instructions on the “share part” of the video YouTube page.



Embedding YouTube Videos

Above, you can see a screenshot from a video page on YouTube. [Visit this page](#) and click on “Share”, then click on “Embed”. You will be presented with an input text box where it has the code that you need to embed into your HTML page code. This will allow you to have this video inside your Web page. Let’s do that:

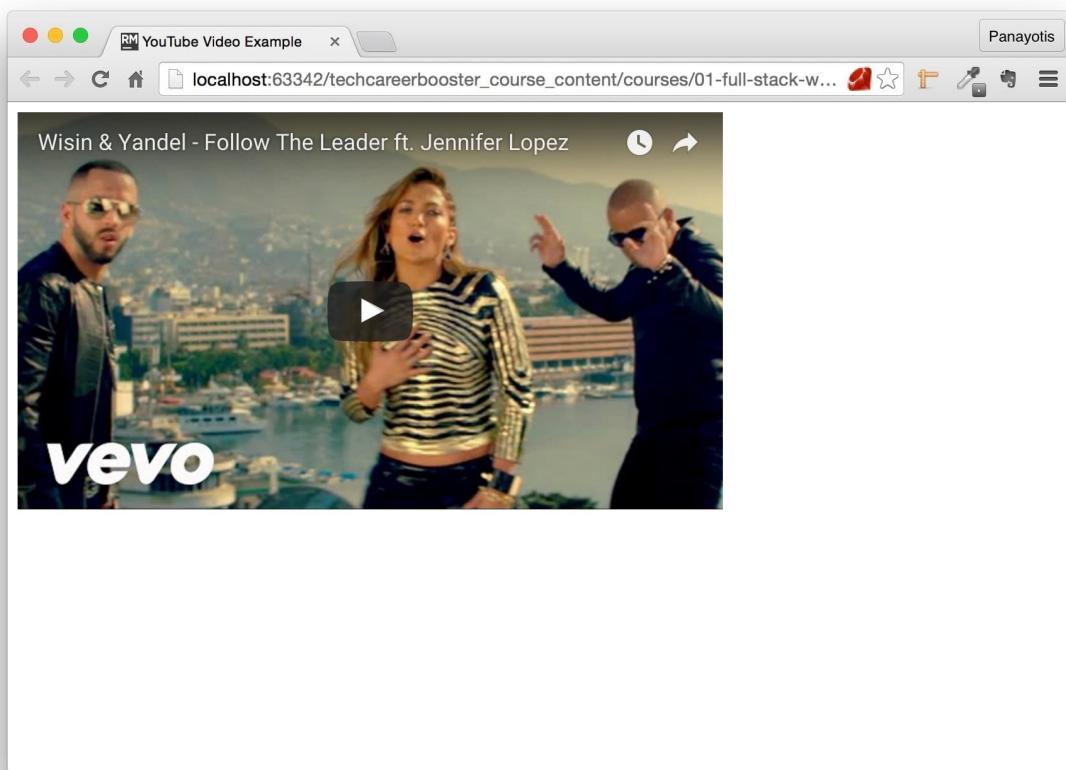
```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>YouTube Video Example</title>
6    </head>
7
8    <body>
9      <iframe width="560" height="315" src="https://www.youtube-nocookie.com/embed/\
10 Xmap94TcDNs?rel=0" frameborder="0" allowfullscreen></iframe>
11  </body>
12 </html>
```

(the above code snippet online)

As you can see above, the body tag contains the code that YouTube is telling us to use. It is an `iframe` element.

Note: for the `iframe` element, see next chapter that has a short reference to some other HTML elements.

Save the above HTML code into an HTML file and load the page on your browser. You will see this:



Example of YouTube video embedded into an HTML page

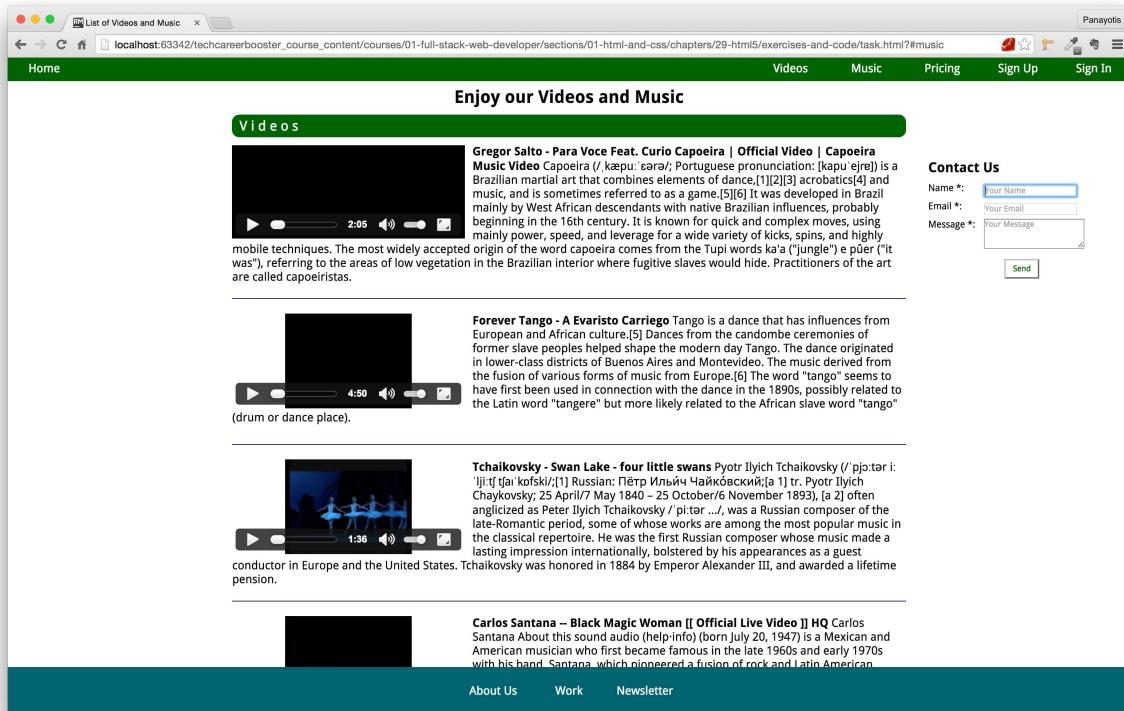
Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. You need to implement the following web page that includes a series of video and audio tags:



Task with Video and Audio Tags

2. Watch this video to see how this page works:

[Video showing the page for the Task with Videos and Audios](#)

3. Here are some requirements and hints to help you finish this task:

1. It uses the font “Droid Sans” from Google Fonts.
2. When the user clicks on “Videos” menu item, the page scrolls at the point where the videos section starts. You many need to read back again the chapter on “Positioning”. It has a tip on how you can solve the problem of content being hidden behind the top navigation bar.
3. The video links that you should use are the following:
 1. [gregor-salto-para-voce-feat-curio-capoeira.mp4](#)
 2. [forever-tango-a-evaristo-carriego.mp4](#)
 3. [tchaikovsky-swan-lake-four-little-swans.mp4](#)
 4. [carlos-santana.mp4](#)
 5. [rachmaninov-rhapsody-on-a-theme-of-paganini_Op-43-variation-18-andantecant.mp4](#)
4. The audio links that you should use are the following:
 1. [The Official 2014 FIFA World Cup Song](#)
 2. [shakira-waka-waka-the-official-2010-fifa-world-cup.mp3](#)
 3. [Shakira - La La La \(Brazil 2014\) ft. Carlinhos Brown.mp3](#)
 4. [Shakira - Loca \(Spanish Version\) ft. El Cata.mp3](#)
 5. [Jennifer Lopez - On The Floor ft. Pitbull.mp3](#)
5. The contact us form on the right

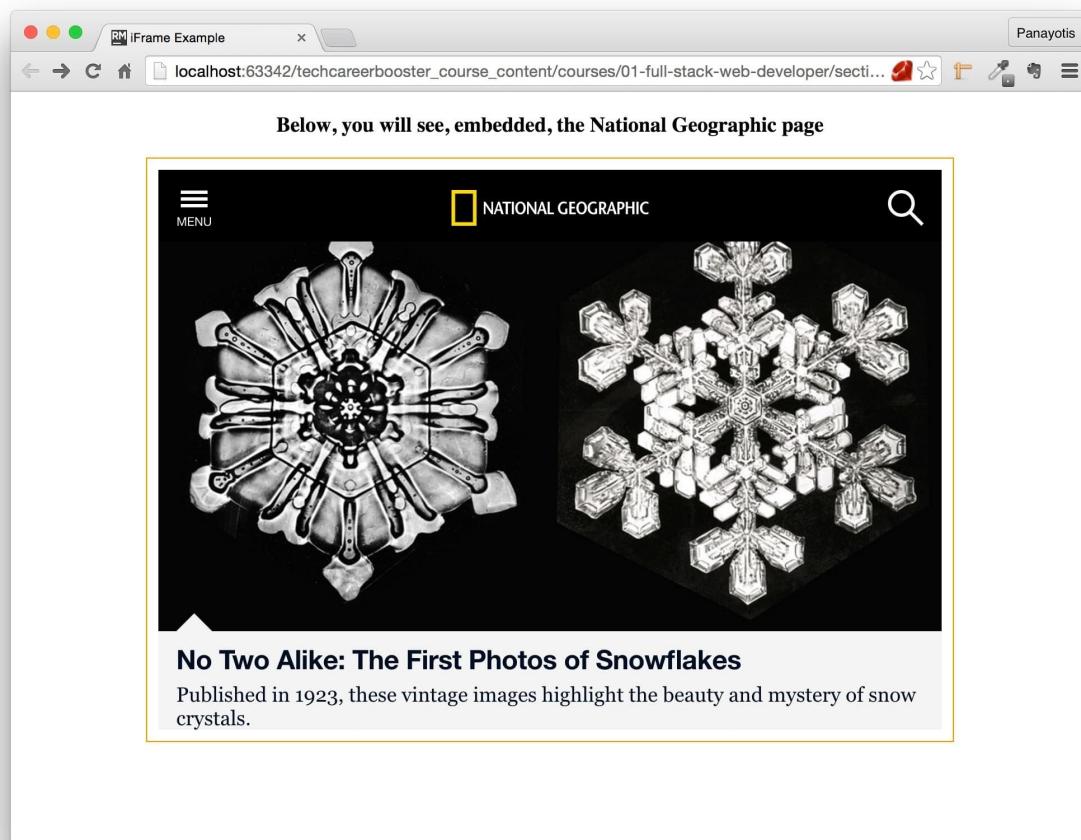
1. Needs to have all the fields required.
2. The focus should be on the Name field when the page loads.
3. The visible rows that user can type inside the message box should be 3.
6. Be careful with the styling:
 1. body element needs to have enough top and bottom margins so that the content is not hidden.
 2. The button of the Contact us form, needs to have the font family “Droid Sans” like the rest of the document.
 3. Choose your own colors, but make sure that
 1. When the mouse moves over any of the menu items, top or bottom, the color changes.
 2. When the mouse moves over the “Send” button on Contact Us form the background color and color of the button change.
 4. You may need to increase the z-index of the top navigation bar. This is because the video player controls UI might be drawn on top of the navigation bar if not done.
4. You will need to upload your work to Netlify.

Chapter 30 - Some Other HTML Elements

Summary

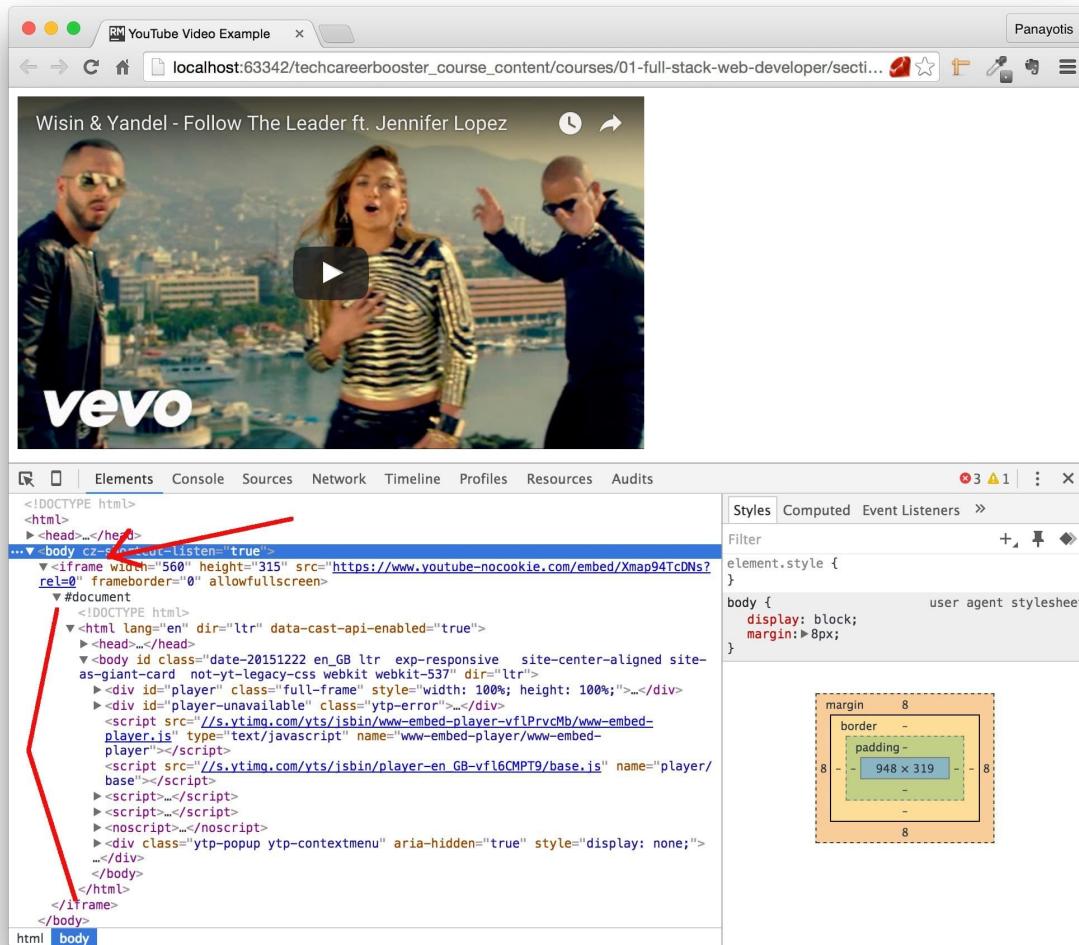
This chapter will give us an overview of some HTML elements that have not been covered in the previous chapters but they will be good to know about. They are quite useful.

We will learn about how we can embed one HTML page inside another.



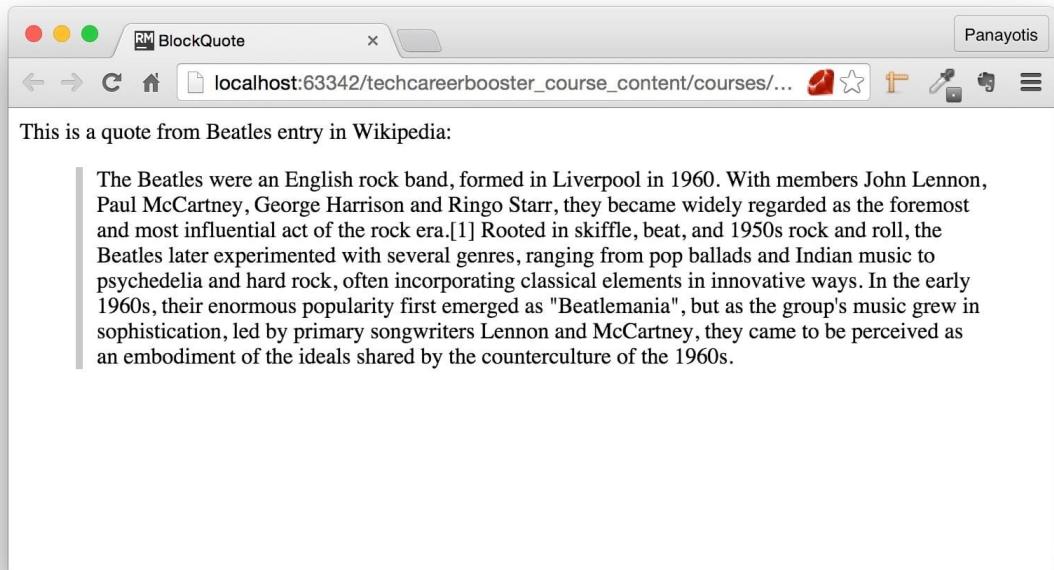
National Geographic Page Embedded Inside our HTML page.

We will learn how the browser is creating another HTML page inside the original HTML source code:



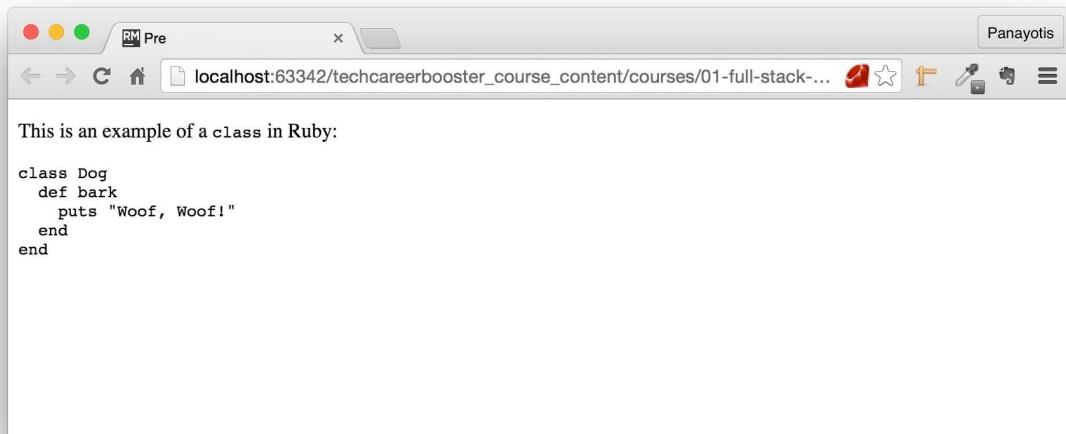
A full HTML content inside another HTML content

Then we will learn how we can quote content inside our HTML page and show that this content originally belongs to another source.



Quoting from another source

Finally, we will learn how we can include programming source code text inside our HTML page:



Programming source code inside our HTML page

You will be requested to build a page like this:

The screenshot shows a web browser window with two tabs open. The left tab is titled "Ruby" and the right tab is titled "Java". Both tabs are from the Wikipedia article on their respective topics.

Ruby

Ruby is a dynamic, reflective, object-oriented, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan.

This is an example source code from this language:

```
class Person
  attr_reader :name, :age
  def initialize(name, age)
    @name, @age = name, age
  end
  def <=>(person) # the comparison operator for sorting
    age <=> person.age
  end
  def to_s
    "#(name) (#(age))"
  end
end
```

Here, you can see the whole reference from Wikipedia:

Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, [12] and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA). [13] meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. [14] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2015, Java is one of the most popular programming languages in use. [15] [16] [17] [18] particularly for client-server web applications, with a reported 9 million developers. [citation needed] Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

This is an example source code from this language:

```
class HelloWorldApp {
  public static void main(String[] args) {
    System.out.println("Hello World!"); // Prints the
  }
}
```

Here, you can see the whole reference from Wikipedia:

Web Page for Task with Other HTML Elements

Learning Goals

1. Learn about the inline frame `iframe` element and how you can embed another HTML page into the current HTML page.
2. Learn how you can use the semantic element `blockquote` to include a text that is copied from another source.
3. Learn how you can have blocks of programming source code inside your HTML content, using:
 1. `code` or
 2. `pre` elements.

`iframe`

The `iframe` stands for `inlinE frame` and it is used to embed into our HTML page the source code of another HTML page. One page inside another.

Let's see this example:

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <title>iFrame Example</title>
7      <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
8  </head>
9
10 <body>
11     <h3>Below, you will see, embedded, the National Geographic page</h3>
12     <div>
13         <iframe src="http://www.nationalgeographic.com"></iframe>
14     </div>
15 </body>
16
17 </html>
```

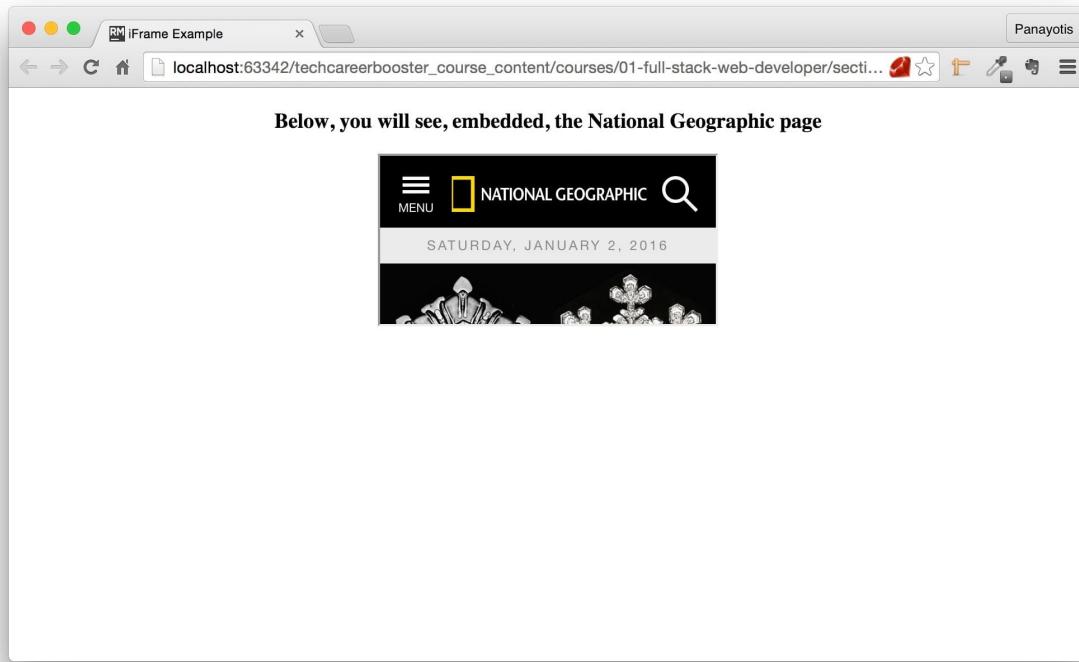
(the above code snippet online)

with CSS:

```
1  body {
2      text-align: center;
3 }
```

(the above code snippet online)

If you save the above two files and load the page on your browser, you will see this:



Embedding the National Geographic Web Site

As you can see above, the browser reserves some area on your page to display the content of another. The content of the other page is fully functional. You can scroll, and click on the links of the embedded page. And the content on the main page does not change. It is the content of the area that shows the embedded page that changes.

Watch this video to understand what we mean:

How the user can use the embedded National Geographic page

The important attribute of the `iframe` tag is the `src`, which takes as value, the URL address of the external Web page to embed.

When we want to style an `iframe` we should use CSS. Let's improve the style of the above example:

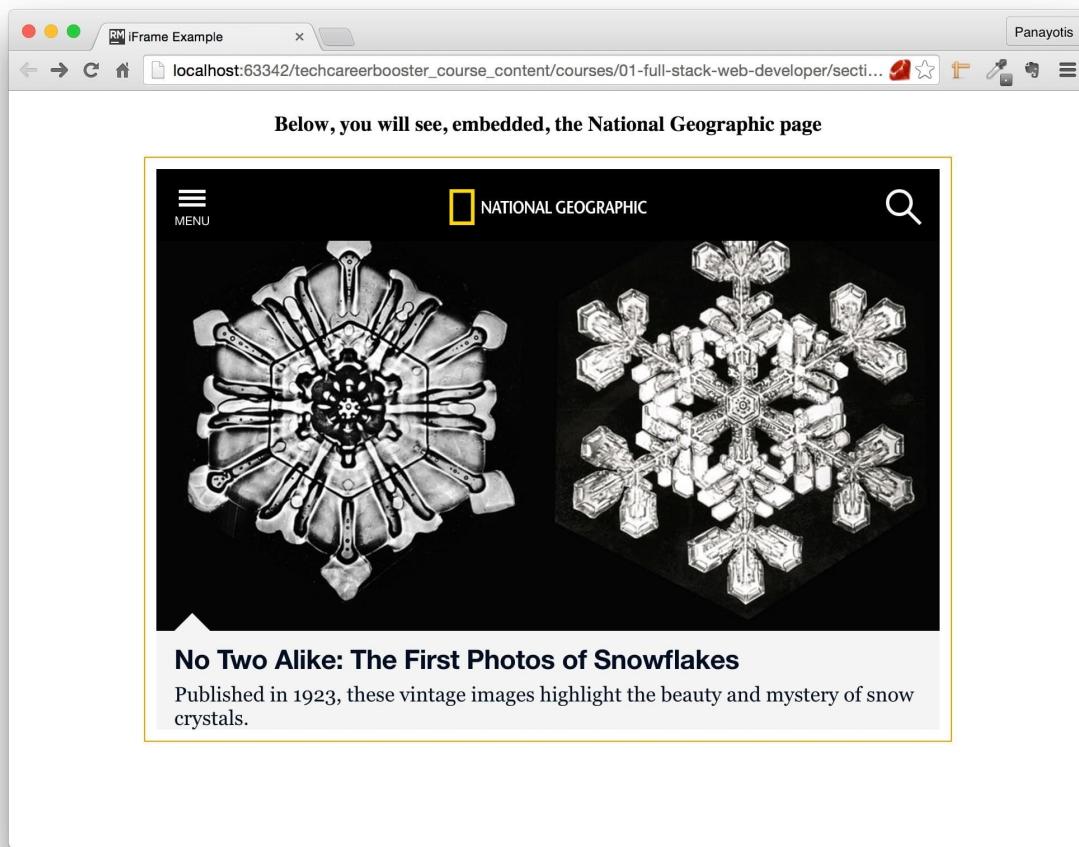
1. We will increase its dimensions
2. We will add some border around it, alongside some padding.

Here is the CSS file:

```
1 body {  
2     text-align: center;  
3 }  
  
4  
5 iframe {  
6     height: 500px;  
7     width: 700px;  
8     border: 1px solid #FF9800;  
9     padding: 10px 10px;  
10}
```

(the above code snippet online)

If you save this file and reload the page, you will see this:



iFrame Styled

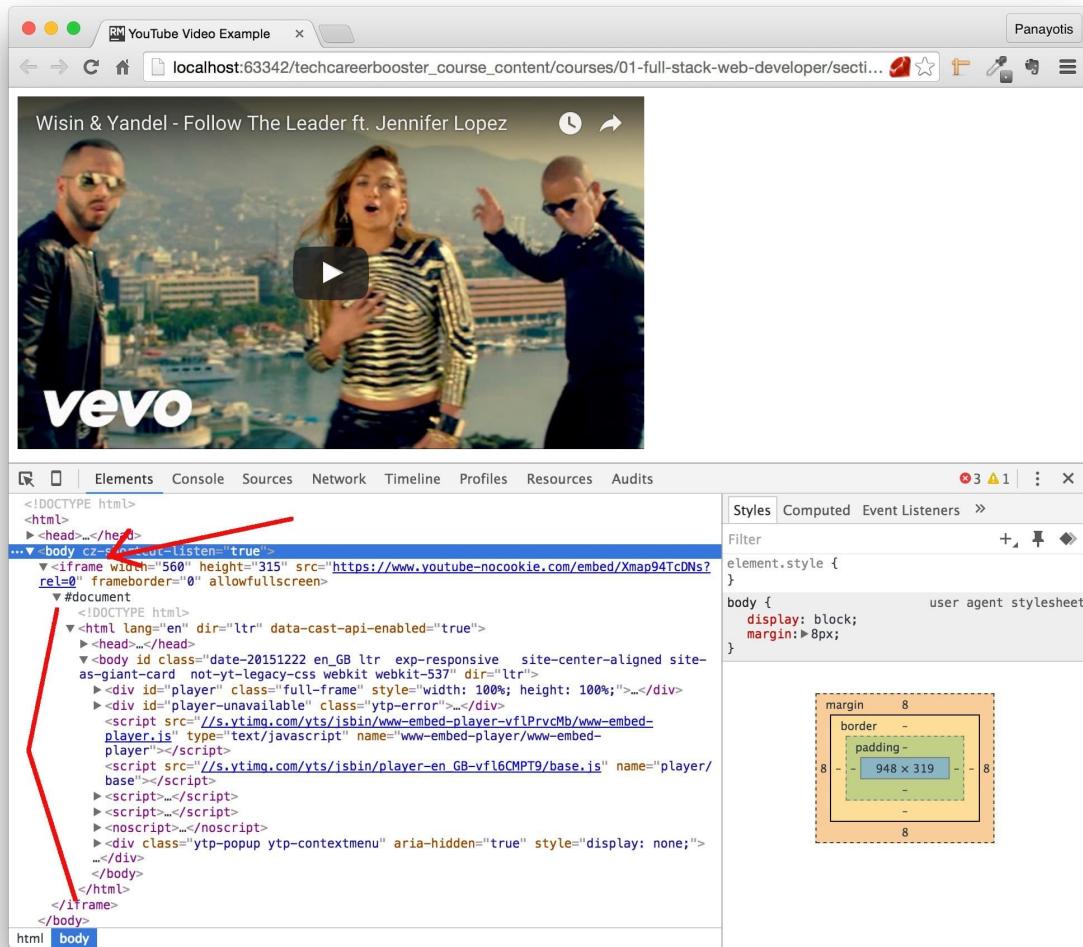
Pretty nice.

Before closing the `iframe` section, we would like to tell you that, usually, when you use an `iframe` your browser creates a whole HTML document inside your current HTML page. Let's take another example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>YouTube Video Example</title>
6      </head>
7
8      <body>
9          <iframe width="560" height="315" src="https://www.youtube-nocookie.com/embed/\
10 Xmap94TcDNs?rel=0" frameborder="0" allowfullscreen></iframe>
11     </body>
12 </html>
```

(the above code snippet online)

It is the page that includes an `iframe` as instructed by YouTube share / embed feature. If you load this page on your browser and open the developer tools to inspect the `iframe` content, you will see this:



iFrame content is an HTML page

As you can see above, browser creates the i frame content as a whole HTML page.

blockquote

The `blockquote` element is used to include a part of HTML content that is quoted from another source. Look at the following example:

```

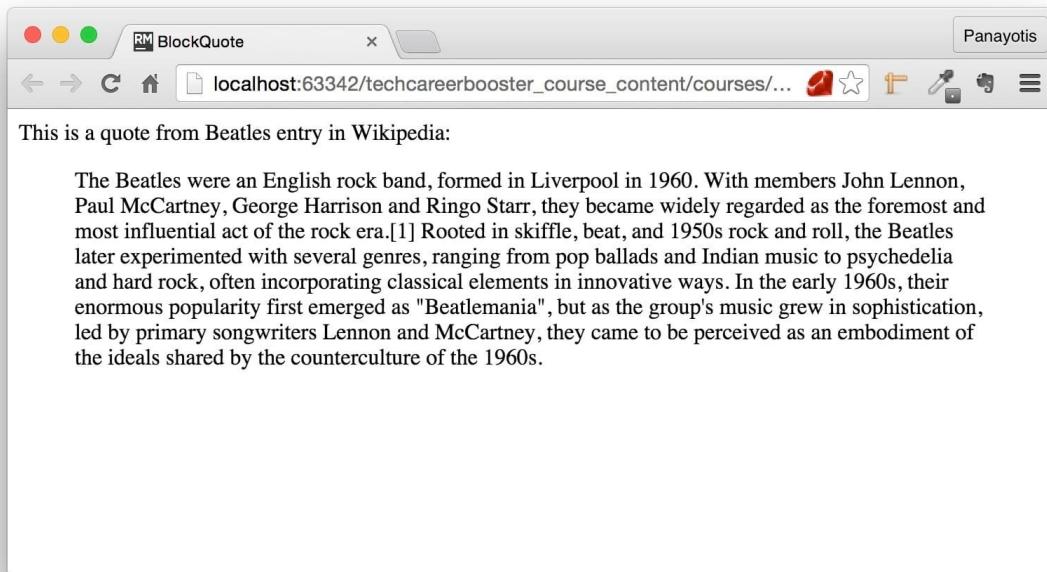
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>BlockQuote</title>
6    </head>
7    <body>
8      This is a quote from Beatles entry in Wikipedia:
9      <blockquote cite="https://en.wikipedia.org/wiki/The_Beatles">
```

```
10      The Beatles were an English rock band, formed in Liverpool in 1960. With me\
11  mbers John Lennon, Paul McCartney, George Harrison and Ringo Starr, they became
12      widely regarded as the foremost and most influential act of the rock era.[1\
13  ] Rooted in skiffle, beat, and 1950s rock and roll, the Beatles later experimente\
14  d
15      with several genres, ranging from pop ballads and Indian music to psychedel\
16  ia and hard rock, often incorporating classical elements in innovative ways.
17      In the early 1960s, their enormous popularity first emerged as "Beatlemania\
18  ", but as the group's music grew in sophistication, led by primary
19      songwriters Lennon and McCartney, they came to be perceived as an embodimen\
20  t of the ideals shared by the counterculture of the 1960s.
21  </blockquote>
22  </body>
23 </html>
```

(the above code snippet online)

Do you see the `blockquote` element? It includes some text that has been copied from Wikipedia, “The Beatles” entry. The actual page that the content was quoted from is given as value to the `cite` attribute of the `blockquote` element.

If you save the above content and load the page on your browser, you will see this:



Blockquote Example

As you can see above, the content of the `blockquote` is right indented. Let's try to style the `blockquote` a little bit:

Write the following CSS code and link to the CSS file, `main.css`, from your HTML page:

```
1 blockquote {  
2     border-left: 5px solid #c8c8c8;  
3     padding-left: 10px;  
4 }
```

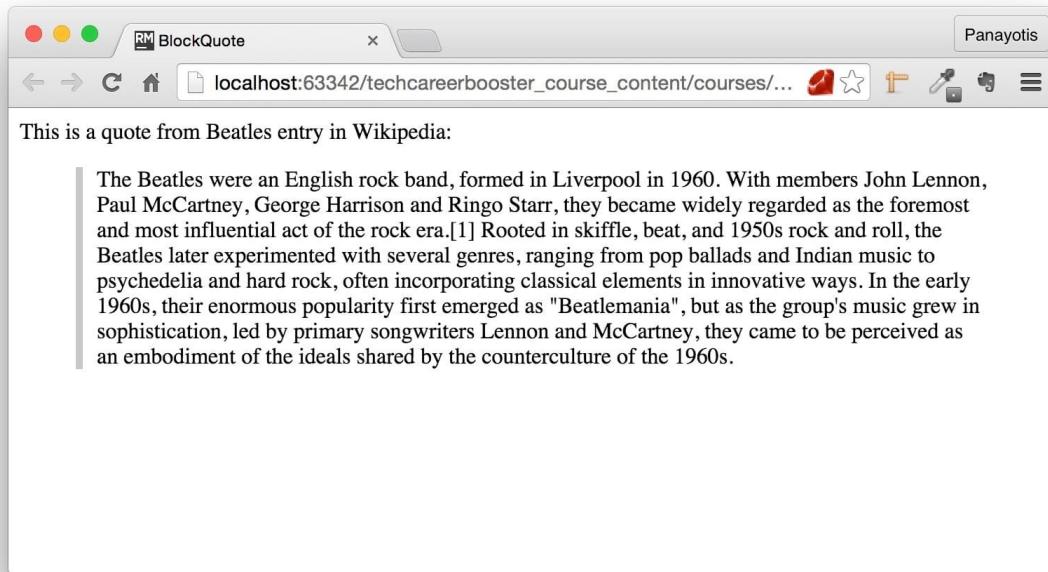
(the above code snippet online)

The HTML content should be:

```
1  <!DOCTYPE html>  
2  <html>  
3      <head>  
4          <meta charset="utf-8">  
5          <title>BlockQuote</title>  
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">  
7      </head>  
8      <body>  
9          This is a quote from Beatles entry in Wikipedia:  
10         <blockquote cite="https://en.wikipedia.org/wiki/The_Beatles">  
11             The Beatles were an English rock band, formed in Liverpool in 1960. With me\  
12             mbers John Lennon, Paul McCartney, George Harrison and Ringo Starr, they became  
13                 widely regarded as the foremost and most influential act of the rock era.[1\  
14             ] Rooted in skiffle, beat, and 1950s rock and roll, the Beatles later experimente\  
15             d  
16                 with several genres, ranging from pop ballads and Indian music to psychedel\  
17             ia and hard rock, often incorporating classical elements in innovative ways.  
18                 In the early 1960s, their enormous popularity first emerged as "Beatlemania\  
19             ", but as the group's music grew in sophistication, led by primary  
20                 songwriters Lennon and McCartney, they came to be perceived as an embodimen\  
21             t of the ideals shared by the counterculture of the 1960s.  
22         </blockquote>  
23     </body>  
24 </html>
```

(the above code snippet online)

If you save the above page and load it on your browser, you will see this:



Blockquote Example with some Styling

As you can see from the above screenshot, the blockquote now has some border to the left side of it.

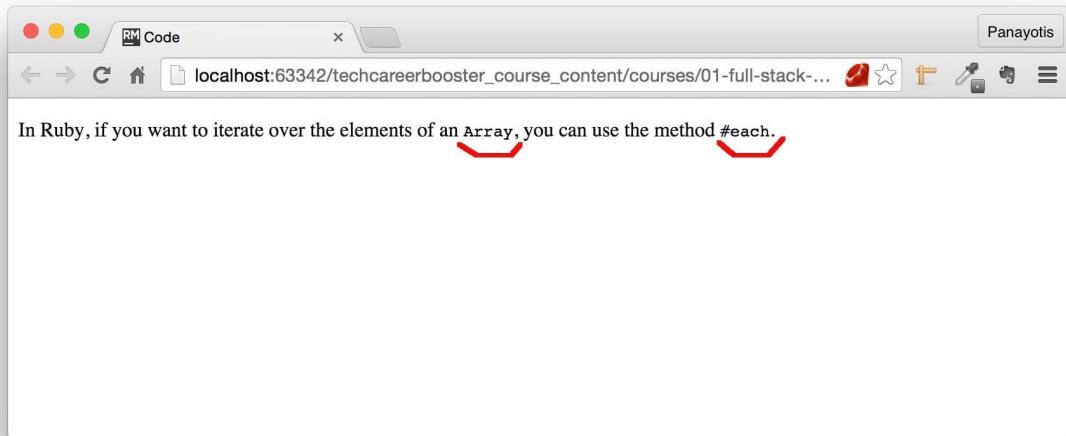
code and pre

Sometimes we want to include programming source code examples. For these blocks, we want to make them stand out from the rest of the text. This can be achieved with the help of the code HTML tag. Look at the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Code</title>
6      </head>
7
8      <body>
9          <p>
10             In Ruby, if you want to iterate over the elements of an <code>Array</code>, \
11             you can use the method <code>#each</code> .
12         </p>
13     </body>
14 </html>
```

(the above code snippet online)

If you save the above code and load the page on your browser, you will see this:



Code tags used to print programming source code phrases

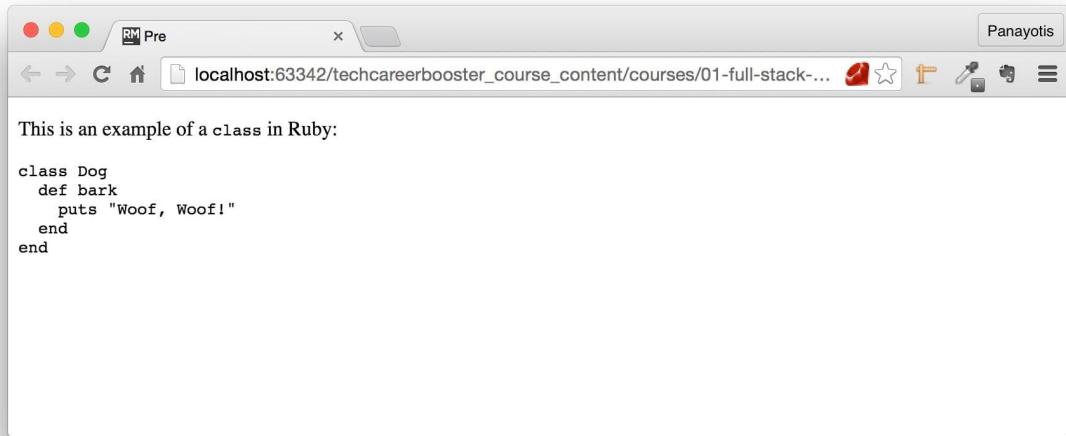
Browser is using a special kind of font family (usually a monospace family) to print text that is included inside code blocks. And you can always style code blocks with CSS according to your preferences. `code` element is used when we want to mark, as source code, only short text. If you want to mark as source code long multi-line text you should prefer the `pre` element. The `pre` element defines a pre-formatted piece of content and preserves the spaces, new lines e.t.c.

Look at the following example:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Pre</title>
6      </head>
7
8      <body>
9          <p>
10             This is an example of a <code>class</code> in Ruby:
11         </p>
12
13         <pre>
14     class Dog
15         def bark
16             puts "Woof, Woof!"
17         end
18     end
19     </pre>
20  </body>
21 </html>
```

(the above code snippet online)

If you save the above file and reload the page you will see this:



Pre-formatted text printed with the help of pre tag

If you compare with the HTML source code, you will see that the content inside the pre element has been printed exactly as given, using a special kind of font family.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. Build a web page with 2 quotes from Wikipedia and 2 inline frames of the corresponding pages. The Wikipedia pages that you have to use need to be about specific programming languages. Also, you need to find some sample code (multi-line) and put it inside your page as example of code of these languages.
2. Here is an example of such a page:

Ruby

Ruby is a dynamic, reflective, object-oriented, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan.

This is an example source code from this language:

```
class Person
  attr_reader :name, :age
  def initialize(name, age)
    @name, @age = name, age
  end
  def <>(person) # the comparison operator for sorting
    age <> person.age
  end
  def to_s
    "#{@name} (#{@age})"
  end
end
```

Here, you can see the whole reference from Wikipedia:

Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, [12] and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA),[13] meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.[14] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2015, Java is one of the most popular programming languages in use,[15] [16][17][18] particularly for client-server web applications, with a reported 9 million developers[citation needed]. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

This is an example source code from this language:

```
class HelloWorldApp {
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

Here, you can see the whole reference from Wikipedia:

Watch this video too to see how it works: [Task page for Other HTML Elements Chapter](#)

3. Here are some hints and requirements:

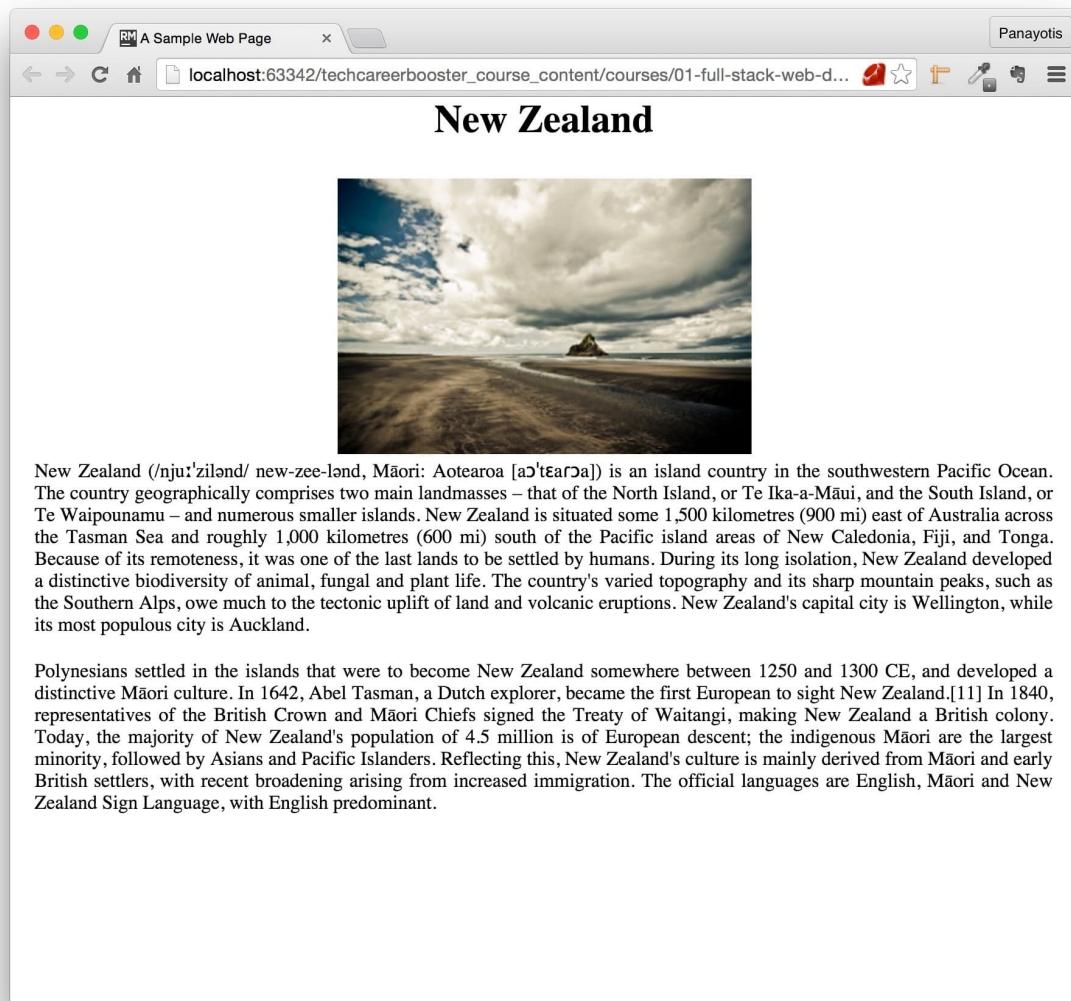
1. Use the font “Comfortaa” (all available weights: 300, 400, 700) from Google Fonts.
2. You can find example of source code inside the corresponding Wikipedia pages.
3. Make sure that the page content is centered and that there are equal margins left and right.
4. The blockquotes need to have a border at the left side.
5. The pre-formatted container needs to be with a background color and needs to have a lot of padding so that the source code inside is quite away from the edges of the container, like in the example page above. Also, make sure the container has rounded corners.
6. The iframe height should be 300px, and the width should be 100%;
7. Inside the pre-formatted container, the content may be quite long, wider than the actual available width. In that case, you can use the property “overflow-x” with value “auto”. In that case, browser will limit the visible source code to the available width, and will add a horizontal bar to allow user to see the rest of the content that is hidden.
4. You will need to upload your work to Netlify.

Chapter 31 - Mobile Friendly Web Pages

Summary

With the advent of tablets and mobile devices, Web page designers faced a serious problem. They had to make their Web pages look good on small displays. They had to design their Web pages to be responsive. In this chapter you will learn how to create responsive Web pages.

A page like this:



Web Page displayed on Desktop

might look good on the desktop computer display. But how can we make it look good on the mobile device too, like this?

The screenshot shows a mobile browser simulation for an Apple iPhone 6 with a 375x627 screen size. The title bar reads "RM A Sample Web Page" and "Panayotis". The header includes a back/forward button, a refresh button, and a search/address bar showing "localhost:63342/techcareerb...". The top navigation bar has tabs for "Device" (set to "Apple iPhone 6"), "Portrait -", "Network" (set to "No throttling"), and "UA" (set to "Mozilla/5.0 (iPhone; C"). Below the header is a horizontal ruler from 0 to 400 pixels. The main content area displays the text "New Zealand" above a landscape image of a beach under a cloudy sky. The sidebar on the left shows vertical scroll position markers at 0, 100, 200, 300, 400, 500, and 600 pixels. At the bottom, the WebKit Inspector's developer tools are visible, showing the "Elements" tab selected, the DOM tree with a node highlighted, and tabs for "Styles", "Console", "Emulation", and "Rendering". The status bar at the very bottom indicates "Device: Apple iPhone 6", "Model: Apple iPhone 6", and "Portrait - default".

0 100 200 300 400 500 600

375px width

New Zealand

New Zealand (/nju:'zilənd/ new-zee-lənd, Māori: Aotearoa [aɔ'tearoɑ]) is an island country in the southwestern Pacific Ocean. The country geographically comprises two main landmasses – that of the North Island, or Te Ika-a-Māui, and the South Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situated some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledonia, Fiji, and Tonga. Because of its remoteness, it was one of the last lands to be settled by humans. During its long isolation, New Zealand developed a distinctive biodiversity of animal, fungal and plant life. The country's varied topography and its sharp mountain peaks, such as the Southern Alps, owe much to the tectonic uplift

Styles Content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)

Console Emulation Rendering

Device Model: Apple iPhone 6 Portrait - default

We are going to teach you how you will adapt the layout of your page. For example, a page like this:

| Football | Basketball | Formula One |
|--|--|---|
| Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football. ^{[1][2]} These different variations of football are known as football codes. | Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports. ^[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level of professional basketball in the world and NBA players are the world's best paid sportsmen, by average annual salary per player. ^{[2][3]} | Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA). The FIA Formula One World Championship has been the premier form of racing since the inaugural season in 1950, although other Formula One races were regularly held until 1983. The "formula", designated in the name, refers to a set of rules to which all participants' cars must conform. ^[2] The season consists of a series of races, known as Grandes Prix (from French, originally meaning great prizes), held throughout the world on purpose-built F1 circuits and public roads. |
| Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century. ^{[3][4]} The expanse of the British Empire allowed these rules of football to spread to areas of British influence outside of the directly controlled Empire, ^[5] though by the end of the nineteenth century, distinct regional codes were already developing: Gaelic football, for example, deliberately incorporating the rules of local traditional football games in order to maintain Irish cultural heritage. ^[6] In 1888, The Football League was founded in England, becoming the first of many professional football competitions. During the twentieth century, several of the various kinds of football grew to become some of the most popular team sports in the world. | A team can score a field goal by shooting the ball through the basket during regular play. A field goal scores three points for the shooting team if the player shoots from behind the three-point line, and two points if shot from in front of the line. A team can also score via free throws, which are worth one point, after the other team was assessed with certain fouls. The team with the most points at the end of the game wins, but additional time (overtime) is issued when the score is tied at the end of regulation. The ball can be advanced on the court by bouncing it while walking or running or throwing it to a teammate. It is a violation to lift or drag one's pivot foot without dribbling the ball, to carry it, or to hold the ball with both hands then resume dribbling. | The results of each race are evaluated using a points system to determine two annual World Championships, one for drivers, one for constructors. The racing drivers are required to be holders of valid Super Licences, the highest class of racing licence issued by the FIA. ^[3] The races are required to be held on tracks graded 1 (formerly A), the highest grade a track can receive by the FIA. ^[3] Most events are held in rural locations on purpose-built tracks, but there are several events in city centres throughout the world, with the Monaco Grand Prix being the most obvious and famous example. |

Page With 3-columns Layout

will be converted to a page like this, when viewed on a mobile device. I.e. a 3-column layout will be converted to a 1-column layout:

The screenshot shows a mobile device simulation for an Apple iPhone 6 in portrait mode. The page content is a three-column layout. The left column contains the word "Sports". The middle column contains the word "Football". The right column contains a photograph of a football match and a descriptive text block. The bottom of the screen shows the WebKit Inspector interface, specifically the Elements tab where the body element is selected. A tooltip in the bottom right corner of the inspector indicates that the content is owned by Tech Career Booster (<https://www.techcareerbooster.com>). The bottom navigation bar of the inspector includes Device, Model: Apple iPhone 6, and Portrait - default.

Sports

Football



Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.[1][2] These different variations of football are known as football codes.

Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.[3][4] The expanse of the British Empire allowed these rules of football to spread to areas

content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)

Or it may be turned automatically to a 2-column layout on tablets:

The screenshot shows the Chrome DevTools Emulator interface. At the top, it indicates a 'Device: Google Nexus 10' and 'Portrait' orientation. The main content area displays a web page titled 'Sports'. Below the title, there are two columns: 'Football' and 'Basketball'. Each column has an image and some descriptive text. The 'Football' section includes a paragraph about the history and variations of football. The 'Basketball' section includes a paragraph about the rules and scoring of basketball. The bottom of the screen shows the DevTools Elements tab with the DOM structure and the Styles tab with the applied CSS rules.

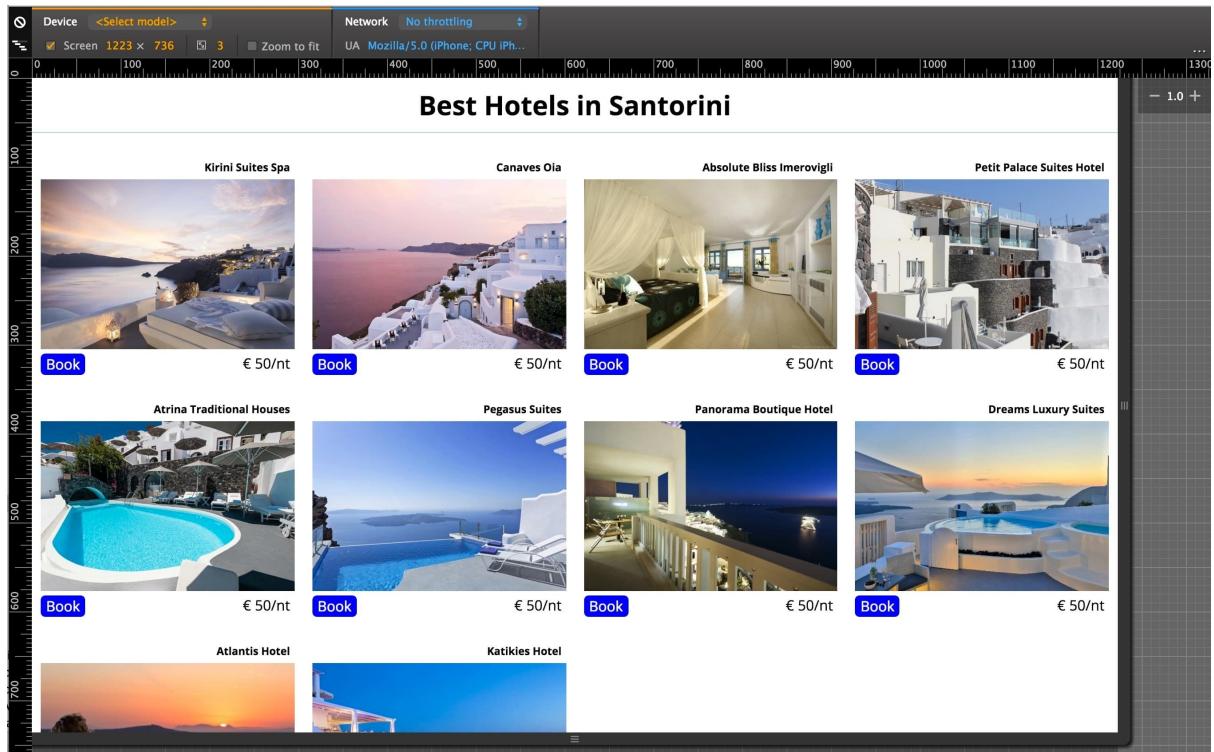
```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body cz-shortcut-listen="true">
    <h1>Sports</h1>
    <div id="left-column" class="column">...</div>
    <div id="right-column" class="column">...</div>
</body>
```

Styles

```
element.style { }
body { }
```

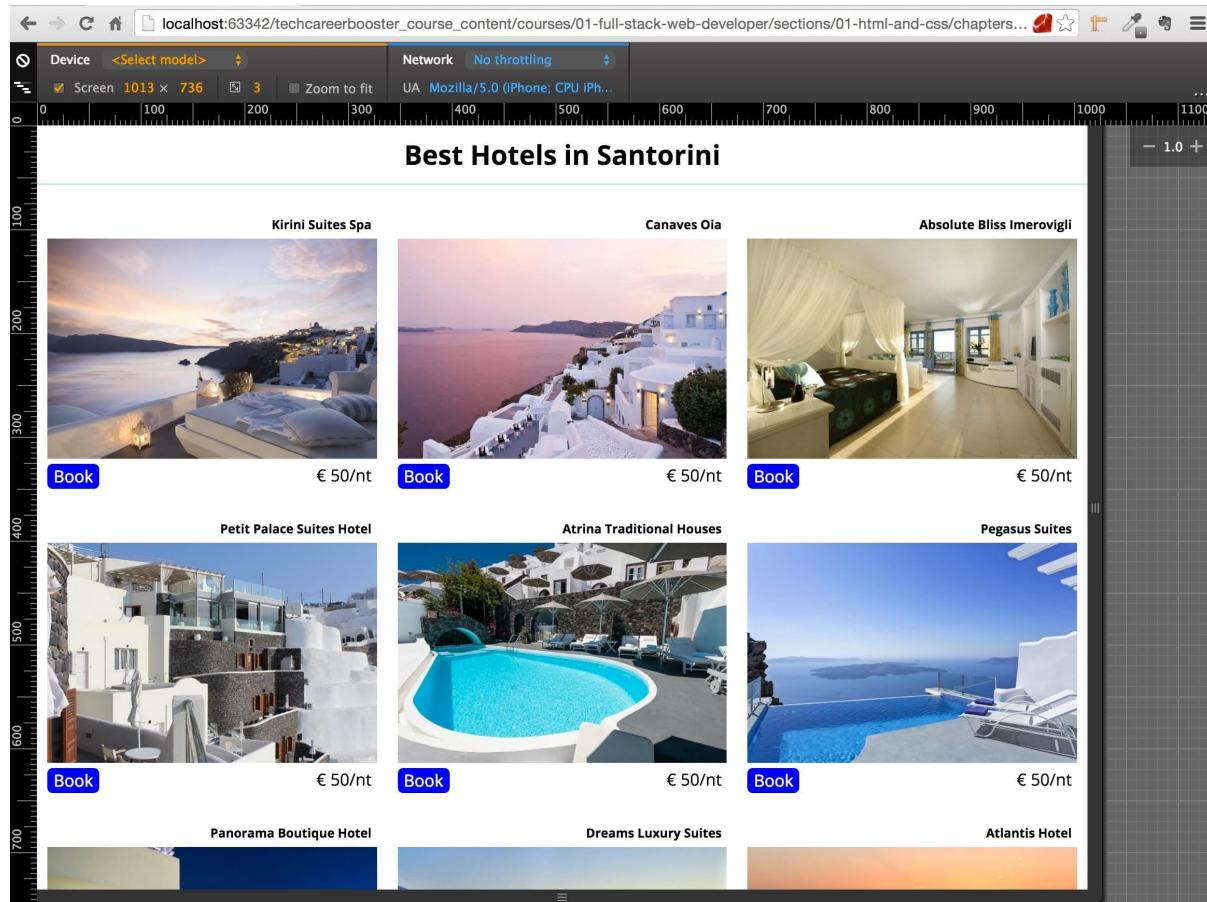
2-column Layout For Tablets

You will learn how the following page that has 4-columns layout on a big screen:



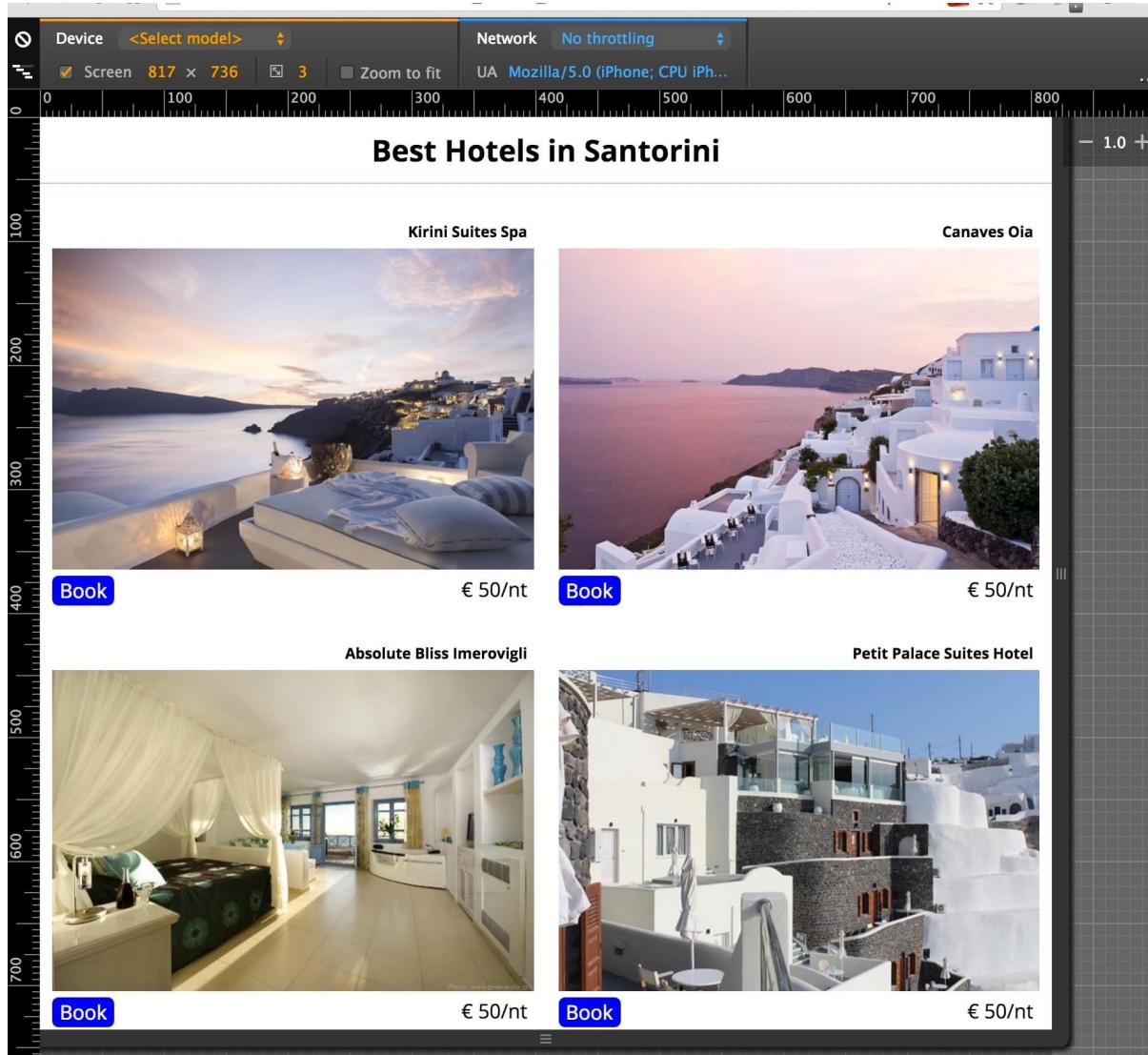
Responsive Page with Hotel Images - 4-columns layout

is automatically turned to a 3-columns layout on medium-sized screens:



Responsive Page with Hotel Images - 3-columns layout

is automatically turned to a 2-columns layout on small-sized screens:



Responsive Page with Hotel Images - 2-columns layout

and is automatically turned to a 1-column layout on extra small screens:

Screenshot of a mobile browser developer tools interface showing a responsive web page for "Best Hotels in Santorini".

The browser window title is "Best Hotels in Santorini" and the URL is "localhost:63342/techcareerboost...". The developer tools header shows "Device: Apple iPhone 6 Plus", "Portrait" orientation, "Network: No throttling", and "UA: Mozilla/5.0 (iPhone; CPU iPhone OS 10_0 like Mac OS X)". A zoom level of "1.0" is selected.

The main content area displays two hotel listings:

- Kirini Suites Spa**: Features a sunset over the sea and a white terrace with a bed. A blue "Book" button is at the bottom left, and the price "€ 50/nt" is at the bottom right.
- Canaves Oia**: Features a sunset over the sea and a white building on a cliff. A blue "Book" button is at the bottom left, and the price "€ 50/nt" is at the bottom right.

The developer tools bottom panel shows the following:

- Elements tab: Shows the DOM structure with a highlighted `<body cz-shortcut-listen="true">`.
- Console tab: Displays the message "content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)".
- Emulation tab: Shows a zoom level of "1.0".
- Rendering tab: Shows a preview of the page with a yellow border around the main content area.

Learning Goals

1. Learn about the viewport concept.
2. Learn how you can switch developer tools to mobile device mode and inspect how your page would look like on various mobile devices.
3. Learn how the browser scales down the page so that its content is fully displayed on particular mobile device width.
4. Learn what you should do to stop browser from scaling down / zooming out your page content. Your page needs to be displayed with the dimensions specified by your CSS rules.
5. Learn about media queries that will allow you to specify CSS rules according to the physical properties of the mobile device display.
 1. Learn about the various media types.
 2. Learn about the expressions and the media features.
 3. Learn about the @media rule syntax.
6. Learn to design Web pages that are responsive and adapt their layout depending on the display properties.
7. Learn about the different responsive design breakpoints that you can set.
8. Learn about the difference between Device VS Viewport.
9. Learn how you can load CSS files with media queries.
10. Learn about responsive images and videos.
11. Learn about the responsive Web frameworks.

Introduction

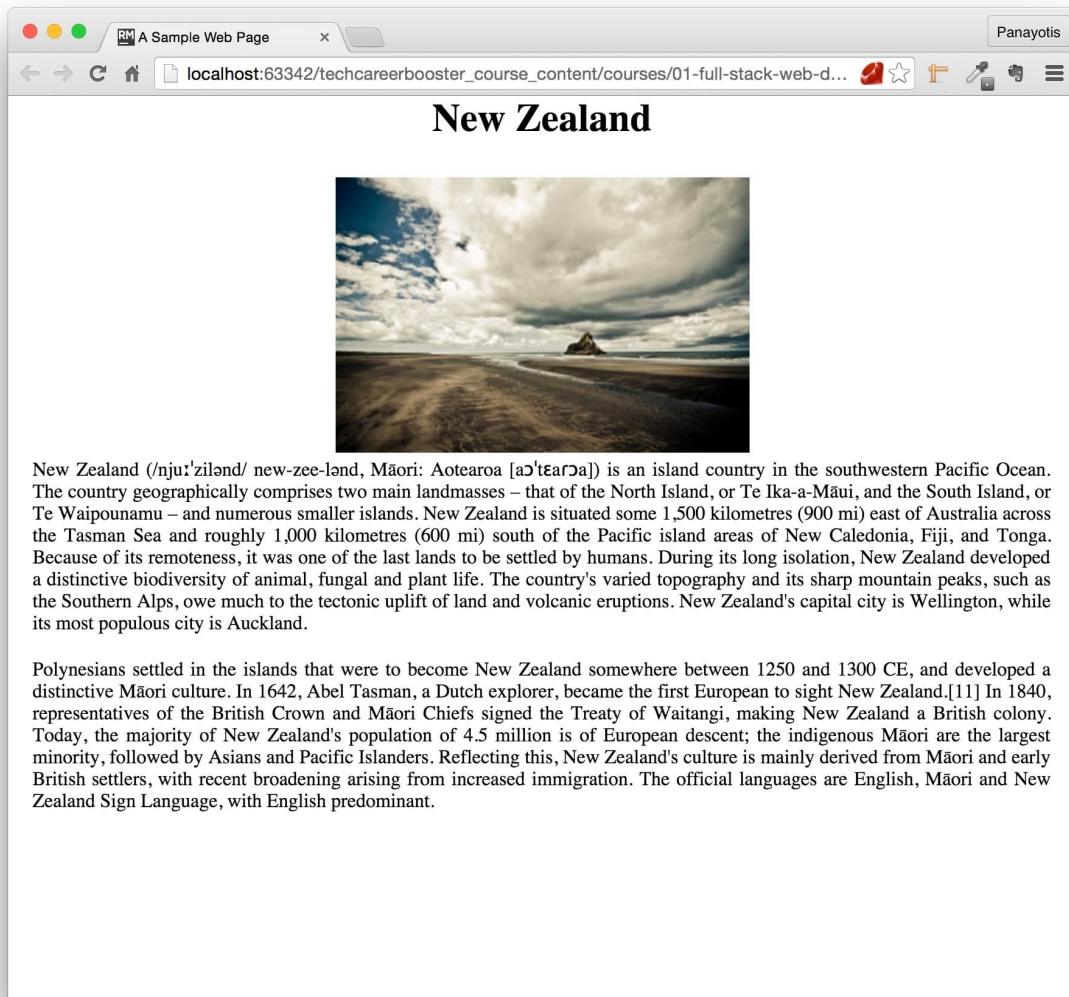
Now that mobile phones have become so popular, it is very likely that our Web site visitors are coming to our Web pages via a mobile browser. We need to make sure that the pages are looking nice and are functioning well on small displays, either mobiles or tablets or other small display devices.

When our Web page adapts nicely on small devices we have a responsive Web page design. Our Web pages need to be responsive.

The Viewport

The first detail that you need to master is the concept of the Viewport. The viewport is the area of the page that is visible to the user. The viewport varies by device. For smaller devices that viewport is smaller.

Imagine that you have designed the following Web page that looks good on your desktop



Web Page displayed on Desktop

The source code for this page is the following:

HTML:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>A Sample Web Page</title>
6     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7   </head>
8
9   <body>
10
11     <h1>New Zealand</h1>
12
```

```
13     
14
15
16     <p>
17         New Zealand (/njuːzilənd/ new-zee-lənd, Māori: Aotearoa [aːtəraːoa]) is an\
18         island country in the southwestern Pacific Ocean. The country geographically com\
19         prises
20             two main landmasses – that of the North Island, or Te Ika-a-Māui, and the S\
21         outh Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situ\
22         ated
23             some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and \
24         roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledo\
25         nia,
26             Fiji, and Tonga. Because of its remoteness, it was one of the last lands to\
27         be settled by humans. During its long isolation, New Zealand developed a distinc\
28         tive
29             biodiversity of animal, fungal and plant life. The country's varied topogra\
30         phy and its sharp mountain peaks, such as the Southern Alps, owe much to the tect\
31         onic
32             uplift of land and volcanic eruptions. New Zealand's capital city is Wellin\
33         gton, while its most populous city is Auckland.
34     </p>
35
36     <p>
37         Polynesians settled in the islands that were to become New Zealand somewher\
38         e between 1250 and 1300 CE, and developed a distinctive Māori culture. In 1642, A\
39         bel
40             Tasman, a Dutch explorer, became the first European to sight New Zealand.[1\
41         1] In 1840, representatives of the British Crown and Māori Chiefs signed the Tre\
42         ty
43             of Waitangi, making New Zealand a British colony. Today, the majority of Ne\
44         w Zealand's population of 4.5 million is of European descent; the indigenous Māor\
45         i
46             are the largest minority, followed by Asians and Pacific Islanders. Reflect\
47         ing this, New Zealand's culture is mainly derived from Māori and early British se\
48         ttlers,
49             with recent broadening arising from increased immigration. The official lan\
50         guages are English, Māori and New Zealand Sign Language, with English predominant\
51         .
52     </p>
53
54     </body>
55
56 </html>
```

(the above code snippet online)

and CSS

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13     text-align: center;
14 }
15
16 h1 {
17     margin-bottom: 3rem;
18 }
19
20 p {
21     padding: 0 2rem;
22     margin-bottom: 2rem;
23     text-align: justify;
24 }
```

(the above code snippet online)

Save the above file and load it on your browser. You will see a nice looking page, like the one displayed on the above screenshot.

Now, let's try to see how this page would look like on a mobile device. We will use the Chrome developer tools to switch to mobile device mode emulation.

1. Open the developer tools.
2. Click on the device mode icon.
3. Select the device “Apple iPhone 6” (for example).
4. Refresh the page.

The instructions to do that are displayed on this video:

[How you can enable mobile device mode on Chrome developer tools](#)

What you see is something like this:

The screenshot shows the Chrome DevTools interface with the mobile emulation panel open. The device is set to an Apple iPhone 6 with a screen size of 375 x 627 pixels. The orientation is portrait, and there is no throttling applied. The user agent is Mozilla/5.0 (iPhone; CPU). The main content area displays a heading "New Zealand" and a photograph of a beach under a cloudy sky. Below the image is a detailed paragraph about New Zealand's geography, history, and biodiversity. At the bottom of the page, there is a note about the indigenous Māori population. The DevTools footer shows the element `<body cz-shortcut-listen="true">` selected in the Elements tab. The bottom right corner shows a zoomed-in view of the margin and border properties for an element.

New Zealand (/nuː'zilənd/ new-zee-lənd, Māori: Aotearoa [aɔ'teərɔa]) is an island country in the southwestern Pacific Ocean. The country geographically comprises two main landmasses – that of the North Island, or Te Ika-a-Māui, and the South Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situated some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledonia, Fiji, and Tonga. Because of its remoteness, it was one of the last lands to be settled by humans. During its long isolation, New Zealand developed a distinctive biodiversity of animal, fungal and plant life. The country's varied topography and its sharp mountain peaks, such as the Southern Alps, owe much to the tectonic uplift of land and volcanic eruptions. New Zealand's capital city is Wellington, while its most populous city is Auckland.

Polynesians settled in the islands that were to become New Zealand somewhere between 1250 and 1300 CE, and developed a distinctive Māori culture. In 1642, Abel Tasman, a Dutch explorer, became the first European to sight New Zealand.[11] In 1840, representatives of the British Crown and Māori Chiefs signed the Treaty of Waitangi, making New Zealand a British colony. Today, the majority of New Zealand's population of 4.5 million is of European descent; the indigenous Māori are the largest minority, followed by

content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)

Styles Event Listeners DOM Breakpoints Properties

Filter + ↕ ◆

element.style { } margin - border -

The browser scales down (zoomed out) the original page, so that the entire page fits inside the screen of the mobile device. In fact, on this particular device, the browser has scaled down the page to 40% of its original size. Do you see the 0.4 on the far right? This is the indicator of the scaling that has been performed.

You will understand that even better if you do the following. Let's add a div at the top of the page with width equal to the screen size of the iPhone 6, a.k.a. 375px.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>A Sample Web Page</title>
6          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
7      </head>
8
9      <body>
10
11         <div id="rectangle">
12             375px width
13         </div>
14
15         <h1>New Zealand</h1>
16
17         
19
20         <p>
21             New Zealand (/njuːzilənd/ new-zee-ənd, Māori: Aotearoa [aːtəraːoa]) is an\
22             island country in the southwestern Pacific Ocean. The country geographically com\
23             prises
24                 two main landmasses – that of the North Island, or Te Ika-a-Māui, and the S\
25                 outh Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situ\
26                 ated
27                 some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and \
28                 roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledo\
29                 nia,
30                 Fiji, and Tonga. Because of its remoteness, it was one of the last lands to\
31                 be settled by humans. During its long isolation, New Zealand developed a distinc\
32                 tive
33                 biodiversity of animal, fungal and plant life. The country's varied topogra\
34                 phy and its sharp mountain peaks, such as the Southern Alps, owe much to the tect\
35                 onic
36                 uplift of land and volcanic eruptions. New Zealand's capital city is Wellin\
37                 gton, while its most populous city is Auckland.
38         </p>
39
```

```
40      <p>
41          Polynesians settled in the islands that were to become New Zealand somewhere\
42 e between 1250 and 1300 CE, and developed a distinctive Māori culture. In 1642, A\
43 bel
44          Tasman, a Dutch explorer, became the first European to sight New Zealand.[1\
45 1] In 1840, representatives of the British Crown and Māori Chiefs signed the Tre\
46 ty
47          of Waitangi, making New Zealand a British colony. Today, the majority of Ne\
48 w Zealand's population of 4.5 million is of European descent; the indigenous Māor\
49 i
50          are the largest minority, followed by Asians and Pacific Islanders. Reflect\
51 ing this, New Zealand's culture is mainly derived from Māori and early British se\
52 ttlers,
53          with recent broadening arising from increased immigration. The official lan\
54 guages are English, Māori and New Zealand Sign Language, with English predominant\
55 .
56      </p>
57
58  </body>
59
60 </html>
```

(the above code snippet online)

and fix its width inside the CSS file:

```
1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7  html {
8      font-size: 62.5%;
9  }
10
11 body {
12     font-size: 1.6rem;
13     text-align: center;
14 }
15
16 h1 {
17     margin-bottom: 3rem;
18 }
19
20 p {
21     padding: 0 2rem;
```

```
22     margin-bottom: 2rem;  
23     text-align: justify;  
24 }  
25  
26 #rectangle {  
27     width: 375px;  
28     border: 1px solid red;  
29 }
```

(the above code snippet online)

If you save the above two files and reload the page on your browser, you will see this (still with the mobile device mode on):

The screenshot shows the Chrome DevTools mobile emulation interface. The top bar indicates the device is an Apple iPhone 6 in portrait mode. The main content area displays a web page titled "New Zealand" featuring a landscape image and a descriptive paragraph about the country's geography and history. The bottom of the screen shows the DevTools navigation bar with tabs for Elements, Console, Sources, Network, Timeline, and a collapsed section. The Console tab is active, showing a log entry from Tech Career Booster. The Emulation tab is also visible.

New Zealand (/nju:'zilənd/ new-zee-lənd, Māori: Aotearoa [aɔ'teərɔa]) is an island country in the southwestern Pacific Ocean. The country geographically comprises two main landmasses – that of the North Island, or Te Ika-a-Māui, and the South Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situated some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledonia, Fiji, and Tonga. Because of its remoteness, it was one of the last lands to be settled by humans. During its long isolation, New Zealand developed a distinctive biodiversity of animal, fungal and plant life. The country's varied topography and its sharp mountain peaks, such as the Southern Alps, owe much to the tectonic uplift of land and volcanic eruptions. New Zealand's capital city is Wellington, while its most populous city is Auckland.

Polynesians settled in the islands that were to become New Zealand somewhere between 1250 and 1300 CE, and developed a distinctive Māori culture. In 1642, Abel Tasman, a Dutch explorer, became the first European to sight New Zealand.[11] In 1840, representatives of the British Crown and Māori Chiefs signed the Treaty of Waitangi, making New Zealand a British colony. Today, the majority of New Zealand's population of 4.5 million is of European descent: the

content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)

Do you see that the fixed width div, has been scaled down? It is now displayed on 40% of its original size. The CSS instructs the browser to draw that div with width 375px, which is the width of the iPhone 6 screen size. The div, should have occupied the whole width of the screen, but it does not.

So, that trick does not work well on modern Web pages, that do not have only text to display. You can also see the problem with the image too. It is very small, although there is a lot of blank space left and right to it, that could have been used to display the image in a larger scale.

With the advent of HTML5, modern web pages give instructions to browser how to load the Web page on mobile devices. This is done with the use of the meta with name equal to viewport, i.e. the viewport meta tag.

All your Web pages should have the following meta tag on their head section:

```
1 <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

(the above code snippet online)

This instructs the browser to do the following:

1. consider as viewport width the width of the device. This is the width=device-width feature. So, if the device width is 375px (like on iPhone 6), then the number of pixels of the page that will be available to view on the horizontal direction will be 375px.
2. do not scale down or scale up the page contents. But, instead, use the original page scale/size. This is the initial-scale=1.0

Add this meta tag to the page above, like here:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>A Sample Web Page</title>
7     <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
8   </head>
9
10  <body>
11
12    <div id="rectangle">
13      375px width
14    </div>
15
16    <h1>New Zealand</h1>
17
18    
```

```
20
21      <p>
22          New Zealand (/njuːzilənd/ new-zee-lənd, Māori: Aotearoa [aːtəraːoa]) is an\
23  island country in the southwestern Pacific Ocean. The country geographically com\
24  prises
25      two main landmasses – that of the North Island, or Te Ika-a-Māui, and the S\
26  outh Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situ\
27  ated
28      some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and \
29  roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledo\
30  nia,
31      Fiji, and Tonga. Because of its remoteness, it was one of the last lands to\
32  be settled by humans. During its long isolation, New Zealand developed a distinc\
33  tive
34      biodiversity of animal, fungal and plant life. The country's varied topogra\
35  phy and its sharp mountain peaks, such as the Southern Alps, owe much to the tect\
36  onic
37      uplift of land and volcanic eruptions. New Zealand's capital city is Wellin\
38  gton, while its most populous city is Auckland.
39      </p>
40
41      <p>
42          Polynesians settled in the islands that were to become New Zealand somewher\
43  e between 1250 and 1300 CE, and developed a distinctive Māori culture. In 1642, A\
44  bel
45          Tasman, a Dutch explorer, became the first European to sight New Zealand.[1\
46  1] In 1840, representatives of the British Crown and Māori Chiefs signed the Treas\
47  ty
48          of Waitangi, making New Zealand a British colony. Today, the majority of Ne\
49  w Zealand's population of 4.5 million is of European descent; the indigenous Māor\
50  i
51          are the largest minority, followed by Asians and Pacific Islanders. Reflect\
52  ing this, New Zealand's culture is mainly derived from Māori and early British se\
53  ttlers,
54          with recent broadening arising from increased immigration. The official lan\
55  guages are English, Māori and New Zealand Sign Language, with English predominant\
56 .
57      </p>
58
59      </body>
60
61  </html>
```

(the above code snippet online)

If you save the above file and reload the page on the mobile device mode, you will see this:

A Sample Web Page

Panayotis

Device: Apple iPhone 6 | Network: No throttling

Screen: 375 x 627 | Zoom: 2 | Zoom to fit

UA: Mozilla/5.0 (iPhone; CPU iPhone OS 11_0 like Mac OS X)

0 100 200 300 400

0 100 200 300 400 500 600 700

375px width

New Zealand



New Zealand (/nju:'zilənd/ new-zee-lənd, Māori: Aotearoa [aɔ'tearoɑ]) is an island country in the southwestern Pacific Ocean. The country geographically comprises two main landmasses – that of the North Island, or Te Ika-a-Māui, and the South Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situated some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledonia, Fiji, and Tonga. Because of its remoteness, it was one of the last lands to be settled by humans. During its long isolation, New Zealand developed a distinctive biodiversity of animal, fungal and plant life. The country's varied topography and its sharp mountain peaks, such as the Southern Alps, owe much to the tectonic uplift

Elements Console Sources Network Timeline

body

Styles Content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)

Console Emulation Rendering

Device: Apple iPhone 6 | Model: Apple iPhone 6 | Orientation: Portrait - default

Do you see that the page is now displayed on its original scale/size? The div is exactly 375px (the size of iPhone 6 screen width) and the image has been displayed with the correct dimensions. You can also see that the scaling factor on the far right is “1.0”:

The screenshot shows a mobile browser simulation for an Apple iPhone 6 with a 375x627 screen size. The device is set to portrait mode. A red circle highlights the 'Scaling Factor' slider at 1.0. The page content includes a title 'New Zealand' and a large image of a beach under a cloudy sky. The bottom of the screen displays the WebKit Inspector's developer tools, showing the element tree, styles, and console tabs.

A Sample Web Page

Panayotis

Device: Apple iPhone 6

Portrait -

Network: No throttling

Screen: 375 x 627

Zoom to fit

UA: Mozilla/5.0 (iPhone; C

0 100 200 300 400

0 100 200 300 400 500 600

375px width

New Zealand

Scaling Factor

New Zealand (/nju:'zilənd/ new-zee-lənd, Māori: Aotearoa [aɔ'tearoɑ]) is an island country in the southwestern Pacific Ocean. The country geographically comprises two main landmasses – that of the North Island, or Te Ika-a-Māui, and the South Island, or Te Waipounamu – and numerous smaller islands. New Zealand is situated some 1,500 kilometres (900 mi) east of Australia across the Tasman Sea and roughly 1,000 kilometres (600 mi) south of the Pacific island areas of New Caledonia, Fiji, and Tonga. Because of its remoteness, it was one of the last lands to be settled by humans. During its long isolation, New Zealand developed a distinctive biodiversity of animal, fungal and plant life. The country's varied topography and its sharp mountain peaks, such as the Southern Alps, owe much to the tectonic uplift

Elements Console Sources Network Timeline >

html body

Styles Content is owned by TechCareerBooster (https://www.techcareerbooster.com)

Console Emulation Rendering

Device: Apple iPhone 6 Model: Apple iPhone 6 Portrait - default

Now that we have instructed the browser to not do any scaling down trick in order to draw the contents of the page on smaller devices, we know that the sizes specified inside our CSS rules are going to be respected.

And here comes another problem now. Sometimes, elements have sizes that are looking good on desktop computers, but, these sizes, are not appropriate for smaller device displays. Or there are page designs that are good for large screens but not adequate for small screens. How can we adapt our page design according to the media type and its features?

Generally, we need to find a way to follow the next rules:

1. The users are used to scroll down vertically, but not horizontally. Horizontal scrolling or need for zoom out actions by the user results in a bad experience. Hence, we need to make sure that the content of our page is of such width that can fit to the viewport width of the device.
2. On the other hand, we should not rely on a particular specific viewport size when thinking about the sizes of our elements. In other words, we should not design our page for iPhone 6 375px width only, or for large desktop displays 1920px only.

There are some tools at your hand that can help you design a Web page that would look good both on large desktops and on tablets and mobile devices. For example:

1. Try to avoid large absolute positioning.
2. Try to use relative sizing for width, i.e. 80% will set the size of an element to 80% of this containing parent.
3. Use media queries to apply different CSS rules depending on the device width.

The last item on the previous list is the new thing that you are going to learn now. Let's expand on this.

Media Queries

A `media` query is composed by a `media` type and one or more expressions. The `media` type defines on which types of media the `media` query is referring to. The expression tells which properties of the particular media types need to be met in order for the CSS rules scoped by the `media` query are going to be applied.

media query



Media Queries Definition

Media Type

The media type can take the following values:

1. `all`, which refers to any display / media type.
2. `print`, which refers to documents that are to be presented in print preview mode.
3. `screen`, which refers to color computer or mobile displays.
4. `speech`, which refers to speech synthesizers.

If you do not specify the media type, then it is assumed to be `all`. Exception to this rule is when using the operators `not` or `only`. Will talk about that later on.

Expressions

The expressions are composed of media features, properties, that, when calculated at run time, they have to evaluate to either true or false.

This is an example of an expression:

```
(max-device-width: 800px)
```

It uses the media feature `max-device-width` and it is true when the media that is used to display the HTML content has maximum screen width 800px. Otherwise it is false. In other words, CSS rules that come with a media query that has this expression, will only be applied on all screens with width up to 800px, including.

@media Rule

In order to use media queries, you need to define them with the `@media` CSS keyword. The `@media` CSS syntax is the following:

```
@media condition set 1, condition set 2, ... {  
    CSS code  
}
```

1. We start with the `@media` word.
2. We then specify one or more condition sets separated by commas. For the condition sets see later on.
3. Then we open a block of CSS code, starting with a left curly brace `{`, then on their own lines, we have the CSS code, and then we close with a right curly brace `}` in its own line.

The condition sets are specified as follows:

`not|only mediatype and (media features)`

1. We specify, optionally, the media type. Note that if we do not specify the media type, then it is assumed all.
2. We can start a condition with the `not` or `only` keyword. The `not` negates the whole condition set. So, if the condition set would have been true without the `not` keyword, it becomes false when the `not` keyword exists. The `only` keyword is not very common and it is used if we want to support older browsers that they didn't have support for condition sets with media features.
3. We can use the boolean operator `and` to combine media types with expressions of media features. This specializes our rules by requiring all the conditions combined with the `and` boolean operator to be true.
4. The media features which are given inside parentheses.

Examples

We will immediately proceed to some examples of media queries as we learn how we can define and use them.

1) Example 1

```
1 @media (max-device-width: 800px) {  
2     .left-column {  
3         width: 30%;  
4     }  
5 }
```

(the above code snippet online)

In the above example:

1. We do not specify any media type. So, all is implied
2. There is one expression: (`max-device-width: 800px`).
3. There is one CSS rule that is scoped by this media query:`css .left-column { width: 30%; }`

This means that if the HTML page contains an HTML element that has the class `left-column` and the page is displayed on a screen with width 800px or less, then that particular element will have size equal to 30% of its container. Otherwise, if the browser displays the HTML page on a media with width greater than 800px, then this rule will not affect the way the element with class `left-column` is displayed.

2) Example 2

```

1 @media (min-device-width: 700px) and (orientation: landscape) {
2     div {
3         width: 700px;
4     }
5 }
```

(the above code snippet online)

This example uses the boolean operator `and`. So, it requires two conditions to be true. The first is that the minimum device width is 700px (`(min-device-width: 700px)`). The second is that the orientation of the display is landscape (`(orientation: landscape)`).

3) Example 3

```

1 @media screen and (min-device-width: 700px) and (orientation: landscape) {
2     div {
3         width: 700px;
4     }
5 }
```

(the above code snippet online)

This is the same example case as example number 2, but it has one more condition that needs to be true. It is about the media type that has to be a screen one.

4) Example 4

```

1 @media (min-device-width: 700px), screen and (orientation: landscape) {
2     div {
3         width: 700px;
4     }
5 }
```

(the above code snippet online)

This is an example that has 2 condition sets. First set is (`min-device-width: 700px`) and second condition set is `screen` and (`orientation: landscape`). As you can see the condition sets are separated by comma (,). In order for the CSS rules to be applied, the conditions are evaluated from left to right, and at least one condition needs to be true. So, the comma works like an or boolean operator.

This means that the above example could have been written as:

```

1 @media (min-device-width: 700px) {
2     div {
3         width: 700px;
4     }
5 }
6
7 @media screen and (orientation: landscape) {
8     div {
9         width: 700px;
10    }
11 }
```

(the above code snippet online)

or it could have been written as:

```

1 @media (min-device-width: 700px),
2 @media screen and (orientation: landscape) {
3     div {
4         width: 700px;
5     }
6 }
```

(the above code snippet online)

As you can understand, the first version is the shortest one and it is preferred.

5) Example 5

```

1 @media (min-device-width: 700px), not print and (orientation: landscape) {
2     div {
3         width: 700px;
4     }
5 }
```

(the above code snippet online)

On this example, we are using two condition sets. The second one is using the operator `not` which actually negates the conditions of the condition set. Hence, the CSS block will be applied to either devices with minimum device width 700px or to devices that are not `print` media and do not have orientation landscape.

Media Features

We have seen examples with some media features, such as `min-device-width` and `orientation`. The most important media features are the ones that have to do with the device or viewport dimensions. This is because they can be used to make our Web pages responsive to various device and display sizes.

Let's see the full list of media features that we can use:

1. `aspect-ratio`, `max-aspect-ratio`, `min-aspect-ratio`: Width over height of the viewport.
2. `color`, `max-color`, `min-color`: The number of bits per color for the output device.
3. `color-index`, `max-color-index`, `min-color-index`: The number of colors the device can display.
4. `device-aspect-ratio`, `max-device-aspect-ratio`, `min-device-aspect-ratio`: Width over height of the device dimensions.
5. `device-height`, `max-device-height`, `min-device-height`: The height of the device.
6. `device-width`, `max-device-width`, `min-device-width`: The width of the device.
7. `height`, `max-height`, `min-height`: The viewport height.
8. `width`, `max-width`, `min-width`: The width of the viewport.
9. `resolution`, `max-resolution`, `min-resolution`: The resolution of the device, using dpi or dp/cm.
10. `orientation`: The orientation of the viewport (landscape or portrait).

A more complex example page with responsive design

Let's try another more complex example with a page that needs to be responsive. Here is the original HTML code:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6          <title>Three Columns Page</title>
7          <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
8      </head>
9
10     <body>
11         <h1>Sports</h1>
12         <div id="left-column" class="column">
13             <h2>Football</h2>
14             
15             <p>
16                 Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to
17             to
18         </div>
19         <div id="middle-column" class="column">
20             <h2>Basketball</h2>
21             
22             <p>
23                 Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop, which is mounted on a backboard at each end of the court.
24             </p>
25         </div>
26         <div id="right-column" class="column">
27             <h2>Tennis</h2>
28             
29             <p>
30                 Tennis is a racket sport that can be played individually against a single opponent (singles) or between two teams of two players each (doubles). Each player uses a tennis racket, strung with cord, to strike a hollow rubber ball covered with felt over or around a net and into the opponent's court.
31             </p>
32         </div>
33     </body>
34 
```

refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.[1][2] These different variations of football are known as football codes.

</p>

<p>

Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.[3][4] The expanse of the British Empire allowed these rules of football to spread to areas of British influence outside of the directly controlled Empire,[5] though by the end of the nineteenth century, distinct regional codes were already developing: Gaelic football, for example, deliberately incorporated the rules of local traditional football games in order to maintain their heritage.[6] In 1888, The Football League was founded in England, becoming the first of many professional football competitions. During the twentieth century, several of the various kinds of football grew to become some of the most popular team sports in the world

</p>

</div>

<div id="middle-column" class="column">

<h2>Basketball</h2>

<p>

Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level of professional basketball in the world and NBA players are the world's best paid sportsmen, by average annual salary per player.[2][3]

</p>

<p>

65 A team can score a field goal by shooting the ball through the basket during regular play. A field goal scores three points for the shooting team if the player
66 shoots from behind the three-point line, and two points if shot from in front of the line. A team can also score via free throws, which are worth one point,
67
68 after the other team was assessed with certain fouls. The team with the most points at the end of the game wins, but additional time (overtime) is issued when
69
70 the score is tied at the end of regulation. The ball can be advanced on the court by bouncing it while walking or running or throwing it to a teammate. It is a
71
72 violation to lift or drag one's pivot foot without dribbling the ball, to carry it, or to hold the ball with both hands then resume dribbling.
73
74 </p>
75 </div>
76
77 <div id="right-column" class="column">
78 <h2>Formula One</h2>
79
80 <p>
81 Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA).
82
83 The FIA Formula One World Championship has been the premier form of racing since the inaugural season in 1950, although other Formula One races were regularly held until 1983. The "formula", designated in the name, refers to a set of rules, to which all participants' cars must conform. [2] The F1 season consists of a series of races, known as Grands Prix (from French, originally meaning great prizes), held throughout the world on purpose-built F1 circuits and public roads.
84
85 </p>
86 <p>
87 The results of each race are evaluated using a points system to determine two annual World Championships, one for drivers, one for constructors. The racing
88
89 drivers are required to be holders of valid Super Licences, the highest class of racing licence issued by the FIA. [3] The races are required to be held on tracks graded 1 (formerly A), the highest grade a track can receive by the FIA. [3] Most events are held in rural locations on purpose-built tracks, but there are several events in city centres throughout the world, with the Monaco Grand Prix being the most obvious and famous example.
90
91 </p>

```
111     </div>
112
113     </body>
114
115 </html>
```

(the above code snippet online)

with the following CSS:

```
1  * {
2      box-sizing: border-box;
3      padding: 0;
4      margin: 0;
5  }
6
7 html {
8     font-size: 62.5%;
9     font-family: Verdana, Arial, sans-serif;
10 }
11
12 body {
13     font-size: 1.6rem;
14 }
15
16 .column {
17     float: left;
18 }
19
20 #left-column,
21 #right-column {
22     width: 30%;
23 }
24
25 #middle-column {
26     width: 40%;
27 }
28
29 img {
30     float: left;
31     padding: 0.5rem 1rem;
32     width: 40%;
33 }
34
35 p {
36     margin-bottom: 2rem;
37     padding: 0 1.5rem;
```

```

38 }
39
40 h1 {
41   text-align: center;
42   margin-bottom: 3rem;
43 }
44
45 h2 {
46   text-align: center;
47   margin-bottom: 1rem;
48   padding-bottom: 0.5rem;
49   border-bottom: 1px solid lightblue;
50 }

```

(the above code snippet online)

If you save the above and load the page on your browser, you will see this:

The screenshot shows a desktop browser window displaying a three-column layout. The title bar reads "Three Columns Page" and "localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-developer/sections/01-html-and-css/chapters/31-mobile-friendly-web-pages/exercises-and-code/three-columns-page....". The main content area has a header "Sports". Below it are three columns:

- Football**: Includes a thumbnail image of a football game, a brief description of the sport, and a detailed paragraph about its history and evolution.
- Basketball**: Includes a thumbnail image of a basketball player, a brief description of the sport, and a detailed paragraph about its rules and popularity.
- Formula One**: Includes a thumbnail image of race cars, a brief description of the sport, and a detailed paragraph about its history, rules, and global reach.

Page with Sports Info

This page looks quite nice on a desktop display. But how does it look on an Apple iPhone 6 mobile device? Switch your mode to that phone emulation and see. You will get something like that:

The screenshot shows a browser window with developer tools open, displaying a mobile version of a sports page. The top status bar indicates the device is an Apple iPhone 6 with portrait orientation, and the network is not throttled. The page title is "Sports".

Sports

Football Basketball Formula One

 Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in

 Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.^[1] The

 Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA). The FIA Formula One World Championship has been

The bottom of the screenshot shows the browser's developer tools with the "Elements" tab selected, showing the DOM structure of the page. The "body" element is currently active.

As you can verify, the page is looking really bad on mobile devices. Why is that? What are the problems that we see?

First problem is that the width of the device is quite small to fit the 3 columns. We have these columns occupy 30%, for left and right, and 40% for middle column, of the width of the viewport. This is quite narrow on a viewport of width 375px like the one we have on iPhone 6.

What we can do to improve that?

First, we can say that on small devices like this, we are not going to use a 3-columns layout like this, but we are going to use 1-column layout. Right? How can we do that? We can say that the columns width is 100% for small devices. Let's change the CSS as follows:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9     font-family: Verdana, Arial, sans-serif;
10}
11
12 body {
13     font-size: 1.6rem;
14 }
15
16 .column {
17     float: left;
18 }
19
20 #left-column, #middle-column, #right-column {
21     width: 100%;
22 }
23
24 @media (min-device-width: 400px) {
25     #left-column,
26     #right-column {
27         width: 30%;
28     }
29
30     #middle-column {
31         width: 40%;
32     }
33 }
34
35 img {
```

```

36     float: left;
37     padding: 0.5rem 1rem;
38     width: 40%;
39 }
40
41 p {
42     margin-bottom: 2rem;
43     padding: 0 1.5rem;
44 }
45
46 h1 {
47     text-align: center;
48     margin-bottom: 3rem;
49 }
50
51 h2 {
52     text-align: center;
53     margin-bottom: 1rem;
54     padding-bottom: 0.5rem;
55     border-bottom: 1px solid lightblue;
56 }
```

(the above code snippet online)

What we did was the following:

1. We inserted a rule to apply `width: 100%`; to all 3 columns. Hence, making all the columns occupying the whole width of the device. AND
2. We have wrapped the part of the CSS code that sets other / different widths on the columns, inside a media query block that would apply only to devices with width 400px or wider.

```

1 @media (min-device-width: 400px) {
2     #left-column,
3     #right-column {
4         width: 30%;
5     }
6
7     #middle-column {
8         width: 40%;
9     }
10 }
```

(the above code snippet online)

If you save the above CSS file and reload the page on the mobile device mode, you will see this:

The screenshot shows a mobile device simulation for an Apple iPhone 6 in portrait mode. The page content is a three-column layout. The left column contains the word "Sports". The middle column contains the word "Football". The right column contains a photograph of a football match and a descriptive text block. The bottom of the screen shows the WebKit Inspector interface, specifically the Elements tab where the body element is selected. A tooltip in the bottom right corner of the inspector indicates that the content is owned by Tech Career Booster (<https://www.techcareerbooster.com>). The bottom navigation bar of the inspector includes Device, Model: Apple iPhone 6, and Portrait - default.

Sports

Football



Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.[1][2] These different variations of football are known as football codes.

Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.[3][4] The expanse of the British Empire allowed these rules of football to spread to areas

content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)

Perfect. As you can see above, the Web page now is looking really good. CSS is now giving 100% width to the column divs, since the device has width which is less than 400px, and hence, the media query block is not applied.

If you load the same page on your desktop display now, then the page would look like before. This is because the @media rules would overwrite the CSS rules for “#left-column, #middle-column, #right-column”, which are exactly before that.

Important - Mobile First: As you can see above, the main CSS rules, the ones outside of the media query blocks, are rules that make the page look good on the mobile device. The exceptions are being handled by the media query rules. Hence, main CSS rules handle the mobile devices and media queries handle the non-mobile devices. This is a **Mobile First** approach. We could have equally dealt with the same problem as follows:

```

1  #left-column,
2  #right-column {
3      width: 30%;
4  }
5
6  #middle-column {
7      width: 40%;
8  }
9
10 @media (max-device-width: 399px) {
11     #left-column, #middle-column, #right-column {
12         width: 100%;
13     }
14 }
```

That would have been a non-mobile first approach, because we first deal for non-mobile devices and then handle exceptionally the mobile devices. Try to use Mobile First approach in your design, because it will make your CSS and your page displayed faster on mobile devices. So, in order to do mobile first design, write your CSS for mobile devices and then introduce media queries to handle exceptions for larger devices.

Breakpoints

With the responsive design that we followed for the previous page, we introduced a limit on 400px width. Devices with width less than 400px and devices with width greater than or equal to 400px. For that reason, we call the 400px a breakpoint.

We have also seen that our page looks good on iPhone 6, which has 375px width. What happens if the mobile phone has larger than 400px width, but still quite small?

On the device mode emulator choose the iPhone 6 Plus. This one has 414px width. Inspect how the page looks on that device:

The screenshot shows the Chrome DevTools mobile emulation interface. The top bar indicates the device is an Apple iPhone 6 Plus in portrait mode, with no throttling applied. The network tab shows Mozilla/5.0 (iPhone; CPU). The main content area displays a three-column layout titled "Sports". The first column contains a thumbnail image and text about football. The second column contains a thumbnail image and text about basketball. The third column contains a thumbnail image and text about Formula One. The bottom of the screen shows the DevTools interface with tabs for Elements, Console, Sources, Network, Timeline, Profiles, and a search bar. The Elements tab is active, showing the current element selected: `body`. The Styles tab is also visible, showing the `element.style { margin: ... }` rule.

Sports

Football

 Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly

Basketball

 Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.^[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level

Formula One

 Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA). The FIA Formula One World Championship has been the premier form of racing since the inaugural season in

content is owned by Tech Career Booster <https://www.techcareerbooster.com>)

The page does not look good. This is because, on that device, the width is greater than 400px, and hence, the 3-columns layout is applied.

How can we fix that? You can guess that we probably have to change the breakpoint value. Instead of 400px, let's put that a little bit higher. To a value that would cover more mobile device widths. One good breakpoint is the 600px. Change your CSS to use 600px instead of 400px:

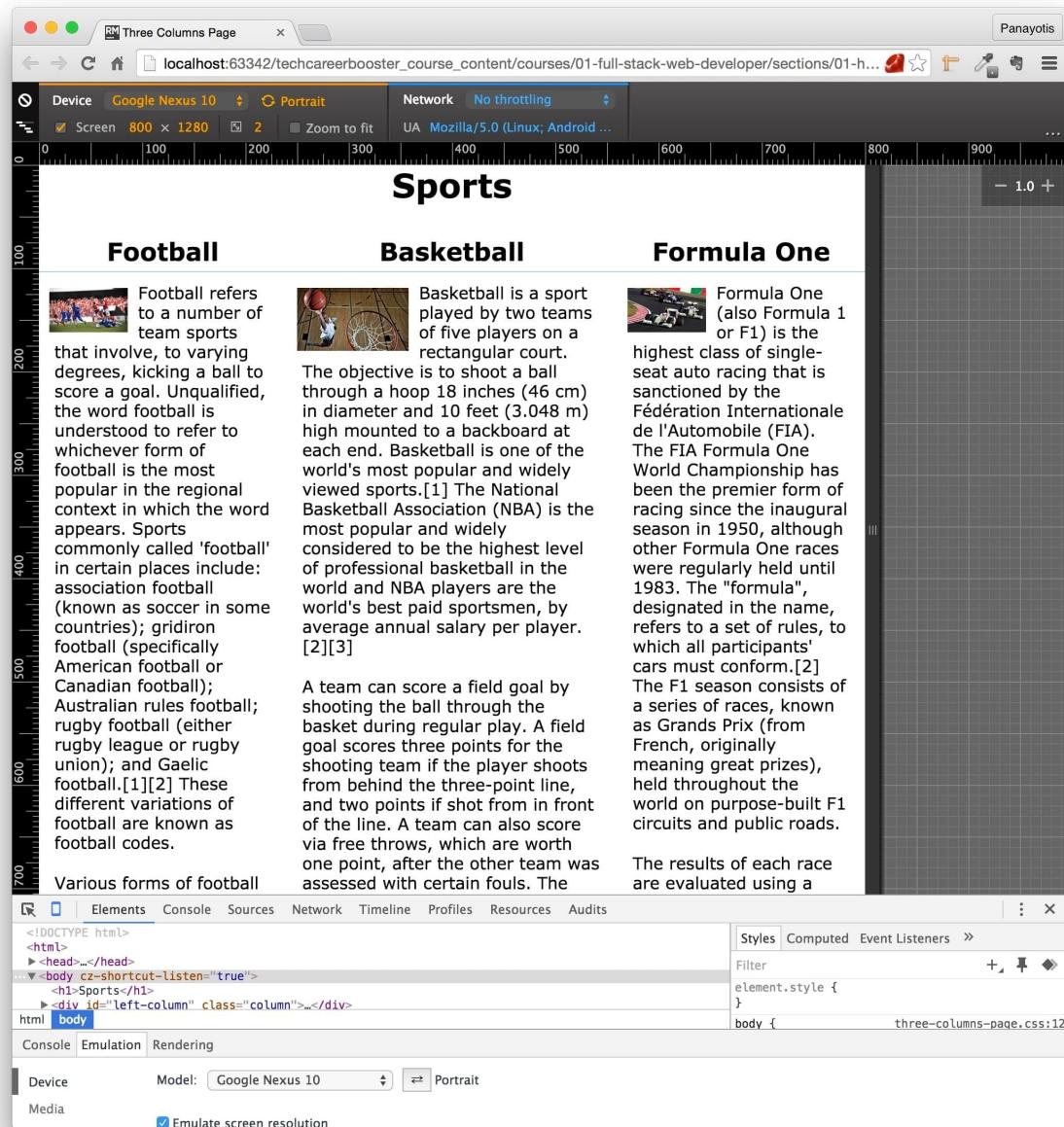
```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9     font-family: Verdana, Arial, sans-serif;
10}
11
12 body {
13     font-size: 1.6rem;
14}
15
16 .column {
17     float: left;
18}
19
20 #left-column, #middle-column, #right-column {
21     width: 100%;
22}
23
24 @media (min-device-width: 600px) {
25     #left-column,
26     #right-column {
27         width: 30%;
28     }
29
30     #middle-column {
31         width: 40%;
32     }
33 }
34
35 img {
36     float: left;
37     padding: 0.5rem 1rem;
38     width: 40%;
39 }
40
```

```
41 p {  
42     margin-bottom: 2rem;  
43     padding: 0 1.5rem;  
44 }  
45  
46 h1 {  
47     text-align: center;  
48     margin-bottom: 3rem;  
49 }  
50  
51 h2 {  
52     text-align: center;  
53     margin-bottom: 1rem;  
54     padding-bottom: 0.5rem;  
55     border-bottom: 1px solid lightblue;  
56 }
```

(the above code snippet online)

If you save the above and you load your page on iPhone 6 Plus, you will see that, now, the browser is using 1-column layout and the page looks good.

Now, try to load the page on “Google Nexus 10”, which has 800px width. You will see this:



Google Nexus 10 view of the Sports Page

As you can see, the columns are looking better, but still, it seems that 3-columns layout is not very good for this width size. What if we decided to have 2-columns layout for such devices? 2 columns may be looking better than 3 columns. Let's do the following: We are going to have 1-column layout for devices with width up to 600px, 2-columns layout for devices with width from 600px up to 1200px and 3-columns layout for devices with width 1200px and above.

In order to do that, you need to update your CSS code as follows:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9     font-family: Verdana, Arial, sans-serif;
10}
11
12 body {
13     font-size: 1.6rem;
14 }
15
16 .column {
17     float: left;
18 }
19
20 #left-column, #middle-column, #right-column {
21     width: 100%;
22 }
23
24 @media (min-device-width: 600px) {
25     #left-column, #middle-column, #right-column {
26         width: 50%;
27     }
28 }
29
30 @media (min-device-width: 1200px) {
31     #left-column,
32     #right-column {
33         width: 30%;
34     }
35
36     #middle-column {
37         width: 40%;
38     }
39 }
40
41 img {
42     float: left;
43     padding: 0.5rem 1rem;
44     width: 40%;
45 }
46
```

```

47 p {
48   margin-bottom: 2rem;
49   padding: 0 1.5rem;
50 }
51
52 h1 {
53   text-align: center;
54   margin-bottom: 3rem;
55 }
56
57 h2 {
58   text-align: center;
59   margin-bottom: 1rem;
60   padding-bottom: 0.5rem;
61   border-bottom: 1px solid lightblue;
62 }
```

(the above code snippet online)

Do you see the changes that have to do with the column widths? These are the following:

```

1 #left-column, #middle-column, #right-column {
2   width: 100%;
3 }
4
5 @media (min-device-width: 600px) {
6   #left-column, #middle-column, #right-column {
7     width: 50%;
8   }
9 }
10
11 @media (min-device-width: 1200px) {
12   #left-column,
13   #right-column {
14     width: 30%;
15   }
16
17   #middle-column {
18     width: 40%;
19   }
20 }
```

(the above code snippet online)

As you can read above, and following the mobile first approach, for mobile devices less than 600px, we are using 1-column layout. For devices with width from 600px up to 1199px, we are using a 2-columns layout and for devices with width 1200px and above, we are using 3-columns layout.

Save the above file and load your page on different devices:

- 1) When loading on iPhone 6:

The screenshot shows the Chrome DevTools mobile emulation interface. The top bar indicates the device is set to 'Apple iPhone 6' in 'Portrait' orientation, with a screen size of 375 x 627 pixels. The network settings are 'No throttling' and the user agent is Mozilla/5.0 (iPhone; CPU iPhone OS 11_0 like Mac OS X). The main content area displays a 'Sports' page with a 'Football' section. The football section includes a small image of a game and a detailed text description. The bottom portion of the interface shows the DevTools elements panel with the 'body' tab selected, displaying the 'Styles' tab and a preview of the 'margin' property being edited.

Sports

Football



Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.^{[1][2]} These different variations of football are known as football codes.

Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.^{[3][4]} The expanse of the British Empire allowed these rules of football to spread to areas

Device content is owned by Tech Career Booster (<https://www.techcareerbooster.com>)

Media Emulate screen resolution

As you can see, the 1-column layout is applied on this device.

2) When loading on Google Nexus 10:

Sports

Football



Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.[1][2] These different variations of football are known as football codes.

Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.[3][4] The expanse of the British Empire allowed these rules of football to spread to areas of British influence outside of the directly controlled Empire,[5] though by the end of the nineteenth century, distinct regional codes were already developing: Gaelic football, for example, deliberately incorporated the rules of local traditional

Basketball



Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level of professional basketball in the world and NBA players are the world's best paid sportsmen, by average annual salary per player.[2][3]

A team can score a field goal by shooting the ball through the basket during regular play. A field goal scores three points for the shooting team if the player shoots from behind the three-point line, and two points if shot from in front of the line. A team can also score via free throws, which are worth one point, after the other team was assessed with certain fouls. The team with the most points at the end of the game wins, but additional time (overtime) is issued when the score is tied at the end of regulation. The ball can be advanced on the court by bouncing it while walking or running or throwing it to a teammate. It is a violation to lift or drag

2-column Layout For Google Nexus 10

As you can see, the 2-columns layout is applied on this device.

3) When loading on desktop computer screen (switch off the developer tools):

Sports

| Football | Basketball | Formula One |
|--|---|--|
|  <p>Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.[1][2] These different variations of football are known as football codes.</p> <p>Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.[3][4] The expanse of the British Empire allowed these rules of football to spread to areas of British influence outside of the directly controlled Empire,[5] though by the end of the nineteenth century, distinct regional codes were already developing: Gaelic football, for example, deliberately incorporated the rules of local traditional football games in order to maintain their heritage.[6] In 1888, The Football League was founded in England, becoming the first of many professional football competitions. During the twentieth century, several of the various kinds of football grew to become some of the most popular team sports in the world</p> |  <p>Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level of professional basketball in the world and NBA players are the world's best paid sportsmen, by average annual salary per player. [2][3]</p> <p>A team can score a field goal by shooting the ball through the basket during regular play. A field goal scores three points for the shooting team if the player shoots from behind the three-point line, and two points if shot from in front of the line. A team can also score via free throws, which are worth one point, after the other team was assessed with certain fouls. The team with the most points at the end of the game wins, but additional time (overtime) is issued when the score is tied at the end of regulation. The ball can be advanced on the court by bouncing it while walking or running or throwing it to a teammate. It is a violation to lift or drag one's pivot foot without dribbling the ball, to carry it, or to hold the ball with both hands then resume dribbling.</p> |  <p>Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA). The FIA Formula One World Championship has been the premier form of racing since the inaugural season in 1950, although other Formula One races were regularly held until 1983. The "formula", designated in the name, refers to a set of rules, to which all participants' cars must conform.[2] The F1 season consists of a series of races, known as Grands Prix (from French, originally meaning great prizes), held throughout the world on purpose-built F1 circuits and public roads.</p> <p>The results of each race are evaluated using a points system to determine two annual World Championships, one for drivers, one for constructors. The racing drivers are required to be holders of valid Super Licences, the highest class of racing licence issued by the FIA.[3] The races are required to be held on tracks graded 1 (formerly A), the highest grade a track can receive by the FIA.[3] Most events are held in rural locations on purpose-built tracks, but there are several events in city centres throughout the world, with the Monaco Grand Prix being the most obvious and famous example.</p> |

3-column Layout For Desktop Display

You can see here the 3-columns layout.

The above proves that we have correctly settled 2 breakpoints. The 600px and the 1200px breakpoints. These are used to switch layout. Our Web page is now responsive and it is displayed nice on all width devices.

Device VS Viewport

However, there is a problem with the current implementation. If you have your developer tools switched off and you just try to shrink the browser width to widths less than 1200px, then the layout remains 3-columns and does not switch to 2-columns:

Three Columns Page

Panayotis

localhost:63342/techcareerbooster_c...

Sports

Football

 Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.^[1] ^[2] These different variations of football are known as football codes.

Various forms of football can be identified in

Basketball

 Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.^[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level of professional basketball in the world and NBA players are the world's best paid sportsmen, by average annual salary per player.^{[2][3]}

A team can score a field goal by shooting the ball through the basket during regular play. A field goal scores three points for the shooting team if the player shoots from behind the three-point line, and two points if shot from in front of the line. A team can also score via free throws, which are worth one point, after the other team was assessed with certain fouls. The team with the most points at the end of the game wins, but additional time (overtime) is issued

Formula One

 Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA). The FIA Formula One World Championship has been the premier form of racing since the inaugural season in 1950, although other Formula One races were regularly held until 1983. The "formula", designated in the name, refers to a set of rules, to which all participants' cars must conform.^[2] The F1 season consists of a series of races, known as Grands Prix (from French, originally meaning great prizes), held throughout the world on purpose-built F1 circuits and public roads.

The results of each race are

This is due to the fact that we have used media queries that use device related media features and not viewport related media features. Remember that `min-device-width` refers to the minimum size of the device width. Instead of using the device related features, use the viewport related features. So, instead of `min-device-width` use `min-width`. Let's do that:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9     font-family: Verdana, Arial, sans-serif;
10 }
11
12 body {
13     font-size: 1.6rem;
14 }
15
16 .column {
17     float: left;
18 }
19
20 #left-column, #middle-column, #right-column {
21     width: 100%;
22 }
23
24 @media (min-width: 600px) {
25     #left-column, #middle-column, #right-column {
26         width: 50%;
27     }
28 }
29
30 @media (min-width: 1200px) {
31     #left-column,
32     #right-column {
33         width: 30%;
34     }
35
36     #middle-column {
37         width: 40%;
38     }
39 }
40
41 img {
```

```
42     float: left;
43     padding: 0.5rem 1rem;
44     width: 40%;
45 }
46
47 p {
48     margin-bottom: 2rem;
49     padding: 0 1.5rem;
50 }
51
52 h1 {
53     text-align: center;
54     margin-bottom: 3rem;
55 }
56
57 h2 {
58     text-align: center;
59     margin-bottom: 1rem;
60     padding-bottom: 0.5rem;
61     border-bottom: 1px solid lightblue;
62 }
```

(the above code snippet online)

If you save the above file and reload the page on your browser, you will see that when you shrink the browser window, it will use 2-columns layout. And if you shrink it below the 600px width, it will automatically turn to 1-column layout.

The page changes layout while shrinking the browser window

Perfect. We have managed to make our page responsive, changing layout at 2 breakpoints, based on the browser window.

Fixing the third section

There is, still, one problem left to be corrected. Do you see that on the 2-columns layout?

Sports

| Football | Basketball |
|--|--|
|  <p>Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.^{[1][2]} These different variations of football are known as football codes.</p> <p>Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.^{[3][4]} The expanse of the British Empire allowed these rules of football to spread to areas of British influence outside of the directly controlled Empire,^[5] though by the end of the nineteenth century, distinct regional codes were already developing: Gaelic football, for example, deliberately incorporated the rules of local traditional football games in order to maintain their heritage.^[6] In 1888, The Football League was founded in England, becoming the first of many professional football competitions. During the twentieth century, several of the various kinds of football grew to become some of the most popular team sports in the world</p> |  <p>Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.^[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level of professional basketball in the world and NBA players are the world's best paid sportsmen, by average annual salary per player.^{[2][3]}</p> <p>A team can score a field goal by shooting the ball through the basket during regular play. A field goal scores three points for the shooting team if the player shoots from behind the three-point line, and two points if shot from in front of the line. A team can also score via free throws, which are worth one point, after the other team was assessed with certain fouls. The team with the most points at the end of the game wins, but additional time (overtime) is issued when the score is tied at the end of regulation. The ball can be advanced on the court by bouncing it while walking or running or throwing it to a teammate. It is a violation to lift or drag one's pivot foot without dribbling the ball, to carry it, or to hold the ball with both hands then resume dribbling.</p> |
| <h3>Formula One</h3>  <p>Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA). The FIA Formula One World Championship has been the premier form of racing since the inaugural season in 1950, although other Formula One races were regularly held until 1983. The "formula", designated in the name, refers to a set of rules, to which all participants' cars must conform.^[2] The F1 season consists of a series of races, known as Grands Prix (from French, originally meaning great prizes), held throughout the world on purpose-built F1 circuits and public roads.</p> | |

2-columns Layout Has Problem with 3rd Section

As you can see above, the 3rd section is displayed to the right. We have already learned how we can fix that. It is the `clear` CSS property that will come handy. We will set `clear: left;` for the `#right-column` on the CSS rules section that deal with the 2-columns layout:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9     font-family: Verdana, Arial, sans-serif;
10}
11
12 body {
13     font-size: 1.6rem;
14}
15
16 .column {
17     float: left;
18}
19
20 #left-column, #middle-column, #right-column {
21     width: 100%;
22}
23
24 @media (min-width: 600px) {
25     #left-column, #middle-column, #right-column {
26         width: 50%;
27     }
28     #right-column {
29         clear: left;
30     }
31 }
32
33 @media (min-width: 1200px) {
34     #left-column,
35     #right-column {
36         width: 30%;
37     }
38
39     #middle-column {
40         width: 40%;
41     }
42 }
43
44 img {
45     float: left;
46     padding: 0.5rem 1rem;
```

```
47     width: 40%;  
48 }  
49  
50 p {  
51     margin-bottom: 2rem;  
52     padding: 0 1.5rem;  
53 }  
54  
55 h1 {  
56     text-align: center;  
57     margin-bottom: 3rem;  
58 }  
59  
60 h2 {  
61     text-align: center;  
62     margin-bottom: 1rem;  
63     padding-bottom: 0.5rem;  
64     border-bottom: 1px solid lightblue;  
65 }
```

(the above code snippet online)

If you save the above and reload the page on your browser, you will see this:

RM Three Columns Page

localhost:63342/techcareerbooster_course_content/courses/01-full-stack-web-developer... Panayotis

Sports

Football



Football refers to a number of team sports that involve, to varying degrees, kicking a ball to score a goal. Unqualified, the word football is understood to refer to whichever form of football is the most popular in the regional context in which the word appears. Sports commonly called 'football' in certain places include: association football (known as soccer in some countries); gridiron football (specifically American football or Canadian football); Australian rules football; rugby football (either rugby league or rugby union); and Gaelic football.^{[1][2]} These different variations of football are known as football codes.

Various forms of football can be identified in history, often as popular peasant games. Contemporary codes of football can be traced back to the codification of these games at English public schools during the nineteenth century.^{[3][4]} The expanse of the British Empire allowed these rules of football to spread to areas of British influence outside of the directly controlled Empire,^[5] though by the end of the nineteenth century, distinct regional codes were already developing: Gaelic football, for example, deliberately incorporated the rules of local traditional football games in order to maintain their heritage.^[6] In 1888, The Football League was founded in England, becoming the first of many professional football competitions. During the twentieth century, several of the various kinds of football grew to become some of the most popular team sports in the world

Basketball



Basketball is a sport played by two teams of five players on a rectangular court. The objective is to shoot a ball through a hoop 18 inches (46 cm) in diameter and 10 feet (3.048 m) high mounted to a backboard at each end. Basketball is one of the world's most popular and widely viewed sports.^[1] The National Basketball Association (NBA) is the most popular and widely considered to be the highest level of professional basketball in the world and NBA players are the world's best paid sportsmen, by average annual salary per player.^{[2][3]}

A team can score a field goal by shooting the ball through the basket during regular play. A field goal scores three points for the shooting team if the player shoots from behind the three-point line, and two points if shot from in front of the line. A team can also score via free throws, which are worth one point, after the other team was assessed with certain fouls. The team with the most points at the end of the game wins, but additional time (overtime) is issued when the score is tied at the end of regulation. The ball can be advanced on the court by bouncing it while walking or running or throwing it to a teammate. It is a violation to lift or drag one's pivot foot without dribbling the ball, to carry it, or to hold the ball with both hands then resume dribbling.

Formula One



Formula One (also Formula 1 or F1) is the highest class of single-seat auto racing that is sanctioned by the Fédération Internationale de l'Automobile (FIA). The FIA Formula One World Championship has been the premier form of racing since the inaugural season in 1950, although other Formula One races were regularly held until 1983. The "formula", designated in the name, refers to a set of rules to which all participants' cars must

Sports Page 2-columns Layout Displays Properly the 3rd Section

As you can see, the 3rd section now is displayed correctly. The `clear: left;` set on the 3rd section (`#right-column`) div instructed the browser to place the div in such a place that no floating elements exist to the left of it. Hence, browser decides to draw the `#right-column` div on its own horizontal line.

Loading CSS files with media query

We have seen in the previous example, that we can have our media query mixed-in inside the main CSS file. But this is not necessary. You can have your special CSS rules, that are applied under specific media conditions, stored in different files, for better organization of your CSS code.

So, instead of having only this:

```
1 <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
```

(the above code snippet online)

you can have:

```
1 <link rel="stylesheet" href="stylesheets/main.css" type="text/css">
2 <link rel="stylesheet" media="(min-width: 600px)" href="stylesheets/min-width-600\
.css" type="text/css">
4 <link rel="stylesheet" media="(min-width: 1200px)" href="stylesheets/min-width-12\
00.css" type="text/css">
```

(the above code snippet online)

In the above case, the file `min-width-600.css` would have normal CSS content without media queries. Same goes for `min-width-1200.css`. But, those files CSS rules would be applied only if the corresponding media conditions specified would be true.

Responsive Images

Another HTML element that you need to take care of is the one used for images. Usually you should not set specific size for the images, but, instead, you should allow the image to shrink to the width of its container. Let's see the following example:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8"/>
5     <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6     <title>Responsive Images</title>
7     <link rel="stylesheet" href="stylesheets/main.css" type="text/css"/>
8   </head>
9   <body>
10    
11  </body>
12 </html>
```

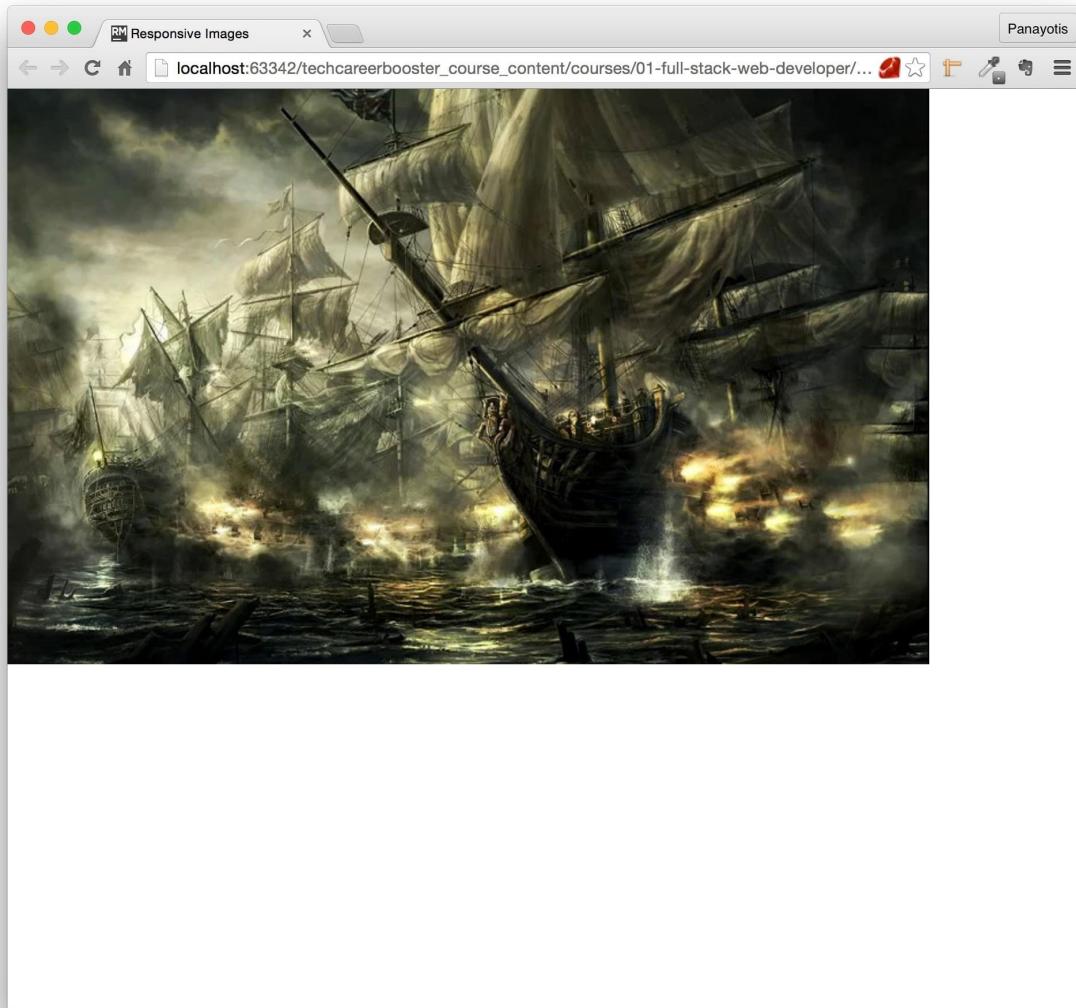
(the above code snippet online)

This is a very simple HTML page. It contains only an image. The size of the image is 800x499.
The CSS is very simple:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13 }
```

(the above code snippet online)

If you save the above files and load the page on your browser, you will see this:



Simple Page with an Image

This page does not work well when the browser width shrinks. And, of course, it does not work on mobile devices. Try that. Shrink the width of the browser window. You will see that the image is cropped.



Page is Not Responsive - Image is Cropped

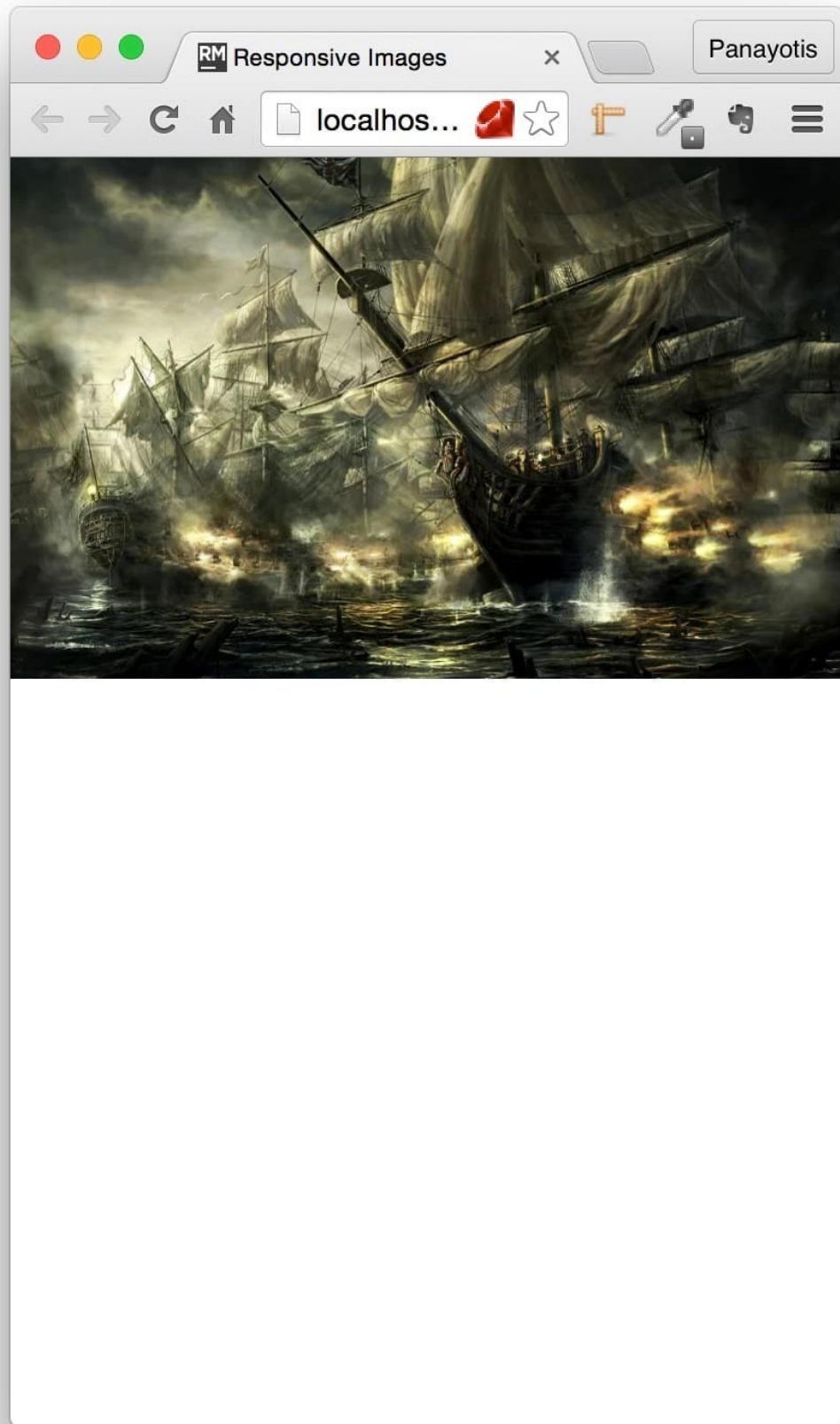
As you can see above, the image is cropped. This Web page is not responsive. There is a simple trick that you can do. Set the width of the image to be 100% and do not specify the height. The height will be auto by default. This will make sure that the image is resized according to the

browser window width. This should be the CSS file:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13 }
14
15 img {
16     width: 100%;
17 }
```

(the above code snippet online)

Save the above CSS file and reload the page on a narrow width browser window. You will see this:



Do you see that the image is displayed completely. It is not cropped. Its width size is equal to the size of its container. This is what the 100% does. There is a small gotcha here that you need to be aware of. Setting the width of the image to 100% of the size of its container, hence always being equal to the size of its container will make the image scale up when the container has width which is greater than the width of the image. Try to resize the browser window so that its width is more than 800px. Set it to 1200px or even more. You can also maximize the browser window. You will see the image scaled up. This might not be something that you would like to happen. How can you avoid that? How can you make the image shrink small but not scale up more than its original size, or other specific size? You can set the property `max-width` to be equal to the size of the image or to other size that makes sense to you, according to your use case. Let's do that for our example page:

```
1 * {
2     box-sizing: border-box;
3     padding: 0;
4     margin: 0;
5 }
6
7 html {
8     font-size: 62.5%;
9 }
10
11 body {
12     font-size: 1.6rem;
13 }
14
15 img {
16     width: 100%;
17     max-width: 800px;
18 }
```

(the above code snippet online)

Save the above file and reload your page on a maximized browser window. You will see that the image is not scaled up to fill in the whole width of the browser window. It has width equal to 800px.

Responsive Videos

We have learned that the `video` tag can be used to display a video playback player. This can be responsive with the same techniques that we have described above about the images.

Responsive Frameworks

Luckily, there are some very good frameworks out there, which are also free, and which can be used from our HTML pages in order to make them responsive. It is not that we only have to

reference their code and do nothing beyond that. It is that they have done a lot of preparation work for us and we only have to learn their CSS rules so that we can quickly apply them to our pages and make them responsive.

One such framework is [Twitter Bootstrap](#), which is being taught in a later chapter. On that framework, if you want, for example, to make an image responsive, you only have to use the class `img-responsive`. Besides helping us make our pages responsive, such frameworks offer numerous other facilities to make us style our pages quickly.

Tasks

Before you continue, you may want to know that: You can sign up to [Tech Career Booster](#) and have a mentor evaluate your tasks, your quizzes and, generally, your progress in becoming a Web Developer. Or you can sign up and get access to Tech Career Booster Slack channel. In that channel, there are a lot of people that can answer your questions and give you valuable feedback.

Task:

Task details:

1. You need to implement a Web page that is responsive. The page should contain a series of images of Santorini hotels. Read further.
2. This is a screenshot of the Web page that we are asking you to build. The screenshot is from the Web page as it should look like on a mobile device:

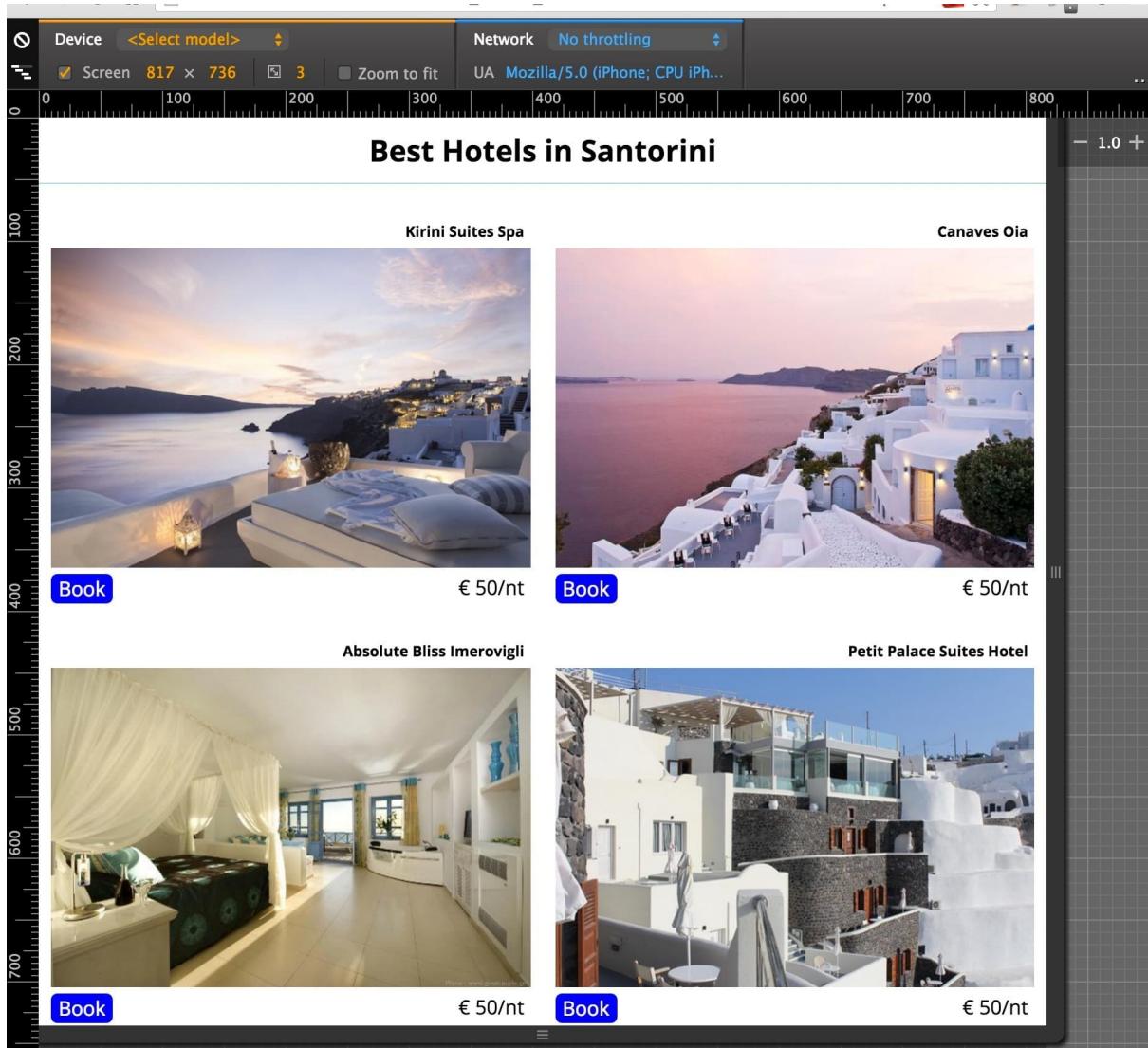
Screenshot of a browser developer tools window showing a mobile view of a website titled "Best Hotels in Santorini". The device is set to "Apple iPhone 6 Plus" in portrait mode. The network is set to "No throttling". The page displays two hotel listings:

- Kirini Suites Spa**: An image of a terrace overlooking the sea at sunset. A blue "Book" button is visible. The price is € 50/nt.
- Canaves Oia**: An image of white-washed buildings built into a cliffside overlooking the sea at sunset. A blue "Book" button is visible. The price is € 50/nt.

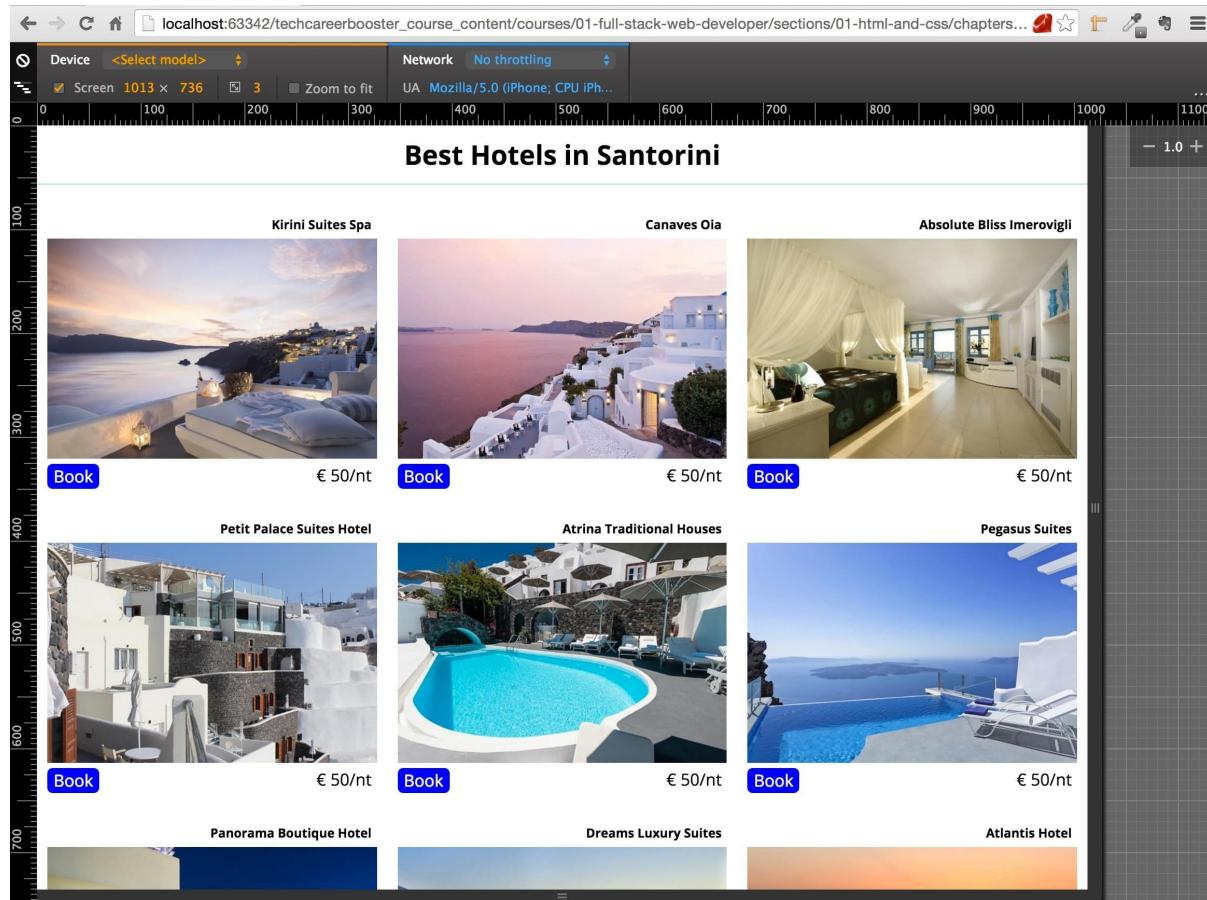
The browser's bottom navigation bar shows "Elements", "Console", "Sources", "Network", "Timeline", "Profiles", and other tabs. The "Styles" tab is selected, showing a CSS rule for "element.style" with "margin: 10" highlighted. The status bar at the bottom indicates the content is owned by Tech Career Booster (<https://www.techcareerbooster.com>).

As you can see, this is a 1-column layout. User can scroll to see the rest of the images:

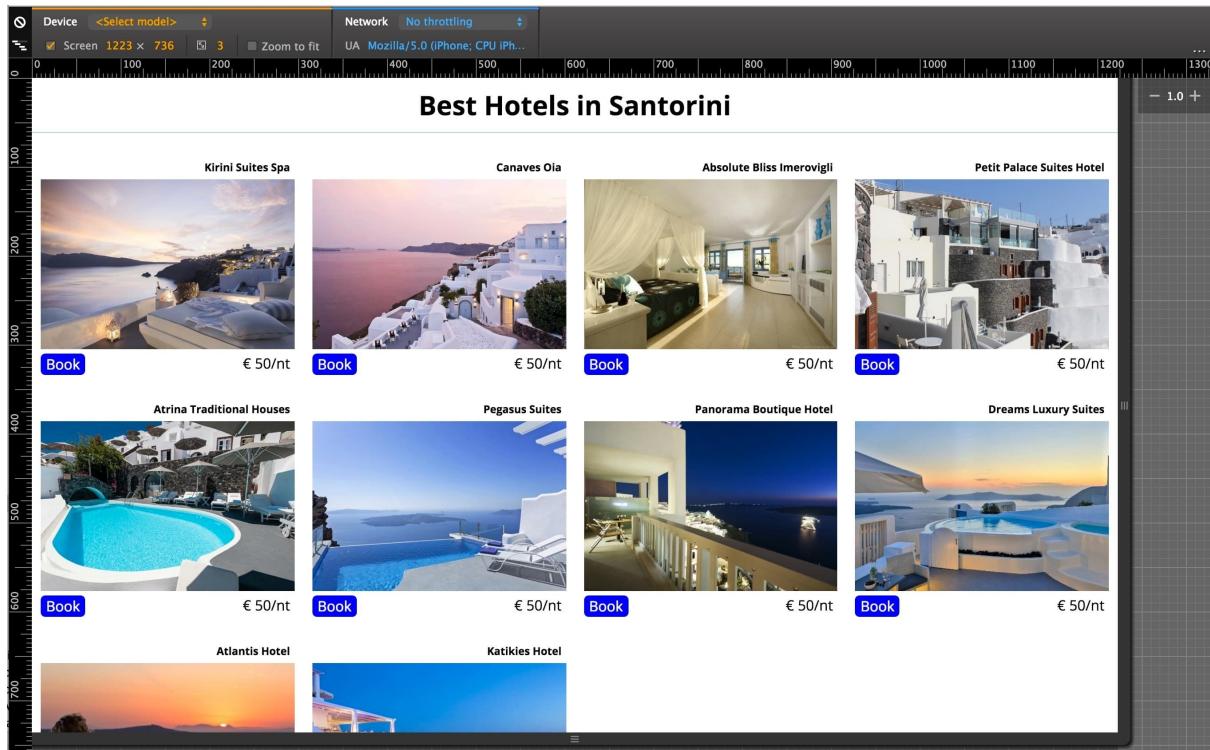
How the user can scroll to see all the hotel images 3. The Web page should be responsive with 3 breakpoints: 1. 768px 2. 992px 3. 1200px 4. On 768px and above, up to 991px, the page should be with 2-columns layout:



5. On 992px and above, up to 1199px, the page should be with 3-columns layout:



6. On 1200px and above, the page should be with 4-columns layout:



Responsive Page with Hotel Images - 4-columns layout

How the Web page changes layout as the device width increases 8. Here are some hints and requirements about this task: 1. Use the font “Open Sans” from Google Fonts. Make sure you include all the available styles. 2. There have to be 10 hotel images. So, 10 blocks of images with the button to book and the price. 3. Design with mobile-first approach. 4. Make sure that you start with CSS rules that reset the box-sizing to border-box and the padding and margin to 0 for all elements. 5. Make sure that the images are responsive. But do not limit their maximum width. 6. The “Book” button should actually be an anchor a styled as a button. Make sure that when the mouse goes over it, the background color and color change. 7. The top header should have responsive font size too. So, give a different font-size according to the breakpoint. For larger screens increase it. For smaller decrease it.