

# PeopleCert

## Software Development Skills

JavaScript Stream

Lessons 8-9-10

Study Guide



## Copyright Details

The contents of this workshop are protected by copyright and can be reproduced under the Terms of Use agreed between PeopleCert and the ATO using this material only.

Material in this presentation has been sourced from the bibliography listed in the certification's Syllabus. All software-related images are used for educational purposes only and may differ across time.

No part of this document may be reproduced in any form without the written permission of PeopleCert International Ltd. Permission can be requested at [www.peoplecert.org](http://www.peoplecert.org).

e-mail: [info@peoplecert.org](mailto:info@peoplecert.org), [www.peoplecert.org](http://www.peoplecert.org)

### **Copyright © 2017-2019 PeopleCert International Ltd.**

All rights reserved. No part of this publication may be reproduced or transmitted in any form and by any means (electronic, photocopying, recording or otherwise) except as permitted in writing by PeopleCert International Ltd. Enquiries for permission to reproduce, transmit or use for any purpose this material should be directed to the publisher.

### **DISCLAIMER**

This publication is designed to provide helpful information to the reader. Although every care has been taken by PeopleCert International Ltd in the preparation of this publication, no representation or warranty (express or implied) is given by PeopleCert International Ltd. as publisher with respect as to the completeness, accuracy, reliability, suitability or availability of the information contained within it and neither shall PeopleCert International Ltd be responsible or liable for any loss or damage whatsoever (indicatively but not limited to, special, indirect, consequential) arising or resulting of virtue of information, instructions or advice contained within this publication.)

- Document Version: PC-SDS\_SG 1.0 | February 2020

**PeopleCert**

All talents, certified.

# PeopleCert: A Global Leader in Certification



- ✓ **Web & Paper based exams in 25 languages**
- ✓ **2,500 Accredited Training Organisations** worldwide
- ✓ **Delivering exams across 200 countries every year**
- ✓ **Comprehensive Portfolio of 500+ Exams and Growing**



**PeopleCert**

All talents, certified.

© PeopleCert



## How to Use This Document

This document is your **PeopleCert Software Developer Skills Study Guide** to help you prepare for the **PeopleCert Software Developer Skills Foundation & Advanced examination**.

It is meant to provide you with a clear outline of everything covered in the course presentation by your instructor that will be on the PeopleCert Software Developer Skills Foundation & Advanced exams.

Your exams will be closed book. You will be given 120 minutes to complete it. It contains 100 multiple choice questions and to pass the exam you must achieve a grade of 65% or higher, or a minimum of 65/100 correct responses. For further details on your exam, including more information on question types and learning objectives, please refer to your course syllabus.

As you follow along, you may see that some material here is not replicated in the trainer presentation. This study guide includes questions, activities, knowledge checks, or other material in the presentation that are facilitated verbally by the instructor. It also does not contain content that is not examinable, but instead is designed to reinforce learning or add value to your course experience. It also provides valuable links and references, throughout the slides, which you can explore further to enhance your learning and understanding of the material provided in the study guide.

# Coding Bootcamp

## LESSONS 8-9-10

### Web Development Tools

---

#### Objectives:

- Web development
- Browsers
- HTTP (Hyper Text Transfer Protocol)
- HTML (Hyper Text Mark Up Language)
- CSS (Cascading Style Sheets)
- URL (Uniform Resource Locator)

---

#### Syllabus Items:

- 4.1 Key Concepts and Principles



# Syllabus

Category	Topic	Task
<b>FSD_4 Web Design and Development Fundamentals (Front-End)</b>	<b>4.1 Key Concepts and Principles</b>	4.1.1 Outline web development, Browsers, HTTP (Hyper Text Transfer Protocol), HTML (Hyper Text Mark Up Language), CSS (Cascading Style Sheets), URL (Uniform Resource Locator)
		4.1.2 Understand and use web development editors, tools and principles
		4.1.3 Understand and use basic design elements: Design concept, page(s) design, Home page layout, navigation bar



## Contents | Learning Objectives

- ✓ Outline web development, Browsers, HTTP (Hyper Text Transfer Protocol), HTML (Hyper Text Mark Up Language), CSS (Cascading Style Sheets), URL (Uniform Resource Locator)
- ✓ Understand and use web development editors, tools and principles
- ✓ Understand and use basic design elements: Design concept, page(s) design, Home page layout, navigation bar



# Web Development Tools

## Tools Installation

- **Web Development**
  - Visual Studio Code
  - Atom
  - Sublime Text
- **Browsing**
  - Mozilla Firefox
  - Google Chrome

## Web Development

### Definition:

**Web development** is a broad term for the work involved in developing a web site for the Internet (World Wide Web) or an intranet (a private network).

- Web development ranges from developing the simplest static single page of plain text to the most complex web-based internet applications (or just 'web apps') electronic businesses, and social network services
- Among web professionals, "web development" usually refers to the main non-design aspects of building web sites: writing markup and coding
- For larger organizations and businesses, web development teams consist of hundreds web developers and follow standard website development methods like Agile methodologies
- Front-end web developers deal with the layout and visuals of a website and code running on the client

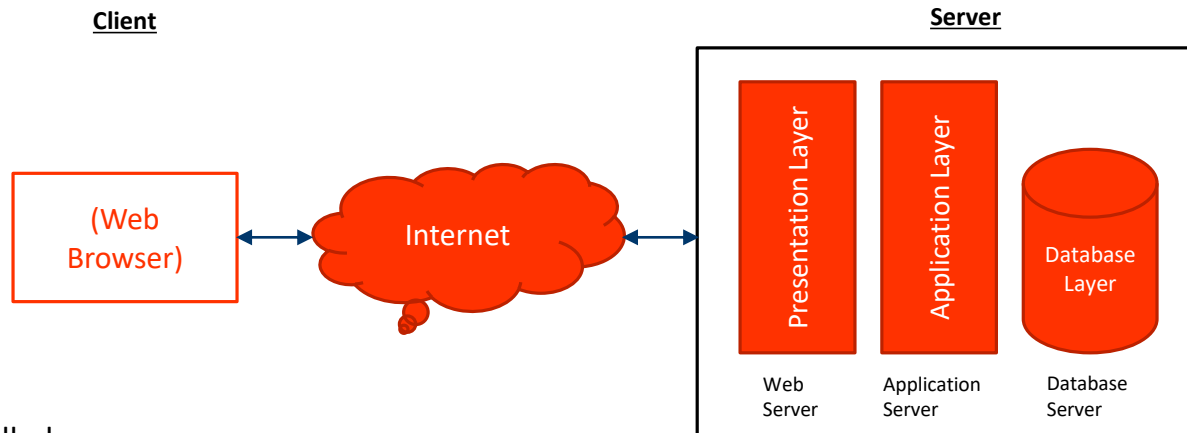
Source: [https://en.wikipedia.org/wiki/Web\\_development](https://en.wikipedia.org/wiki/Web_development)



# Architectural Design

## Definition:

IEEE defines **architectural design** as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.”



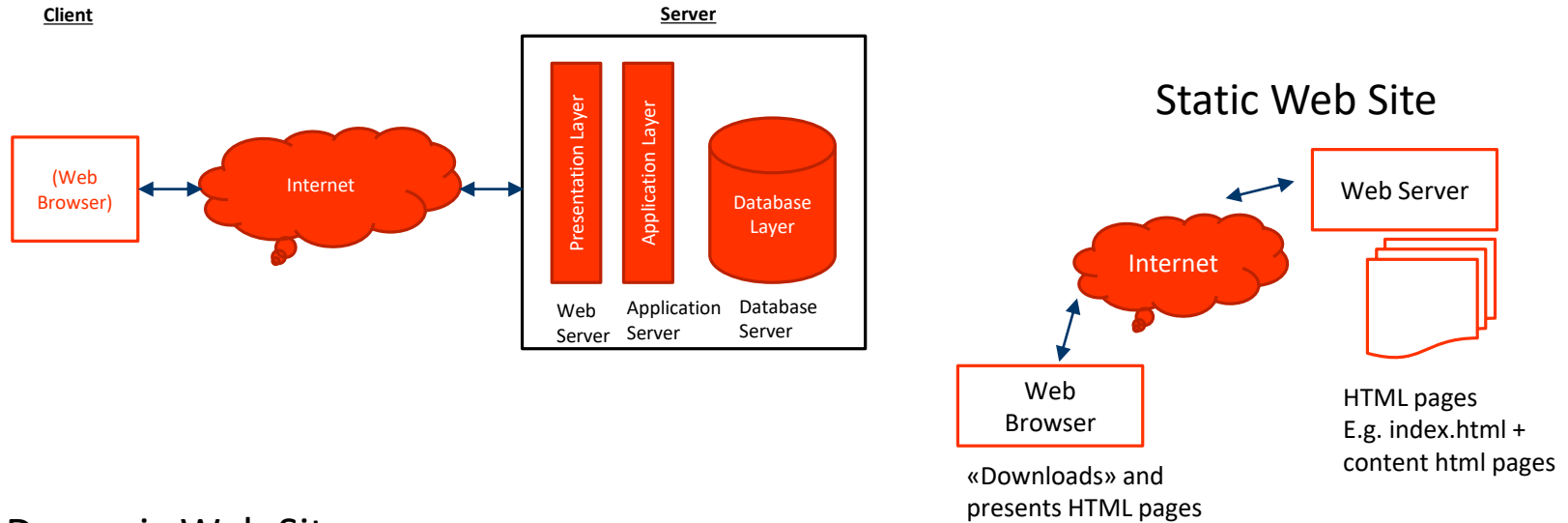
- Also called
  - **High Level Architecture**
  - **High-level view of a system**

## Detailed Software Design

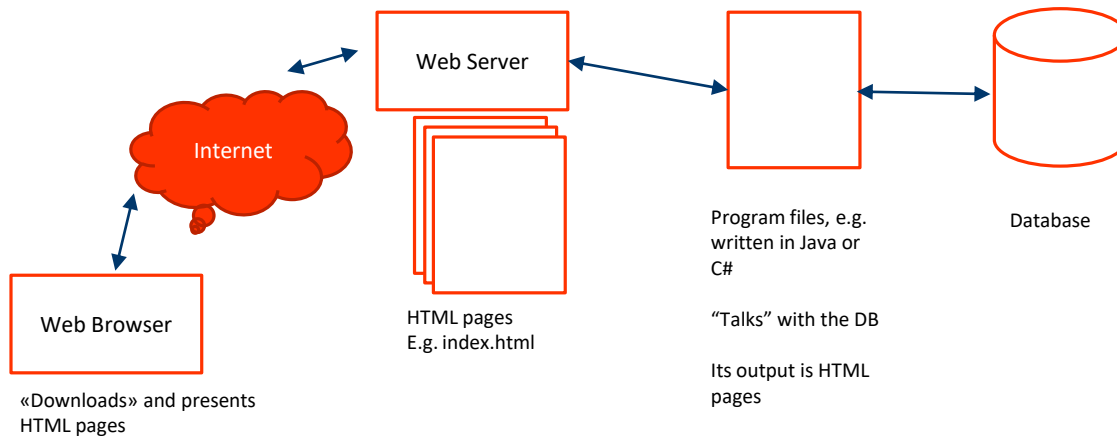
- Depicts relationships among software elements/ components at lower level
- **Examples:** UI design, class diagram, database schema, component diagram

Source: <https://www.geeksforgeeks.org/software-engineering-architectural-design/>

# Typical Web Application Architecture

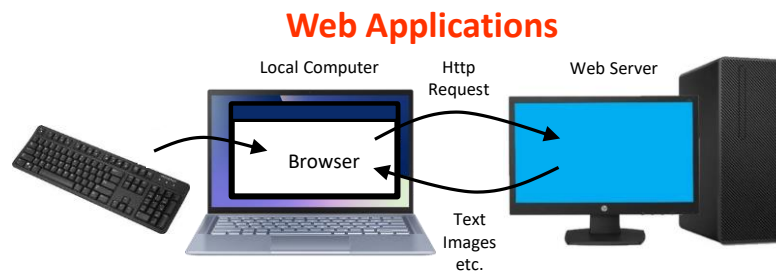
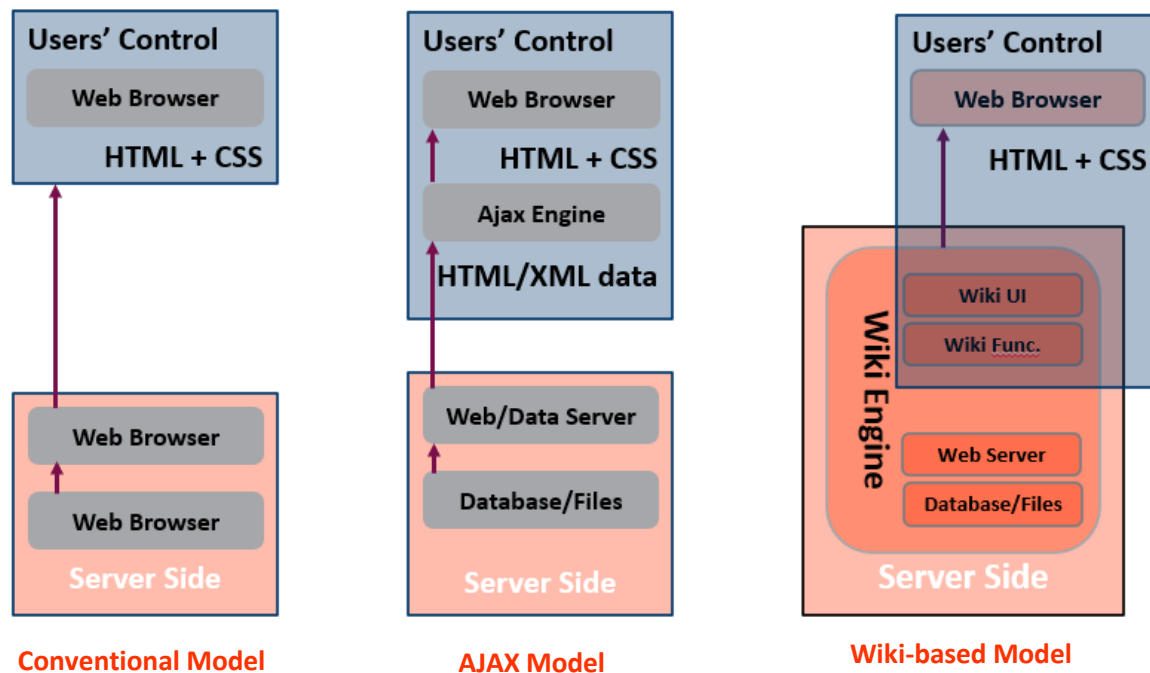


## Dynamic Web Site



Source: <https://www.geeksforgeeks.org/software-engineering-architectural-design/>

# Web Application Model



Source: <https://www.geeksforgeeks.org/software-engineering-architectural-design/>

# End-to-End Workflow Of A Web Request

**DNS Lookup:** The client tries to resolve the domain name for the request.

- Client sends DNS Query to local ISP DNS server.
- DNS server responds with the IP address for hostname.com

**Connect:** Client establishes TCP connection with the IP address of hostname.com

- Client sends SYN packet.
- Web server sends SYN-ACK packet.
- Client answers with ACK packet, concluding the three-way TCP connection establishment.

**Send:** Client sends the HTTP request to the web server.

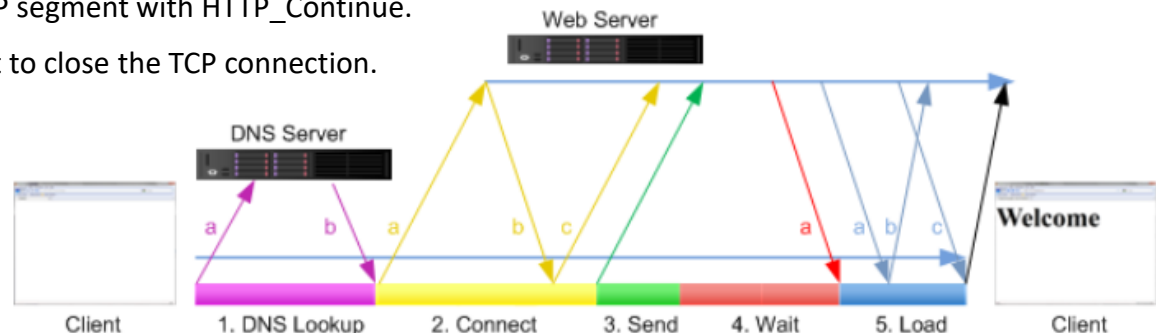
**Wait:** Client waits for the server to respond to the request.

- Web server processes the request, finds the resource, and sends the response to the Client. Client receives the first byte of the first packet from the web server, which contains the HTTP Response headers and content.

**Load:** Client loads the content of the response.

- Web server sends second TCP segment with the PSH flag set.
- Client sends ACK. (Client sends ACK every two segments it receives. from the host)
- Web server sends third TCP segment with HTTP\_Continue.

**Close:** Client sends a FIN packet to close the TCP connection.



Sources: <http://www.steves-internet-guide.com/dns-lookups/>  
<https://blog.catchpoint.com/2010/09/17/anatomyhttp/>

# Primary Protocols - HTTP (Hyper Text Transfer Protocol)

## Definition:

A **communications protocol** is a set of rules that must be obeyed by all users in a computer network.

- A data communications protocol deals with the *rules* for the transmission of data between two or more points (or *nodes*, as they may also be called).
- Internet protocol suite is the set of communication protocols that implement the protocol stack on which the internet runs. The Internet protocol suite is sometimes called the TCP/IP protocol suite, after TCP\IP, which refers to the important protocols in it, the Transmission Control Protocol(TCP) and the Internet Protocol(IP). The Internet protocol suite can be described by the analogy with the OSI model, but there are some differences. Also not all of the layers correspond well.
- So, in order for computers to communicate, they need to follow a common set of rules and instructions known as protocols
- **TCP/IP:** The Internet Protocol Suite, which is often referred to as TCP/IP (Transmission Control Protocol (TCP) and the Internet Protocol (IP), is the set of communications protocols used for the Internet. The three most common protocols are:
  - **Hypertext Transfer Protocol (HTTP)**  
HyperText Transfer Protocol is the protocol used to transfer data over the World Wide Web.
    - That is why Web site addresses often begin with http:// (or https:// for secure sites).
    - When you enter a URL into our browser and press ENTER, your computer sends an HTTP request to the appropriate Web server.
    - The Web server, which is designed to handle HTTP requests, then sends to us the requested HTML page over the Internet.
  - **File Transfer Protocol (FTP)**  
FTP is common method of transferring files via the Internet from a server to a client.

Sources: <https://www.sciencedirect.com/topics/computer-science/communication-protocols>  
[https://en.wikipedia.org/wiki/Internet\\_protocol\\_suite](https://en.wikipedia.org/wiki/Internet_protocol_suite)



# HTML (Hyper Text Mark Up Language)

- HTML is the standard markup language for creating Web pages.
- HTML stands for Hyper Text Markup Language
  - Hypertext describes the way by which pages are linked to one another
  - Markup language describes the structure of the page (and how it will be displayed)
- HTML **describes the structure** of Web pages using **markup**
- **HTML elements** are the building blocks of HTML pages
- **HTML elements** are represented by **tags**
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page
- HTML documents are just text files they can be written in even a simple text editor, even Notepad.
- HTML is a standard language, and it is maintained, together with other open web standards, by the World Wide Web Consortium (**W3C**)
- Benefits of W3C activities include:
  - Interoperability of websites across browsers - web pages can be viewed across a range of devices and browsers.
  - Enhanced accessibility - Improves search result rankings, reduces maintenance costs, and increases audience reach.
  - Consistent document type declarations - web pages can have a consistent look.

Sources: [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)  
<https://www.webopedia.com/TERM/H/HTML.html>  
[http://ism.dmst.aueb.gr/bootcamp/html\\_bootcamp.pdf](http://ism.dmst.aueb.gr/bootcamp/html_bootcamp.pdf)



## CSS (Cascading Style Sheets) | Benefits

### Definition:

**CSS** stands for **C**ascading **S**tyles **S**heets and is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

- Sets the content style and layout for a web page
- Can be applied to a single page, range of pages, or an entire website
- Describes the presentation of Web pages, including colors, layout, and fonts
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- Allows page adaptation to different types of devices
- CSS saves a lot of work. It can control the layout of multiple web pages **all at once**
- External stylesheets are stored in CSS files

### Benefits of CSS

- Ease of editing
- Faster loading of web pages
- Smaller file size
- Speed of development
- Standardization of design
- Ease of presenting different styles to different viewers

Sources: [https://www.w3schools.com/css/css\\_intro.aspx](https://www.w3schools.com/css/css_intro.aspx)  
[https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)



## Approaches to CSS Rules

There are three ways to embed CSS in a Web page:

**1) Using an external CSS file.** The file (a simple text file containing CSS rules) must have the extension .css and can be loaded in a Web page using the <link> HTML tag or using the @import rule inside another CSS file.

**2) Using the <style> HTML tag.** CSS rules can be placed inside the opening <style> and closing </style> tags, which in turn must be placed inside the <head> tag of the Web page:

```
<head>
```

```
<style>div { color: red; }</style>
```

```
...
```

```
</head>
```

**3) Using inline styles,** one can place specific CSS rules inside an HTML tag using the style attribute. For example: **<div style="color:red;">Red Text</div>.**

The use of inline styles is discouraged due to its inefficiency and other shortcomings.

Source: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)



# URL (Uniform Resource Locator)

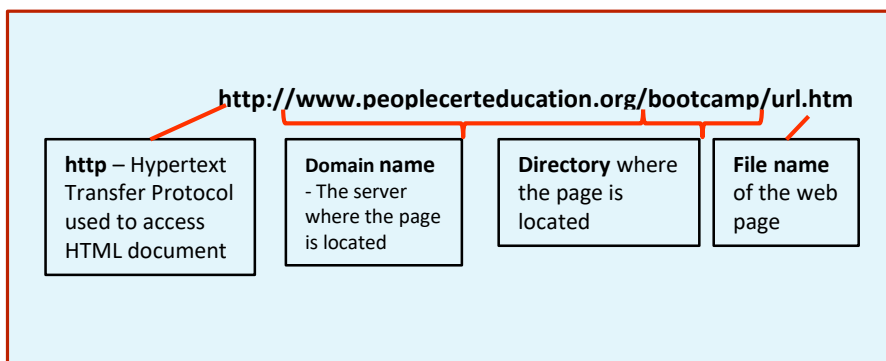
- A **URL** (Uniform Resource Locator) is the address of a resource (such as a document or website) on the Internet that consists of:
  - a communications protocol
  - the name or address of a computer on the network
  - and additional locating information
- Commonly termed as **web address**
- A URL is a specific type of Uniform Resource Identifier (URI), although many people use the two terms interchangeably

## URL | Examples:

- <http://www.yahoo.com/>
- <http://ecd1.org/>
- <ftp://info.apple.com/>

As we can see, not all URLs begin with "**http**".

- The first part of a URL indicates what kind of resource it is addressing. Two common prefixes are:
  - **http** - a hypertext directory or document (such as a Web page)
  - **ftp** - a directory of files or an actual file available to download
- The second part of a URL (after the "://") contains the address of the computer where the resource is located as well as the path to the file.



## URL - EXAMPLE

<http://www.peoplecerteducation.org/index.html>

Where

protocol (http), a hostname (www.peoplecerteducation.org), and a file name (index.html)

Also, the last part of the hostname can show:

- Type of organization
- Location/Country of organization

Source: <https://searchnetworking.techtarget.com/definition/URL>

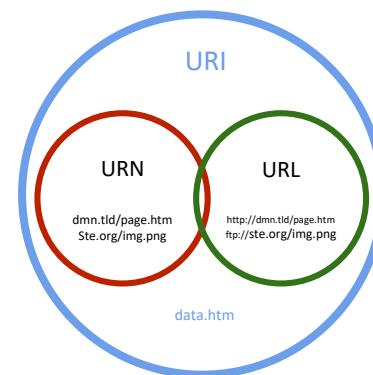
# URI | URL | URN | HTTP Network Protocol Stack

## Definitions:

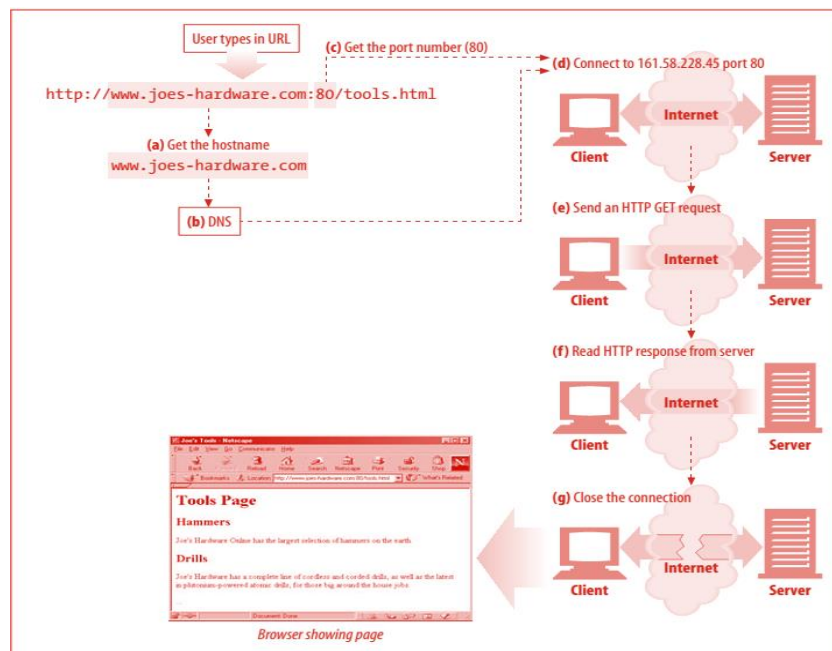
**Uniform Resource Identifier** URI is a text which is used to identify any resource or name on Internet.

**Uniform Resource Locator** URL includes location as well as the protocol to retrieve the resource (For example :HTTP Protocol is used to retrieve resource)

**Uniform Resource Name** URN is completely different than URL as it doesn't include any protocol.

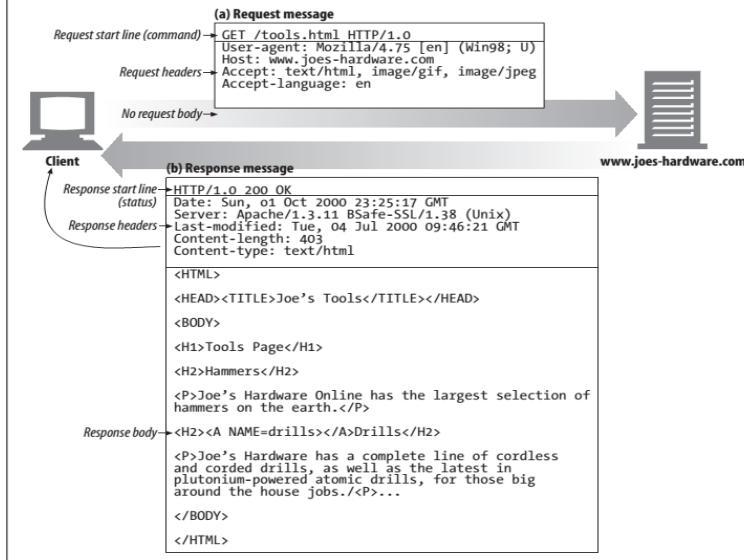


OSI Model	Internet Model	Internet Protocols
Application	Application	HTTP, HTTPS, SSH, DNS, SSL, FTP, POP3, SMTP, IMAP, Telnet, NNTP
Application		
Application		
Transport	Transport	TCP, UDP
Network	Network	IP, ICMP, ARP, DHCP
Datalink	Network Link	Ethernet, PPP, ADSL
Physical		



Sources: <https://www.oreilly.com/library/view/http-the-definitive/1565925092/ch01s06.html>  
<https://vichargrave.github.io/programming/tcp-ip-network-programming-design-patterns-in-cpp/>

# HTTP Request-Response



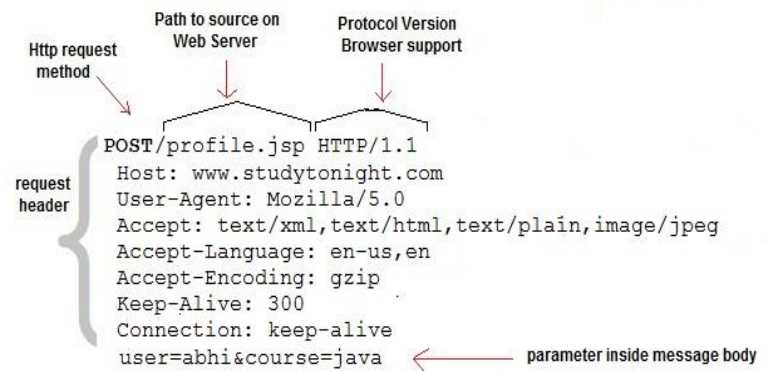
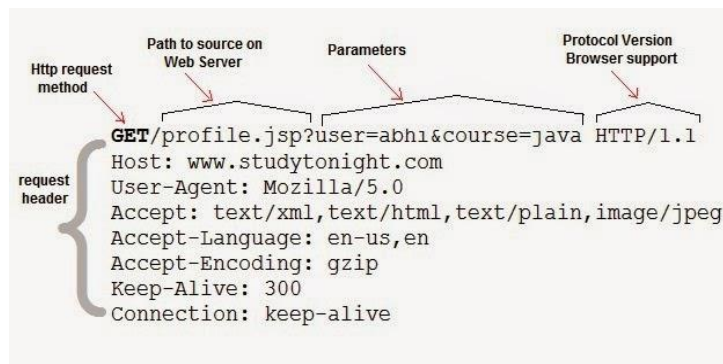
# Methods –Status Codes

HTTP method	Description
GET	Send named resource from the server to the client.
PUT	Store data from client into a named server resource.
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

HTTP status code	Description
200	OK. Document returned correctly.
302	Redirect. Go someplace else to get the resource.
404	Not Found. Can't find this resource.

## GET Request



Sources: <https://www.oreilly.com/library/view/http-the-definitive/1565925092/ch01s06.html>  
David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, Sailu Reddy: HTTP: The Definitive Guide



## Browsers

### Definition:

A **web browser** (or **browser**) is a software application for retrieving, presenting and traversing information resources on the World Wide Web (WWW)

- An **information resource** is identified by a **Uniform Resource Identifier (URI/URL)** that may be a web page, image, video or other piece of content.
- **Hyperlinks** present in resources enable users easily to navigate their browsers to related resources.
- Although browsers are primarily intended to use the World Wide Web, they can also be used to access information provided by web servers in private networks or files in file systems.

### Examples of Browsers:

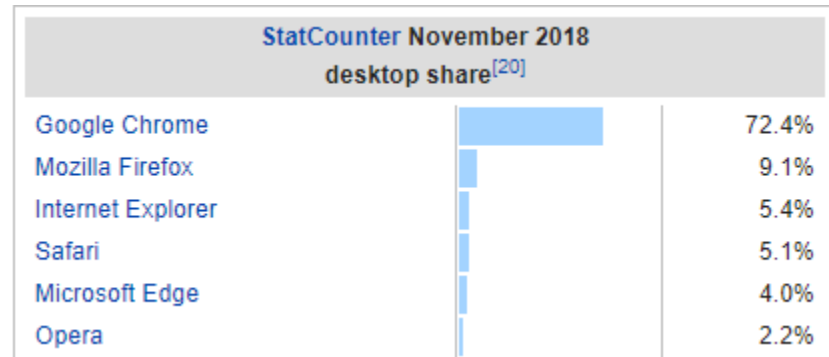
- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Internet Explorer
- Microsoft Edge
- Opera

Source: [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser)

## Examples of Browsers - Class Exercise1

Six browsers account for over 97% of users:

- Internet Explorer (or IE)
- Edge (replaced IE in Windows)
- Chrome
- Safari
- Opera
- Firefox



### Exercise 1

Show and discuss the http request – response using developer tools of a browser (Firefox-Chrome).  
Focus on return codes and method types

Source: [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser)

# Web Development Editors

- Web development editors are programs/application that help you create web pages or websites
- For many developers, a trusty text editor is all you need for even the most complex web applications.
- Popular Web development editors include:
  - VSCode (FREE) (Highly recommended)
  - SublimeText (FREE) (Highly recommended)
  - Atom (FREE)
  - KomodoEdit (FREE)

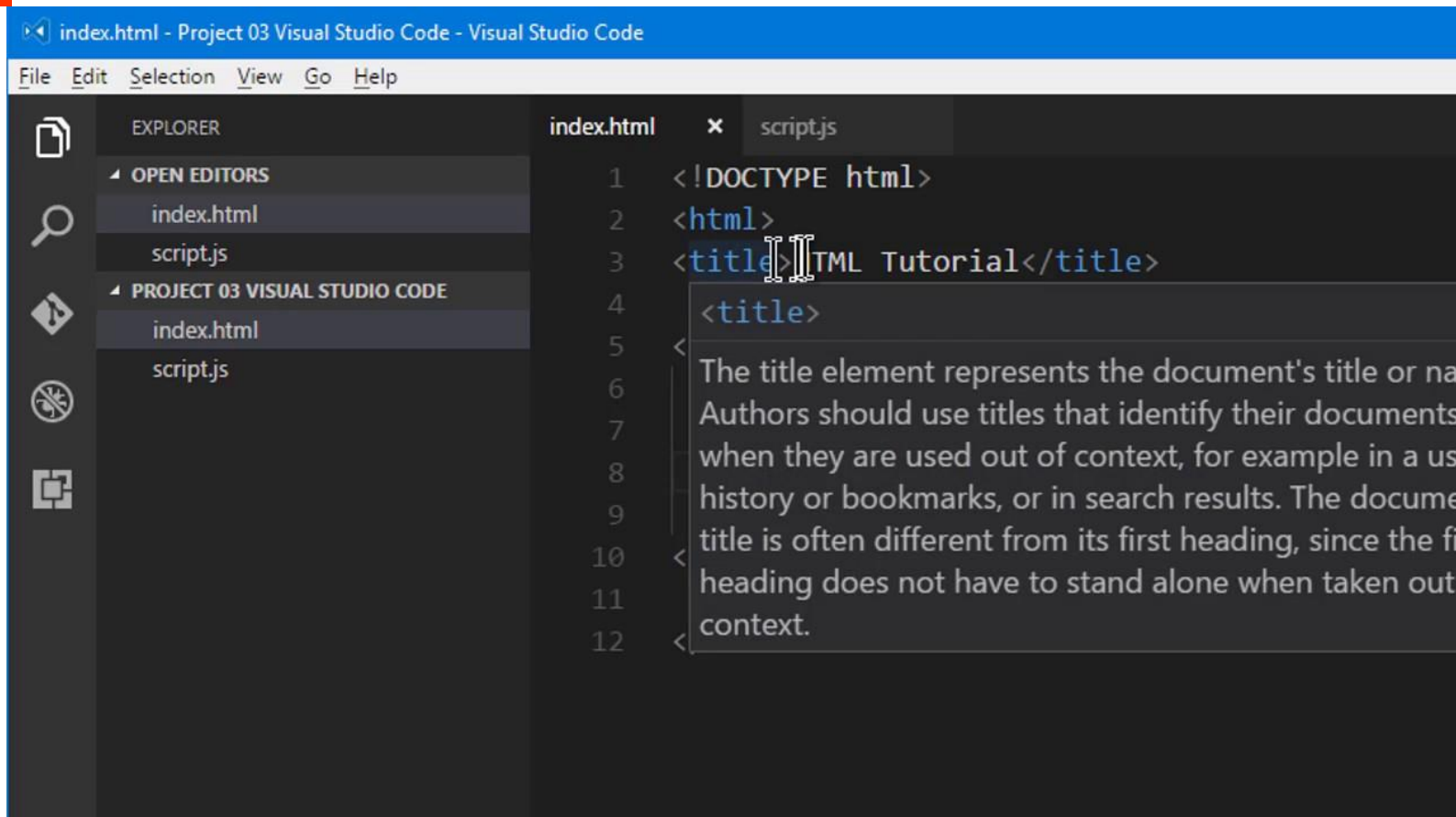
## Class Activity

- Open your code editor of choice
- Type:
 

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```
- Save as myfirstpage.html onto a convenient location
- Locate the file myfirstpage.html and double click it (it opens in your browser as a web page)
- Check out where everything is!
- Change the text My First Heading to anything you like (i.e. Mike's Website)
- Change the text My first paragraph to anything you like (i.e. Welcome to my site! I hope you like it!)
- Save as myfirstpage2.html onto a convenient location
- Locate the file myfirstpage2.html and double click it (it opens in your browser as a web page)
- Notice the changes

Source: <https://www.webfx.com/blog/web-design/the-15-most-popular-text-editors-for-developers/>

# Web Development Editors



*Any screen captures for **Visual Studio** Code editor are used for educational purposes only*



## Web Development Tools | Web Development Principles

### Web Development Tools

- **Web development tools** allow web developers to test and debug their code.
- They are different from website builders and integrated development environments (IDEs) in that they do not assist in the direct creation of a webpage, rather they are tools used for testing the user interface of a website or web application.
- Web development tools come as browser add-ons or built-in features in web browsers. The most popular web browsers today, Google Chrome, Firefox, Opera, Internet Explorer, UC browser and Safari , have **built-in tools** to help web developers, and many additional add-ons can be found in their respective plugin download centers.

### Web Development Principles

- **Focus on the User**
- **Focus on Quality**
- **Keep It Simple**
- **Think Long-Term**
  - looking into high quality content and services
  - creating designs that are usable and appealing
  - writing code to last
  - passionately reject “fire and forget.”
- **Don’t Repeat Yourself**
- **Know Your Field**

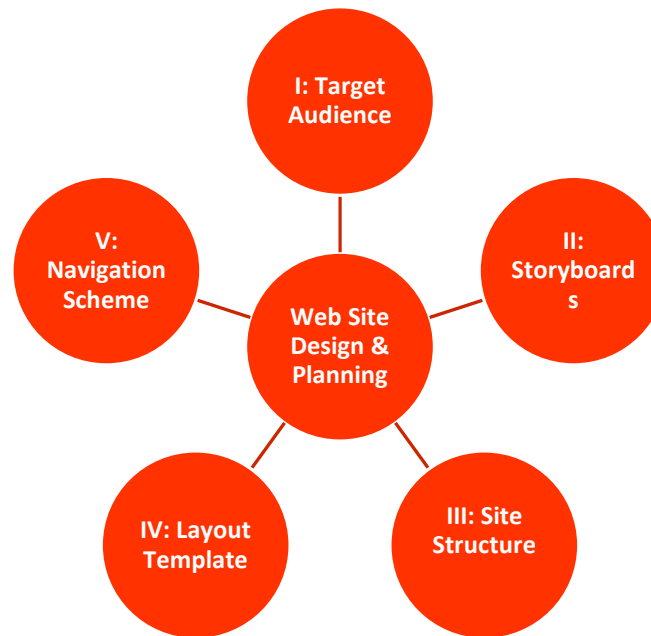
Source: [https://en.wikipedia.org/wiki/Web\\_development\\_tools](https://en.wikipedia.org/wiki/Web_development_tools)



## Planning a Web Site

- When thinking about creating a Web site, follow a series of **planning steps** to make sure our site is successful.
- Plan the site carefully in order to make sure everyone will be able to use it easily.
- Planning and design techniques such as evaluating the needs of the target audience, creating storyboards, organising the site structure, creating a page layout template and deciding on a navigation scheme are important when designing web sites

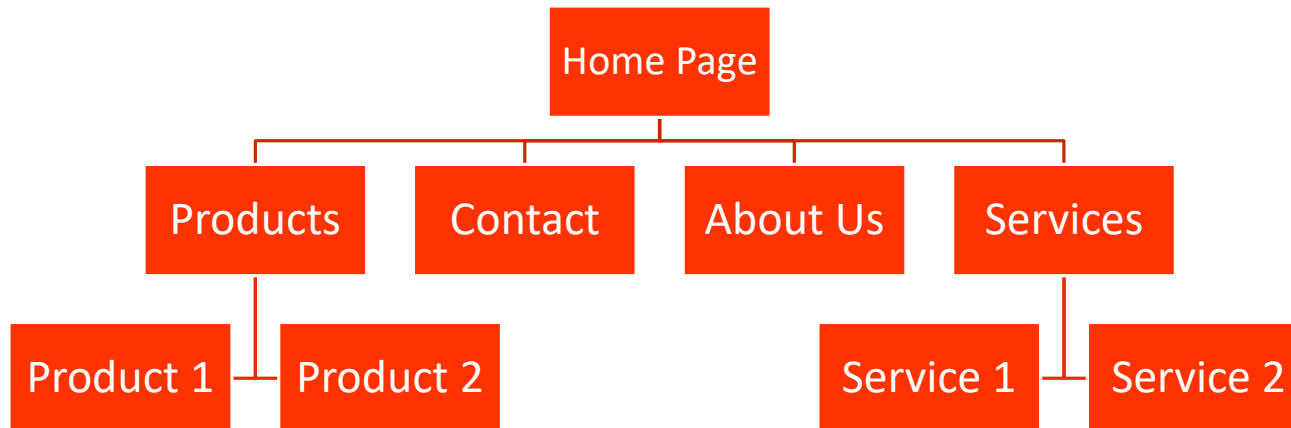
### Planning a Web Site & Design Steps



Source: <https://www.dreamtemplate.com/blog/web-design-tips/5-steps-for-the-process-of-design-and-development-of-a-website/>

# Factors Affecting Website Design

- **Target Audience:** Understand who you want to communicate with. Consider their age, interests, or motivation
- **Storyboards:** Think about the rough structure and design of the website by creating a storyboard (a sketch of the different components of a website, such as the main pages and the layout of the pages)
- **Site Structure:** Build the site structure (aka site map), which lists the main topic areas of the site, as well as sub-topics, and acts as a guide as to what content will be on the site. In addition, it allow you to plan how users will navigate through the site.



Adapted from <https://books.google.gr/books?id=m00EOqYkbcwC>

# Basic Design Elements

## Design Concept

- Web design is a constantly evolving field. As consumer needs change, as tests evolve, as technology advances, we in the web design and development industry are pushing ourselves to meet and exceed new needs, adapt, and deliver new experiences for clients and end-users.
- Think about the look and feel of your website. Create a detailed design of how your web pages should look. Although the content of the pages will of course differ, the pages should have a common theme, including colors, font, design, and layout. These criteria should be recorded in a **page layout template**.

## Page(s) Design/Layout



- Page layout is the part of web design that deals in the arrangement of visual elements on a page.
- It generally involves organizational principles of composition to achieve specific communication objectives

## Home Page Layout and Fonts

- The home page must be indicative of your web site and target the audience you are aiming for
- The home page must be interesting and appealing to the visitor
- Consider carefully which **font** you should use on your website.
- Usually san serif fonts (Arial, Courier, and Helvetica) are preferable for web pages as they display well in high and low resolution.
- Other considerations regarding fonts:
  - Use real text rather than text within graphics.
  - Use a limited number of fonts.
  - Ensure sufficient contrast between the text and the background.
  - Avoid small font sizes.

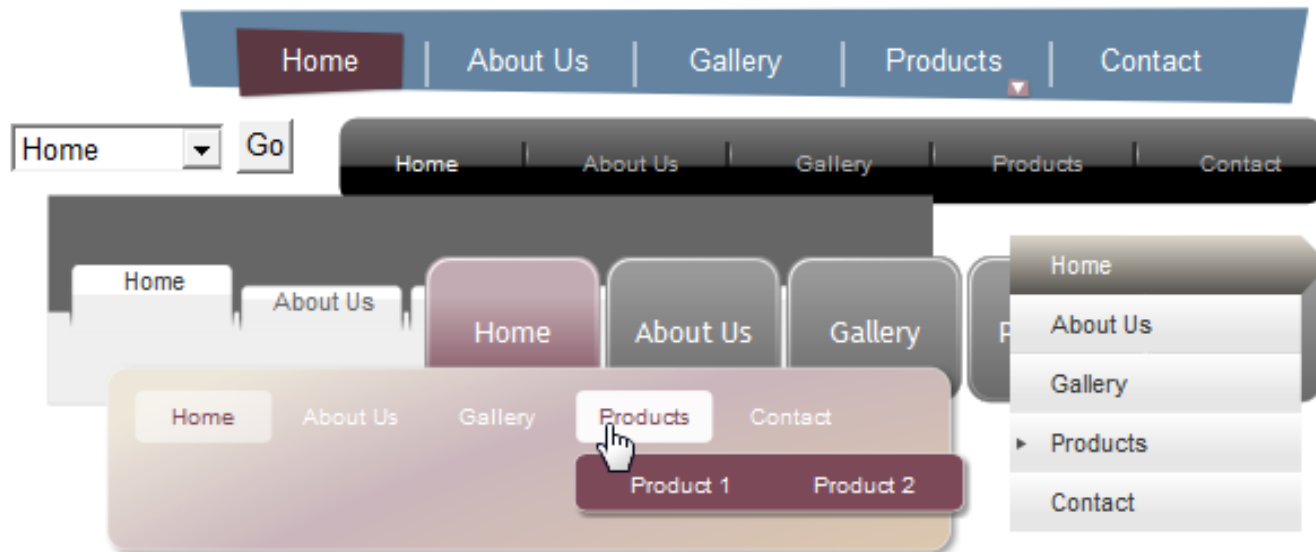
Source: <https://www.webdesign.org/web-design-basics/design-principles/page-1.html#ixzz4zdkyGUFj>

## Basic Design Elements

### Navigation Bar:

#### Definition:

A **navigation bar** is the section of a web page intended to help visitors navigate through the available web pages and page sections.



Source: <https://www.webdesign.org/web-design-basics/design-principles/page-1.html#ixzz4zdkyGUFj>



## Class Exercise

Visit a URL – i.e. [www.afdemp.org](http://www.afdemp.org) or any website you prefer:

- Right click on the page and select VIEW SOURCE
- Try to identify:
  - Home page layout elements
  - Home page format elements
  - Navigation elements
  - Design elements

Write down:

- Home page layout elements

---

---

- Home page format elements

---

---

- Navigation elements

---

---

- Design elements

---

---



**It is time for a Knowledge Check!**



## Sample Questions

1. What does HTML stand for?
  - A. High Text Marking Language
  - B. Hyper Text Mark Up Language
  - C. High Text Mark Up Layout
  - D. Hyper Text Mark Up Layout
  
2. What does www.cepis.org denote in the address http://www.cepis.org/media/pressr1.pdf ?
  - A. Address or domain name of the host computer
  - B. Geo-location of the organization
  - C. The type of the organization using this URL
  - D. The protocol for communication with this site
  
3. To roughly sketch the different components of a website, such as the main pages and the layout of the pages you should use a:
  - A. Any web development text editor
  - B. Navigation Bar
  - C. Storyboard
  - D. A web site template



## Sample Questions | Answers

1. What does HTML stand for?
  - A. High Text Marking Language
  - B. Hyper Text Mark Up Language**
  - C. High Text Mark Up Layout
  - D. Hyper Text Mark Up Layout
  
2. What does www.cepis.org denote in the address http://www.cepis.org/media/pressr1.pdf ?
  - A. Address or domain name of the host computer**
  - B. Geo-location of the organization
  - C. The type of the organization using this URL
  - D. The protocol for communication with this site
  
3. To roughly sketch the different components of a website, such as the main pages and the layout of the pages you should use a:
  - A. Any web development text editor
  - B. Navigation Bar
  - C. Storyboard**
  - D. A web site template





## Exercises

Draw a storyboard for your personal website which should at least include:

- A home page
- A contact me page
- An activities/interest page

### References

- ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.
- <https://www.webdesign.org/web-design-basics/design-principles/page-1.html#ixzz4zdkyGUFj>

# Coding Bootcamp

## Lessons 8-9-10

### HTML

---

#### Objectives:

- HTML Document Structure
  - Tags – Elements
  - Headings
  - Paragraph
  - Formatting Elements
  - Line Break, Comments
- Special Characters
- Empty Elements
- Lists
- Create a simple web page using HTML
- HTML 5 structural & input elements, attributes

---

#### Syllabus Items:

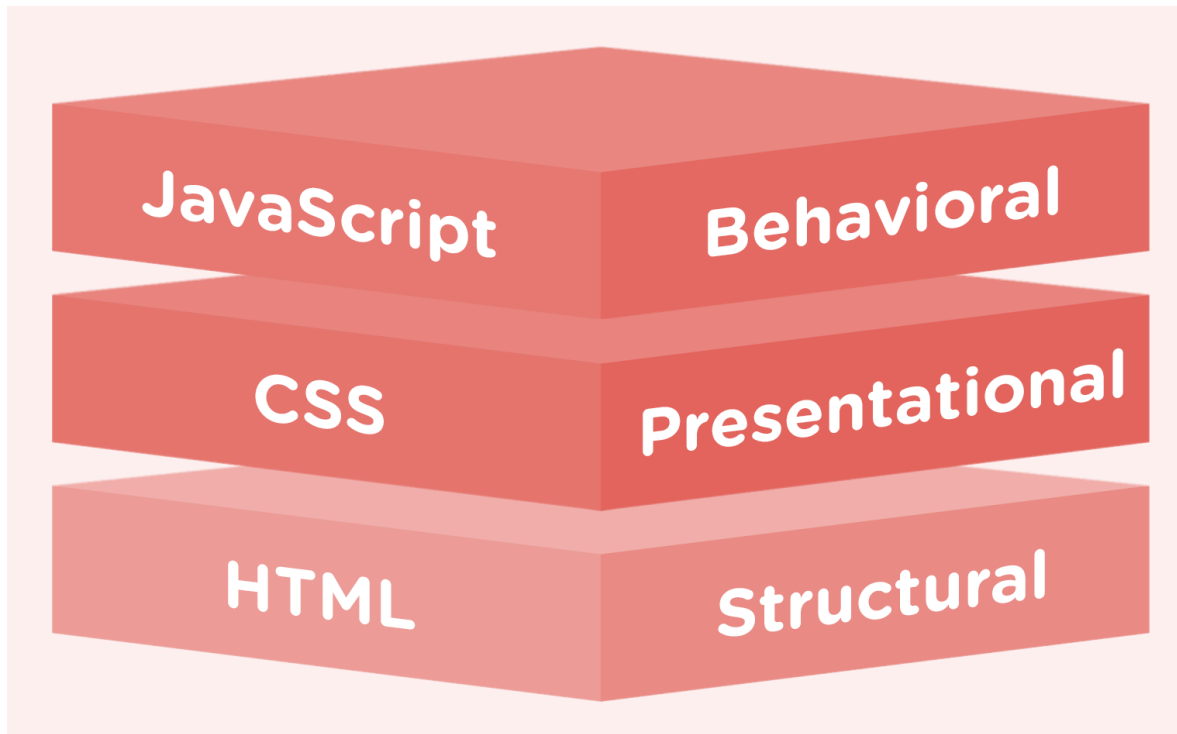
- 4.2 HTML



# Syllabus

Category	Topic	Task
<b>FSD_4 Web Design and Development Fundamentals (Front-End)</b>	<b>4.2 Key Concepts and Principles</b>	4.2.1 Understand what HTML, HTML Document Structure, Tags - Elements, Headings, Paragraph, Formatting Elements, Line Break, Comments, Special Characters, Empty Elements, Lists are
		4.2.2 Create a page using HTML Attributes, Images, Links, Tables, Block level & Inline Elements, Div & Span Elements, Forms, Form elements
		4.2.3 Understand and use HTML 5 elements, like: structural elements, input elements, attributes

## Role of HTML in Web Pages & Applications



# HTML

- Remember....HTML is short for: **HyperText Markup Language**

## HTML Versions

- November 24, 1995, HTML 2.0 was published as IETF RFC 1866
- December 18, 1997, HTML 4.0 was published as a W3C Recommendation
- 28 October 2014, HTML5 was released as a stable W3C Recommendation

## HTML 4 vs HTML 5

- Simplified and Clear Syntax
- Multimedia Elements
- Accessing User Geographical location
- Client Side storage
- Client Server Communication
- JavaScript Threading Mechanism
- Browser Compatibility
- Tag Differences
- HTML5 New Tags
- Important improvements in terms of accessibility

## HTML Files

- HTML files are simply text files with an .htm or .html file extension
- HTML uses special tags to provide instructions to the browser (for content, layout etc.)
- Major parts of a HTML page
  - Title (web page title, displayed on browser window)
  - Head
  - Body

## Layout of a HTML file

<code>&lt;!DOCTYPE html&gt;</code>	<code>&lt;html&gt;</code>
<code>&lt;html&gt;</code>	<code>&lt;head&gt;</code>
<code>&lt;head&gt;</code>	<code>&lt;title&gt; My title</code>
<code>&lt;title&gt; This is the</code>	<code>&lt;/title&gt;</code>
<code>title of the page &lt;/title&gt;</code>	<code>&lt;/head&gt;</code>
<code>&lt;/head&gt;</code>	<code>&lt;body&gt;</code>
<code>&lt;body&gt;</code>	<code>... ..</code>
<code>This is the main content</code>	<code>... ..</code>
<code>of the page</code>	<code>... ..</code>
<code>&lt;/body&gt;</code>	<code>&lt;/body&gt;</code>
<code>&lt;/html&gt;</code>	<code>&lt;/html&gt;</code>

*DOCTYPE is not an HTML tag that is why it is not closed. Also it is only used once as it informs the browser which HTML language will be used.*

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

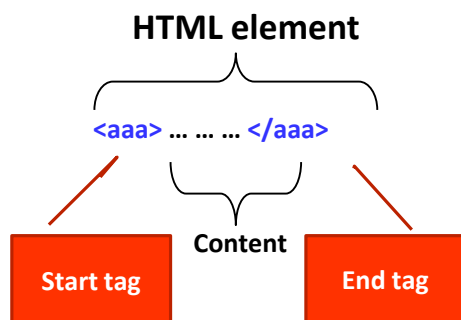
# DOCTYPE | DOCTYPES | HTML Elements

## DOCTYPE

- The `<!DOCTYPE>` declaration must be the very first thing in your HTML document, before the `<html>` tag.
- The `<!DOCTYPE>` declaration is not an HTML tag. It is an instruction to the web browser about what version of HTML the page is written in.

## DOCTYPES

- HTML 4.01 Strict  
`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- HTML 4.01 Transitional  
`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
- HTML 4.01 Frameset  
`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">`
- HTML 5  
`<!DOCTYPE html>`



Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

# Tags

- All elements must have a closing tag : `<aaa> ... .. </aaa>`
- You can have elements inside other elements (beware of tags):  
`<aaa> ...<bbb> ... </bbb>... </aaa>`  
~~`<aaa> ...<bbb> ... </aaa>... </bbb>...`~~
- Empty elements (no content) close on start tag: `<ccc/>`
- Attributes provide additional information about HTML elements.

## Attributes (intro)

- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

### Examples

`<a href="page2.html">Page 2</a>`

- Multiple attributes (name/value pairs) needs a space between them  
i.e. `<body bgcolor="gray" leftmargin="20">`

### <html>

- The first tag that any Web page must have is the `<html>` tag.
- Tells the browser that the following text will be marked up in HTML format, and not some other format.
- Includes only the `<head>` and `<body>` tags
- The closing tag `</html>` is required and is the last tag in our document.

### <head>

- The `<head>` tag is used to contain information relating to the head elements of the document, including the title and the style of the document.
- Also includes page information that will not display on the page (i.e. meta charset="UTF-8" , which denotes that the character set used for the page also supports Greek characters)
- The closing tag `</head>` is required.

### <title>

- The `<title>` tag belongs in the `<head>` section of our document.
- It is the title of our Web page, and is usually displayed by the browser at the top of the browser pane.

### <body>

- The `<body>` defines the primary portion of our Web page.
- Contains all the contents of the document, including text, images, and hyperlinks.
- Only ONE body tag per web page
- The closing tag `</body>` is required.
- Includes additional tags which may include text, images or other tags.
- Anything in the `<body>` will be displayed by the browser

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

# HTML Elements | Headings | Paragraphs | Comments

**Comments** `<!-- -->`

`<!-- This is a comment -->`

**Paragraphs** `<p>`

`<p>This is a paragraph</p>`

`<pre>Keeps formatting</pre>`

**Headings & Paragraphs**

- `<h1>`This is heading 1`</h1>`
- `<h2>`This is heading 2`</h2>`
- `<h3>`This is heading 3`</h3>`
- `<h4>`This is heading 4`</h4>`
- `<h5>`This is heading 5`</h5>`
- `<h6>`This is heading 6`</h6>`
- Heading tags are important tags for organizing information on a web page.
- Think of your site as a newspaper, so headings are the headlines on the newspaper.
- The main headline is an `<h1>` and subsequent heading is `<h2>` (and down as far as `h6`).
- You need to close the heading with ending tag `</h1>` or `</h2>`
- The `<p>` tag defines a paragraph.
- Paragraphs are blocks of text that are usually separated from adjacent blocks by vertical blank space.
- A `<p>` tag can be anywhere in the `<body>` of an html page and will create a new paragraph.
- A comment begins with `<!--` and ends with `-->`
- Comments are used within HTML code, as like in any programming language
- Comments are not shown on the screen, since the browser ignores them
- In HTML most elements have a start and an end tag along with content

However there are several **EMPTY elements** that are only start tags, like:

- **Line Break**
  - `<br/>`
- **Horizontal line**
  - `<hr/>`
- **Image**
  - `<img>` (more on image below)

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



## Example: Headings | Paragraphs | Empty Elements

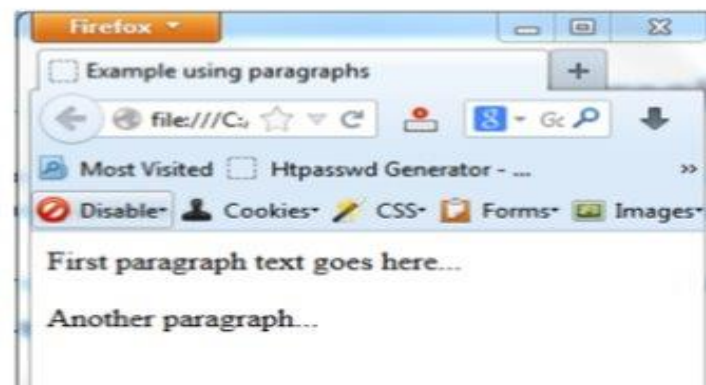
**This is heading 1**

**This is heading 2**

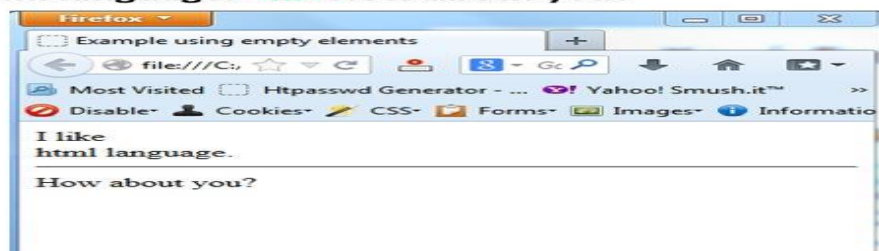
**This is heading 3**

**This is heading 4**

```
<html>
  <head>
    <title>Example using paragraphs</title>
  </head>
  <body>
    <p>First paragraph text goes here...</p>
    <p>This is another paragraph...</p>
  </body>
</html>
```



```
<html>
  <head>
    <title>Example using empty elements</title>
  </head>
  <body>
    I like<br>html language. <hr>How about you?
  </body>
</html>
```



Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



# Class Exercise

## Exercise 1

- Create a html page with the following tags
- `<html>`
- `<head>`
- `<title>`
- This is the title of the page.
- `</title>`
- `</head>`
- `<body>`
- The body text goes here.
- `</body>`
- `</html>`
- Input your own text into the bold text
- Save as **test.html**

## Exercise 2

- Create a html page that has 2 paragraphs and 2 headings with different sizes

## Exercise 3

- Comment the second heading and see what happens

## Exercise 4

- Use one of the previous simple html files you created and add:
- Headings
- Paragraphs
- Line breaks
- Horizontal lines

## Formatting Text

Tag	Description
<code>&lt;b&gt;....&lt;/b&gt;</code>	Bold text (obsolete – use CSS instead)
<code>&lt;strong&gt;...&lt;/strong&gt;</code>	Strong text (similar to bold) but defines important text
<code>&lt;i&gt;....&lt;/i&gt;</code>	Italics text
<code>&lt;em&gt; ...&lt;/em&gt;</code>	Renders as emphasized text
<code>&lt;u&gt;...&lt;/u&gt;</code>	Underlined text
<code>&lt;sup&gt; ...&lt;/sup&gt;</code>	Superscript text
<code>&lt;sub&gt; ...&lt;/sub&gt;</code>	Subscript text
<code>&lt;del&gt; ...&lt;/del&gt;</code>	Text that has been deleted from a document
<code>&lt;pre&gt; ...&lt;/pre&gt;</code>	Preformatted text in a fixed-width font (usually Courier), preserving both spaces and line breaks
<code>&lt;small&gt;...&lt;/small&gt;</code>	smaller text
<code>&lt;big&gt;...&lt;/big&gt;</code>	bigger text (not supported by HTML5)

## Special Characters

Character	HTML	NAME
&	<code>&amp;amp;</code>	ampersand
€	<code>&amp;euro;</code>	euro
©	<code>&amp;copy;</code>	copyright
÷	<code>&amp;divide;</code>	division
>	<code>&amp;gt;</code>	greater than
<	<code>&amp;lt;</code>	less than
«	<code>&amp;laquo;</code>	left-pointing double angle quotation mark
»	<code>&amp;raquo;</code>	right-pointing double angle quotation mark
	<code>&amp;nbsp;</code>	non-breaking space ( <b>KEVŐ</b> )
£	<code>&amp;pound;</code>	pound
¶	<code>&amp;para;</code>	paragraph
®	<code>&amp;reg;</code>	registered
§	<code>&amp;sect;</code>	section
™	<code>&amp;trade;</code>	trade mark

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



## Class Exercise

### Exercise 5

Let's try this:

```
<!DOCTYPE html>
<html>
<body>
  <p>Coding Bootcamp 3 </p>
  <p><small>created 2017 by AfDEMP</small></p>
  <strong>Full time class</strong>
  <hr/>
  <p>This text contains <sub>subscript</sub> text.</p>
  <p>This text contains <sup>superscript</sup> text.</p>
  <pre>
    Text in a pre element is displayed in a fixed-width
    font, and it preserves both spaces and line breaks
  </pre>
</body>
</html>
```

# Grouping

- `<article>`  
specifies independent, self-contained content.
- `<section>`  
defines sections in a document, such as chapters, headers, footers, or any other sections of the document.
- `<details>` `<summary>`  
additional details that the user can view or hide on demand. (class practice)
- `<address>`  
contact information
- `<div>`  
division or a section in an HTML document.  
The `<div>` tag is used to group block-elements to format them with CSS.

## Element Types

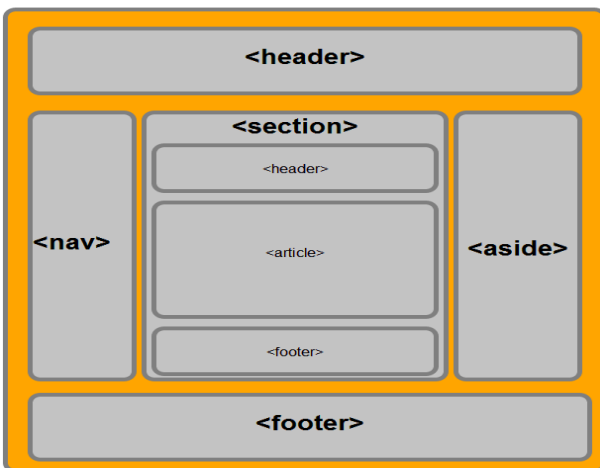
- **Block Level:** A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). Can hold text, other block or inline elements. They begin on a new line.  
Examples:  
`<p>`, `<h1>...<h6>`, `<hr>`, `<pre>`, `<ul>`, `<ol>`, `<li>`, `<table>`, `<tr>`, `<div>`
- **Inline:** An inline element does not start on a new line and only takes up as much width as necessary. Can only have text or other inline elements. No dimensions (width or height) but automatically defined by their length.  
Examples:  
`<b>`, `<i>`, `<em>`, `<br>`, `<strong>`, `<u>`, `<a>`, `<span>`, `<img>`

## Attributes (2)

- Most tags have attributes
- The style attribute will be used extensively in CSS but for now, keep in mind that you can set a style for parts of your page within your HTML code. Basic attributes include:
  - background-color
  - font-family
  - font-size
  - color
  - text-align

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

# HTML Layout



## ATTRIBUTES EXAMPLE

```
<p align="left">... paragraph text goes here ...</p>
<p align="center">...paragraph text goes here...</p>
<p align="right">... paragraph text goes here ...</p>
<p align="justify">... paragraph text goes here ...</p>
```

## STYLE EXAMPLE

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1 {color: red;}
    p {color: blue;}
  </style>
</head>
<body>
  <h1>A heading</h1>
  <p>A paragraph.</p>
</body>
</html>
```

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

# Lists

- HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements
  - `<ul>` – An **unordered** list. This will list items using plain bullets.
  - `<ol>` – An **ordered** list. This will use different schemes of numbers to list your items.
  - `<dl>` – A **description** list. This arranges your items in the same way as they are arranged in a dictionary.

## Unordered List

```
<ul>
  <li>White</li>
  <li>Black</li>
  <li>Red</li>
  <li>Yellow</li>
</ul>
```

An Unordered List:

- Item
- Item
- Item
- Item

```
<b>Unordered List</b>
<ul>
  <li>Milk</li>
  <li>Coffee</li>
  <li>Tea</li>
</ul>
```

## Ordered List

```
<ol>
  <li>White</li>
  <li>Black</li>
  <li>Red</li>
  <li>Yellow</li>
</ol>
```

An Ordered List:

1. First item
2. Second item
3. Third item
4. Fourth item

```
<b>Ordered List</b>
<ol>
  <li>Basket</li>
  <li>Football</li>
  <li>Tennis</li>
</ol>
```

## Nested Elements

- Elements in HTML can be nested within each other
- Will display as: `<p>Hello, my name is <b>John</b> and I am a <i>student</i>.</p>`  
Hello, my name is **John** and I am a *student*.

## Nested Elements - Tips

- When using nested elements we must close tags in reverse order and not in the order we opened them
- Example: WRONG: `<strong> <em>John Smith</strong> </em>`  
RIGHT: `<strong> <em>John Smith</em> </strong>`

## Nested List

```
<ol>
  <li>Santorini</li>
  <li>Mykonos</li>
  <li>Naxos</li>
  <li>Sifnos</li>
</ol>
```

Sources: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol>  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul>

ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017



## Class Exercise

### Exercise 5

Create a html page with all appropriate elements and the list in the previous slide:

Things to do tomorrow

Service car.

Go to super market.

Cheese.

Milk.

Coffee.

Sodas.

Call mom.

### Exercise 6

Create a html page with all appropriate elements and the list in the previous slide:

Add any other elements you seem fit.



## Description Lists

- `<dl></dl>` defines the entire list.
- `<dt></dt>` defines each definition term.
- `<dd></dd>` defines the definition data.

### Example:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

## Images

- To add an image to our website, we use the `<img>` tag
- The `<img>` tag does not have a closing tag
- There are two attributes (mandatory)
  - `src` (URL – specifies the source of the image)
  - `alt` (alternative text, which will be displayed when for some reason the image can not be displayed by the browser)

### Example:

```
<img src= "images/jlo.jpg" alt= "Jennifer Lopez photo" >

```

- Images are usually .gif, .png, or .jpeg (or newer formats like .webp)
- Images do not create line breaks as they are inline elements
- The `<img>` tag has two optional attributes:
- `width`
- `height`

### Example

```
<img src= "images/myphoto.jpg"
alt="me" width="90" height="80"/>
```

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

## Pictures/Images | Example

```
<figure>
  
  <figcaption>
    This is a koala
  </figcaption>
</figure>
```

### Pictures/Images | Example

Attribute	Function
alt	<ul style="list-style-type: none"> <li>• Alternative text to display when an image is being loaded</li> <li>• Especially useful for screen readers, which translate information on a computer screen into audio output</li> <li>• Should be a brief representation of the purpose of the image</li> <li>• Generally should stick to 50 characters or fewer</li> </ul>
height	<ul style="list-style-type: none"> <li>• Defines the height of the image, measured in pixels</li> <li>• Improves loading time</li> </ul>
hspace	<ul style="list-style-type: none"> <li>• Defines the horizontal space that separates the image from the text</li> </ul>
src	<ul style="list-style-type: none"> <li>• Defines the URL of the image to be loaded</li> </ul>
vspace	<ul style="list-style-type: none"> <li>• Defines the vertical space that separates the image from the text</li> </ul>
width	<ul style="list-style-type: none"> <li>• Defines the width of the image, measured in pixels</li> <li>• Improves loading time</li> </ul>

### Pictures/Images | Example

```
<html>
  <head>
    <title>Html Example 1 - The Maldives Honeymoon</title>
  </head>
  <body>
    <h1 style="text-align: center;">The Maldives Honeymoon</h1>
    <p style="text-align: center;"></p>
    <hr>
    <p style="text-align: justify;">If <em>a honeymoon is meant to be <strong>a celebration of love</strong></em> in an intimate, secluded, and most importantly, beautiful setting, <u>then the <b>Maldives</b> is the world's best backdrop for all these things</u>. There are endless ways to let the magic of the islands dazzle you on your holiday as a couple. A dinner under the stars with the occasional flicker of candle light to bring you back into the real world, a daring getaway to a nearby uninhabited island all by yourselves for the whole day, or just lazing around in your private bungalow watching the endless turquoise waters while you are treated to an spa treatment in the room.
  </p>
</body>
</html>
```

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



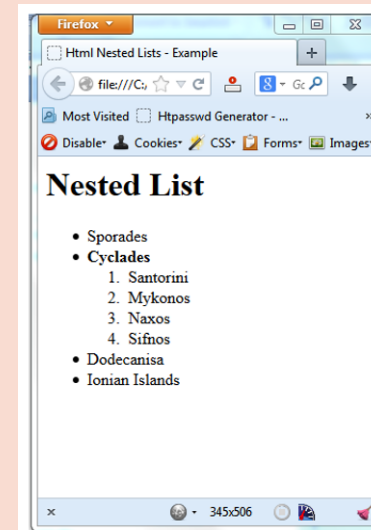
# Class Exercise

## Class Exercise 1

- Add a couple of images to the html page you create in Exercise 6.
- If you do not have any images available on your device, please locate some images from the Internet.

## Class Exercise 2

- Pick your favorite island and locate one picture from the Internet.
- Use that picture and some information about this island to create a simple web page, similar to the Maldives example provided before.



# Hyperlinks (or Links)

- A hyperlink can exist on text or an image and will transfer us to a new site
- We can use hyperlinks to:
  - Create an internal link (to our site)
  - Create an external link (to another site)
  - Create an internal bookmark to another location on our page

**Example:** `<a href="http://www.google.com">click to Visit Google</a>`  
`<a href="people/jsmith.html">view John Smith profile</a>`  
`<a href="cars.html"></a>`  
`<a href="#top">go to the top of the page</a>`

## Attributes

- href (URL, the address to which we want to transfer upon click)
- target (\_blank or \_self) – where to open the link

Tips:

- Always close the <a> attribute with an </a>
- Pay attention to the image paths especially if stored locally
- All external URL references must start with http://

## Links - href attribute

```
<a href="page2.html">Page 2</a>
<a href="http://www.google.com">Google</a>
```

href: The url of the page we want to go to

## Links - href attribute

- Attribute target
  - \_blank (opens in a new window)
  - \_self (opens in the same window)
  - \_parent (opens in the parent frame)
  - \_top (opens in the whole window)

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

# rel Attribute

## Definition:

*Specifies the relationship of the target object to the link object. The value is a space-separated list of link types:*

- **alternate**
  - Links to an alternate version of the document (i.e. print page, translated or mirror)
- **author**
  - Links to the author of the document
- **bookmark**
  - Permanent URL used for bookmarking
- **help**
  - Links to a help document
- **nofollow**
  - Links to an unendorsed document, like a paid link. ("nofollow" is used by Google, to specify that the Google search spider should not follow that link)
- **noreferrer**
  - Specifies that the browser should not send a HTTP referrer header if the user follows the hyperlink
- **prefetch**
  - Specifies that the target document should be cached
- **prev**
  - The previous document in a selection
- **search**
  - Links to a search tool for the document
- **license**
  - Links to copyright information for the document
- **next**
  - Links to the next nexturce of the sequence the current page is in.
- **tag**
  - A tag (keyword) for the current document

*Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.*

## Anchors

- Define the page “anchor” (the line we want to go to)

```
<p id="top">Help</p>
```

- Create a link with href value, the anchor id:

```
<a href="#top">Show Help</a>
```

### Example:

```
<!-- anchor linking to external file -->
<a href="https://www.mozilla.com/">External Link</a>

<!-- links to element on this page with id="attr-href" -->
<a href="#attr-href">Description of Same-Page Links</a>

<!-- creating a clickable image -->
<a href="https://developer.mozilla.org/en-US/" target="_blank">
  
</a>

<!-- creating an email link -->
<a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>

<!-- creating a telephone link -->
<a href="tel:+491570156">+49 157 0156</a>
```

### e-mail – file links

```
<a href="mailto:slee@yourdomain.com">contact Sean Lee</a>
```

```
<a href="got.jpg" download>Link to image</a>
```

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



## Class Exercise

- Using your ISLAND page, add another island and use anchors and links like in the previous example to create a website that has information about TWO of your favorite islands.

# HTML Tables

The <table> tag inserts a table into our web page

Apple	Microsoft	Google
2	4	3
5	2	7
3	6	2
10	12	12

Diagram labels for the table structure:

- head: Points to the first row (th headers).
- body: Points to the three rows of data (td cells).
- foot: Points to the last row (tfoot).

Apple	Microsoft	Google
2	4	3
5	2	7
3	6	2
10	12	12

Diagram labels for the table structure:

- Table row: Points to the first row of data (2, 4, 3).
- Table cell: Points to the cell containing the value 7.

```

<table border="1">
  <thead>
    ...
  </thead>
  <tbody>
    ...
    ...
  </tbody>
  <tfoot>
    ...
  </tfoot>
</table>

```

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



# HTML Tables

Table Head

```
<thead>
  <tr>
    <th>Apple</th>
    <th>Microsoft</th>
    <th>Google</th>
  </tr>
</thead>
```

Table Rows

```
<tr>
  <td>2</td>
  <td>4</td>
  <td>3</td>
</tr>
<tr>
  <td>5</td>
  <td>2</td>
  <td>7</td>
</tr>
```

<tr> : Table row  
<td> : Table data

colspan

The colspan attribute defines the number of columns a cell should span.

```
<td colspan="2">
```

rowspan

The rowspan attribute defines the number of rows a cell should span.

```
<td rowspan="2">
```

<table> attributes

**cellspacing**: Set the space between the cells in pixels (table tag)

**cellpadding**: Set the space between the cell wall and the cell content (table tag)

**bgcolor**: background color (all tags)

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

# Tables | Example

```

<html>
<head>
<title>Example of inserting a table</title>
</head>
<body>
<table border="1">
  <caption>2x2 Table</caption>
  <tr>
    <td>row 1, cell 1</td>
    <td>row 1, cell 2</td>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
</table>
</body>
</html>

```

**1st Line**

**2nd Line**

**Caption**

**Cells**

2x2 Table	
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

```

<html>
<head>
<title>Table with header cells and specific width</title>
</head>
<body>
<table border="1" width="400px">
  <caption>Salary</caption>
  <tr>
    <th>June</th>
    <th>July</th>
  </tr>
  <tr>
    <td>1250<euro></td>
    <td>1120<euro></td>
  </tr>
</table>
</body>
</html>

```

Salary	
June	July
1250€	1120€

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

## Tables | Example (2)

```
...
<table border="1" width="300" >
  <tr>
    <th colspan="3">Summer Profit</th>
  </tr>
  <tr>
    <th>June</th>
    <th>July</th>
    <th>August</th>
  </tr>
  <tr>
    <td>2850&euro;</td>
    <td>3220&euro;</td>
    <td>4300.55&euro;</td>
  </tr>
</table>
...
```

Summer Profit		
June	July	August
2850€	3220€	4300.55€

```
...
<table border="1" width="300" >
  <caption>Summer Profit</caption>
  <tr>
    <th>June</th>
    <th>July</th>
    <th>August</th>
  </tr>
  <tr>
    <td>2850&euro;</td>
    <td>3220&euro;</td>
    <td rowspan="2">-</td>
  </tr>
  <tr>
    <td>2350&euro;</td>
    <td>3220&euro;</td>
  </tr>
</table>
...
```

Summer Profit		
June	July	August
2850€	4300€	-
2350€	2500€	

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



## Class Exercise

- Create the following table

Title goes here			A	B
C	D	E	F	G
	H	I		J
	K	L	M	
N	O			

## <div> tag

- The <div> tag defines a division or a section in an HTML document.
- The <div> tag is used to group block-elements to format them with CSS.
- Can include all tags that go into <body> like: <p>, <h1>,<ul>,<ol>,<table>,<img>,<a> etc.
- Must be close with a </div> tag
- A major element for the layout of a page

## <span> tag

- The <span> tag is used to group inline-elements in a document.
- The <span> tag provides no visual change by itself.
- The <span> tag provides a way to add a hook to a part of a text or a part of a document
- Must be close with a </span> tag
- <span> brings no visual changes to our web page and can include only text and inline elements
- <span> is an inline element

### <div> Example

```
<body>
  <div id="container">
    <div id="header">
      <h1>Site name</h1>
    </div>
    <div id="navigation">
      <a href="index.html">Home</a>
      <a href="about.html">About</a>
      <a href="contact.html">Contact</a>
    </div>
    <div id="content">
      <h2>Page heading</h2>
      <p>Lorem ipsum dolor sit amet consectetur elit sed diam nonummy nibh euismod tinunt ut laoreet dolore magna aliquam erat volut. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.
    </p>
    </div>
    <div id="footer">
      <span>Copyright &copy; Site name, 20XX</span>
    </div>
  </div>
</body>
...
```

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.

# HTML Forms | HTML 5 Input Types

- The HTML <form> element defines a form that is used to collect user input:
- An HTML form contains form elements.
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

```
<form action="server side something" method="get/post">
```

```
...
```

```
...
```

```
</form>
```

The action attribute specifies where to send the form-data when a form is submitted.

The method attribute specifies how to send form-data

- Text defines a one-line text input field

```
<input type="text" name="firstname" maxlength="50" />
```

```
<input type="search" name="firstname" maxlength="50" />
```

**placeholder attribute:** specifies a short hint that describes the expected value of an input field

```
<input type="password" name="password" maxlength="25" />
```

- More text inputs

```
<input type="email" name="mymail" />
```

```
<input type="tel" name="mytel" />
```

```
<input type="url" name="mysite" />
```

Source: ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.



**Any Questions?**



## References | Further Reading

- David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, Sailu Reddy: HTTP: The Definitive Guide
- ECDL Web-Editing SV2.0, Learning materials. Dublin, Ireland: internal publication, ECDL Foundation, 2017.
- Gary B. Shelly, H. Albert Napier, Ollie N. Rivers. Web Design: Introductory Concepts and Techniques
- [http://ism.dmst.aueb.gr/bootcamp/html\\_bootcamp.pdf](http://ism.dmst.aueb.gr/bootcamp/html_bootcamp.pdf)
- <http://www.steves-internet-guide.com/dns-lookups/>
- <https://blog.catchpoint.com/2010/09/17/anatomyhttp/>
- <https://books.google.gr/books?id=m00EOqYkbcwC>
- [https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input%3Cinput%3E\\_types](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input%3Cinput%3E_types)
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol>
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul>
- [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [https://en.wikipedia.org/wiki/Internet\\_protocol\\_suite](https://en.wikipedia.org/wiki/Internet_protocol_suite)
- [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser)
- [https://en.wikipedia.org/wiki/Web\\_development](https://en.wikipedia.org/wiki/Web_development)
- [https://en.wikipedia.org/wiki/Web\\_development\\_tools](https://en.wikipedia.org/wiki/Web_development_tools)
- [https://public.wsu.edu/~ericsson/story\\_bd.html](https://public.wsu.edu/~ericsson/story_bd.html)
- <https://searchnetworking.techtarget.com/definition/URL>
- <https://vichargrave.github.io/programming/tcp-ip-network-programming-design-patterns-in-cpp/>
- <https://www.dreamtemplate.com/blog/web-design-tips/5-steps-for-the-process-of-design-and-development-of-a-website/>
- <https://www.geeksforgeeks.org/software-engineering-architectural-design/>
- <https://www.oreilly.com/library/view/http-the-definitive/1565925092/ch01s06.html>
- <https://www.sciencedirect.com/topics/computer-science/communication-protocols>
- [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)
- [https://www.w3schools.com/css/css\\_intro.aspx](https://www.w3schools.com/css/css_intro.aspx)
- [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- <https://www.webdesign.org/web-design-basics/design-principles/page-1.html#ixzz4zdKyGUFj>
- <https://www.webfx.com/blog/web-design/the-15-most-popular-text-editors-for-developers>
- <https://www.webhostdesignpost.com/website/websitestoryboarding.html>
- <https://www.webopedia.com/TERM/H/HTML.html>



# Coding Bootcamp

## Lessons 8-9-10

### Web Design and Development Fundamentals (Front End) - HTML

---

#### Objectives:

- Understand and use web development editors, tools and principles
- Understand and use basic design elements
- Create a page using HTML
- Understand and use HTML 5 elements

---

#### Syllabus Items:

- 4.1 Key Concepts and Principles
- 4.2 HTML



# Syllabus

Category	Topic	Task	
<b>FSD_4 Web Design and Development Fundamentals (Front-End)</b>	<b>4.1 Key Concepts and Principles</b>	4.1.1	Outline web development, browsers, HTTP (Hyper Text Transfer Protocol), HTML (Hyper Text Mark Up Language), CSS (Cascading Style Sheets), URL (Uniform Resource Locator)
		4.1.2	Understand and use web development editors, tools and principles
		4.1.3	Understand and use basic design elements: design concept, page(s) design, Home page layout, navigation bar
	<b>4.2 HTML</b>	4.2.1	Understand what HTML, HTML Document Structure, Tags - Elements, Headings, Paragraph, Formatting Elements, Line Break, Comments, Special Characters, Empty Elements, Lists are
		4.2.2	Create a page using HTML Attributes, Images, Links, Tables, Block level & Inline Elements, Div & Span Elements, Forms, Form elements
		4.2.3	Understand and use HTML 5 elements, like: structural elements, input elements, attributes



## Learning Objectives

- ✓ Understand and use web development editors, tools and principles
- ✓ Understand and use basic design elements
- ✓ Create a page using HTML
- ✓ Understand and use HTML 5 elements



## Input Types | Overview

### Definition:

The **HTML <input> element** is used to create **interactive controls** for **web-based forms** in order to accept data from the user; a wide variety of types of input data and control widgets are available, depending on the device and user agent.

- The **<input>** element is one of the most powerful and complex in all of HTML due to the sheer number of combinations of input types and attributes.
- The **<input>** element is so powerful because of its **attributes**; the type attribute being the most important.
- Every **<input>** element, regardless of type, is based on the **HTMLInputElement** interface, and technically share, the exact same set of attributes. However, most attributes have an effect on only a specific subset of input types. The way some attributes impact an input depends on the input type, impacting different input types in different ways.
- The **<input>** element, covering the original values of the type attribute has been available since the early days of HTML.
- In the next slides, the functionality of newer form controls will be shown in detail, including some new input types, which were added in HTML5 to allow collection of specific types of data.

Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)

## Input Types

Type	Description
<b>button</b>	A push button with no default behavior displaying the value of the value attribute, empty by default.
<b>checkbox</b>	A check box allowing single values to be selected/deselected.
<b>color</b>	A control for specifying a color; opening a color picker when active in supporting browsers.
<b>date</b>	A control for entering a date (year, month, and day, with no time). Opens a date picker or numeric wheels for year, month, day when active in supporting browsers.
<b>datetime-local</b>	A control for entering a date and time, with no time zone. Opens a date picker or numeric wheels for date- and time-components when active in supporting browsers.
<b>email</b>	A field for editing an email address. Looks like a text input, but has validation parameters and relevant keyboard in supporting browsers and devices with dynamic keyboards.
<b>file</b>	A control that lets the user select a file. Use the accept attribute to define the types of files that the control can select.
<b>hidden</b>	A control that is not displayed but whose value is submitted to the server. There is an example in the next column, but it's hidden!
<b>image</b>	A graphical submit button. Displays an image defined by the src attribute. The alt attribute displays if the image src is missing.
<b>month</b>	A control for entering a month and year, with no time zone.
<b>number</b>	A control for entering a number. Displays a spinner and adds default validation when supported. Displays a numeric keypad in some devices with dynamic keypads.
<b>password</b>	A single-line text field whose value is obscured. Will alert user if site is not secure.

Source: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>

## Input Types

Type	Description
<b>radio</b>	A radio button, allowing a single value to be selected out of multiple choices with the same name value.
<b>range</b>	A control for entering a number whose exact value is not important. Displays as a range widget defaulting to the middle value. Used in conjunction <code>htmlattrdefmin</code> and <code>htmlattrdefmax</code> to define the range of acceptable values.
<b>reset</b>	A button that resets the contents of the form to default values. Not recommended.
<b>search</b>	A single-line text field for entering search strings. Line-breaks are automatically removed from the input value. May include a delete icon in supporting browsers that can be used to clear the field. Displays a search icon instead of enter key on some devices with dynamic keypads.
<b>submit</b>	A button that submits the form.
<b>tel</b>	A control for entering a telephone number. Displays a telephone keypad in some devices with dynamic keypads.
<b>text</b>	The default value. A single-line text field. Line-breaks are automatically removed from the input value.
<b>time</b>	A control for entering a time value with no time zone.
<b>url</b>	A field for entering a URL. Looks like a text input, but has validation parameters and relevant keyboard in supporting browsers and devices with dynamic keyboards.
<b>week</b>	A control for entering a date consisting of a week-year number and a week number with no time zone.

Source: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>

# Attributes

Attributes	Type(s)	Description
<b>accept</b>	file	Hint for expected file type in file upload controls
<b>alt</b>	image	alt attribute for the image type. Required for accessibility
<b>autocomplete</b>	all	Hint for form autofill feature
<b>autofocus</b>	all	Automatically focus the form control when the page is loaded
<b>capture</b>	file	Media capture input method in file upload controls
<b>checked</b>	radio, checkbox	Whether the command or control is checked
<b>dirname</b>	text, search	Name of form field to use for sending the element's directionality in form submission
<b>disabled</b>	all	Whether the form control is disabled
<b>form</b>	all	Associates the control with a form element
<b>formaction</b>	image, submit	URL to use for form submission
<b>formenctype</b>	image, submit	Form data set encoding type to use for form submission
<b>formmethod</b>	image, submit	HTTP method to use for form submission
<b>formnovalidate</b>	image, submit	Bypass form control validation for form submission

Source: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>

## Attributes

Attributes	Type(s)	Description
<b>formtarget</b>	image, submit	Browsing context for form submission
<b>height</b>	image	Same as height attribute for <img>; vertical dimension
<b>list</b>	almost all	Value of the id attribute of the <datalist> of autocomplete options
<b>max</b>	numeric types	Maximum value
<b>maxlength</b>	password, search, tel, text, url	Maximum length (number of characters) of value
<b>min</b>	numeric types	Minimum value
<b>minlength</b>	password, search, tel, text, url	Minimum length (number of characters) of value
<b>multiple</b>	email, file	Boolean. Whether to allow multiple values
<b>name</b>	all	Name of the input form control. Submitted with the form as part of a name/value pair.
<b>pattern</b>	password, text, tel	Pattern the value must match to be valid

Source: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>



# Attributes

Attributes	Type(s)	Description
placeholder	password, search, tel, text, url	content to be appear in the form control when the form control is empty
readonly	almost all	Boolean. The value is not editable
required	almost all	Boolean. A value is required or must be check for the form to be submittable
size	email, password, tel, text	Size of the control
src	image	Same as src attribute for <code>&lt;img&gt;</code> ; address of image resource
step	numeric types	Incremental values that are valid.
type	all	Type of input form control
value	all	Current value of the form control. Submitted with the form as part of a name/value pair.
width	image	Same as width attribute for <code>&lt;img&gt;</code>
placeholder	password, search, tel, text, url	content to be appear in the form control when the form control is empty

Source: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>



## Input Restrictions

Attribute	Description
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field

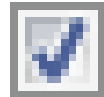
Source: [https://www.w3schools.com/html/html\\_form\\_attributes.asp](https://www.w3schools.com/html/html_form_attributes.asp)

## Input Types

**Checkbox:** Checkboxes let a user select **ZERO** or **MORE** options of a limited number of choices.

```
<input type="checkbox" name="news" value="technology" checked/>
```

Attribute: checked (boolean)



**Radio:** Radio buttons let a user select **ONLY ONE** of a limited number of choices:

```
<input type="radio" name="color" value="blue"/>
```

```
<input type="radio" name="color" value="red"/>
```

```
<input type="radio" name="color" value="green"/>
```

Attribute: checked (boolean)



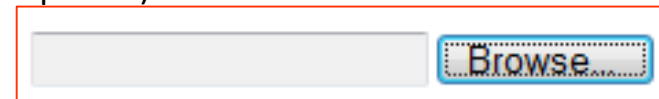
**Hidden:** Define a hidden field (not visible to a user).

A hidden field often stores a default value, or can have its value changed by a JavaScript:

```
<input type="hidden" name="productid" value="193"/>
```

**File:** Defines a file-select field and a "Browse..." button (for file uploads)

```
<input type="file" name="uploadedvideo" multiple/>
```

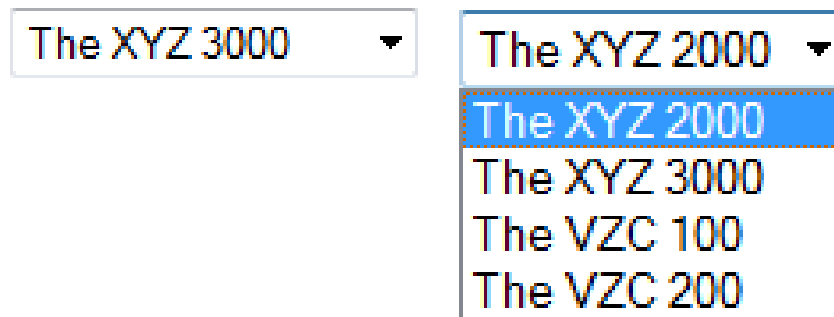


Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)

## Select

The **select** tag creates a drop-down list

```
<select name="stuff">  
  <option value="XYZ2000">The XYZ 2000</option>  
  <option value="XYZ3000">The XYZ 3000</option>  
  <option value="VZC 100">The VZC 100</option>  
  <option value="VZC 200">The VZC 200</option>  
</select>
```



The image shows two examples of HTML select dropdown menus. The left example shows a dropdown menu with 'The XYZ 3000' selected. The right example shows a dropdown menu with 'The XYZ 2000' selected, and the dropdown is open, displaying a list of all four options: 'The XYZ 2000', 'The XYZ 3000', 'The VZC 100', and 'The VZC 200'.

### More <select> attributes

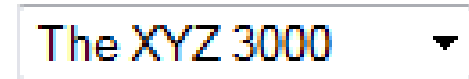
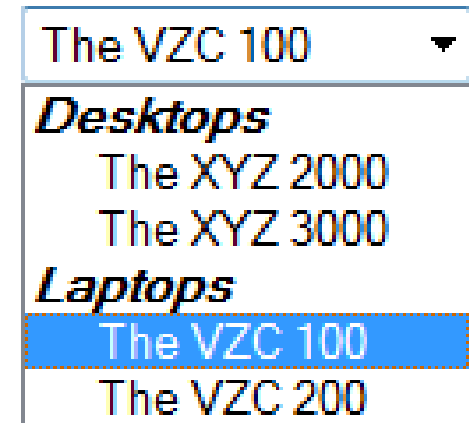
- multiple (boolean – allows multiple selections)
- size="4" (number of visible options in dropdown)

Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)

## Optgroup

Group related options with <optgroup> tags:

```
<select name="computers">
  <optgroup label="Desktops">
    <option value="xyz2000">The XYZ 2000</option>
    <option value="xyz3000">The XYZ 3000</option>
  </optgroup>
  <optgroup label="Laptops">
    <option value="vzc100">The VZC 100</option>
    <option value="vzc200">The VZC 200</option>
  </optgroup>
</select>
```

Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)

## Datalist & Textarea

### Datalist

- The **<datalist>** tag is used to provide an "autocomplete" feature on **<input>** elements. Users will see a drop-down list of pre-defined options as they input data.
- Use the **<input>** element's list attribute to bind it together with a **<datalist>** element.

```
<input list="cars">
  <datalist id="cars">
    <option value="Ferari">
    <option value="Fiat">
    <option value="Subaru">
    <option value="Ford">
    <option value="BMW">
  </datalist>
</input>
```

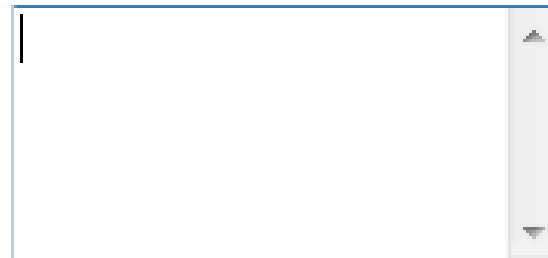
### Textarea

- The **<textarea>** tag defines a multi-line text input control.
- A text area can hold an unlimited number of characters
- The size of a text area can be specified by the cols and rows attributes, or even better through CSS height and width properties.


### More <textarea> attributes

- readonly
- disabled

```
<textarea rows="4" cols="50"></textarea>
```



Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)



## input type="submit"

- **<input>** elements of type "**submit**" are rendered as **buttons**.
- When the **click event** occurs (typically because the user clicked the button), the user agent attempts to **submit the form to the server**.

```
<input type="submit" name="submit" value="SubmitForm"/>
```

**<input type="reset">** defines a **reset button** that will reset all form values to their default values.

```
<input type="reset" name="reset" value="ResetForm" />
```

Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)

## Examples

- **Textfield:** `<input type="text" name="surname" id="surname" placeholder="give your surname">`

- **Password field:** `<input type="password" name="mypassword" placeholder="password">`

- **Checkbox:**

Over 18? `<input type="checkbox" name="age">`

Over 18? ☒

- **Radio buttons:**

Male `<input type="radio" name="gender" value="male">`

Female `<input type="radio" name="gender" value="female">`

Male ☒ Female ☐

- **Image button:** `<input type="image" name="image1" src="images/register.gif">`



- **Button:** `<input type="button" id="mybutton" value="click me now!">`

- **Submit button:** `<input type="submit" name="mysubmit" value="Next">`

- **Reset Button:** `<input type="reset" name="myreset" value="Clear fields">`

- **Hidden field:** `<input type="hidden" name="totalcost" value="1000">`

Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)





## Class Exercise 1

- Write the appropriate HTML code that will create the following form elements:

The screenshot shows a web form with three distinct input elements. At the top is a text input field with the placeholder text "give your surname". Below this is a dropdown menu with the label "Please select your OS:" and the current selection "Mac". At the bottom is a list box with the label "Please select your favorite fruits:" containing a list of fruits: apple, banana, orange, pear, blackberry, and watermelon. The "orange" and "watermelon" options are highlighted with a blue background.

## Fieldset & Other Input Types

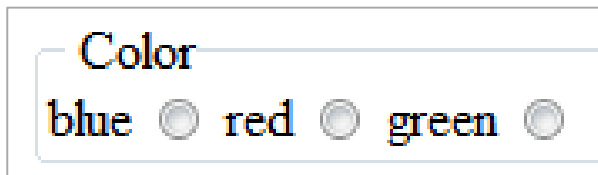
### fieldset

The **<fieldset>** tag is used to group related elements in a form, it draws a box around the related elements.

```
<fieldset>
<legend>Color</legend>
```

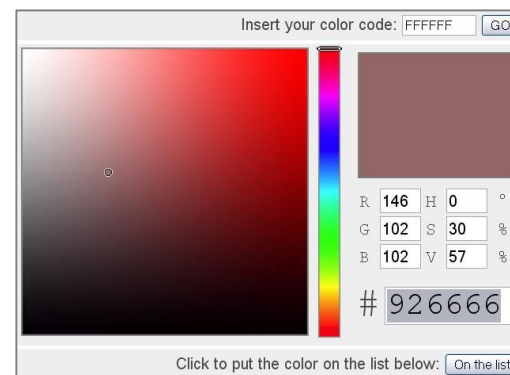
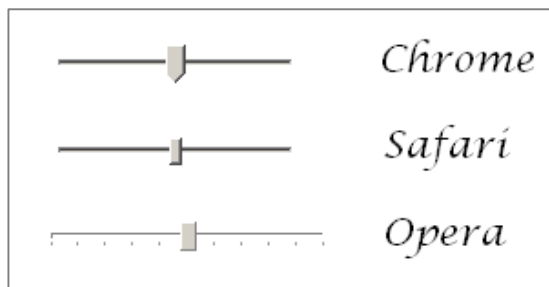
```
...
...
```

```
</fieldset>
```



### Other input types

- Dates `<input type="date" name="date" value="Ημερομηνια"/>`
- Numbers `<input type="number" id="number" value="42" />`
- Slider `<input type="range" name="number" id="price"/>`
- Color picker `<input type="color" name="color" id="color"/>`



Sources: [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)  
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>  
[https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)



**It is time for a Knowledge Check!**



## Sample Questions

1. Which of the following tags defines an unordered list in HTML?
  - A. `<ul>`
  - B. `<ol>`
  - C. `<li>`
  - D. `<h1>`
  
2. `<h2>` is a HTML tag that defines an:
  - A. Line break
  - B. Horizontal line
  - C. Page title
  - D. Heading level 2



## Sample Questions | Answers

1. Which of the following tags defines an unordered list in HTML?
  - A. `<ul>`
  - B. `<ol>`
  - C. `<li>`
  - D. `<h1>`
  
2. `<h2>` is a HTML tag that defines an:
  - A. Line break
  - B. Horizontal line
  - C. Page title
  - D. **Heading level 2**



## Sample Questions

### 3. Which code will yield this form?

**A.**

```
<form action="myscript.php" method="POST" name="myform">
  <fieldset>
    <legend>Personal Data</legend>
    Name: <input type="text" name="namefield"><br>
    Email: <input type="email" name="emailfield">
  </fieldset>
  <fieldset>
    <legend>Preferences</legend>
    Please select your favorite sports:
    <select name="select" size="6" multiple="multiple">
      <option value="football">football</option>
      <option value="basket">basket</option>
      <option value="tennis">tennis</option>
      <option value="polo">polo</option>
      <option value="volley">volley</option>
      <option value="table tennis">table tennis</option>
    </select>
  </fieldset>
  <input type="submit" value="next" name="sbutton1">
  <input type="reset" value="clear" name="rbutton1">
</form>
```

**B.**

```
<form action="myscript.php" method="POST" name="myform">
  <fieldset>
    <legend>Preferences</legend>
    Please select your favorite sports:
    <select name="select" size="6" multiple="multiple">
      <option value="football">football</option>
      <option value="basket">basket</option>
      <option value="tennis">tennis</option>
      <option value="polo">polo</option>
      <option value="volley">volley</option>
      <option value="table tennis">table tennis</option>
    </select>
  </fieldset>
  <input type="submit" value="next" name="sbutton1">
  <input type="reset" value="clear" name="rbutton1">
</form>
```



## Exercise 1

- Create a form for requesting information about a car and include:
  - Personal data field set, including Name and Email

A screenshot of a web form titled "Personal Data". It contains two input fields: "Name:" and "Email:", each followed by a text box.

- Select car List as follows:

A screenshot of a web form showing a car selection list. The text "Please select Car:" is on the left. The list on the right is titled "Swedish Cars" and includes "Volvo" and "Saab". Below that, it is titled "German Cars" and includes "Mercedes" and "Audi". The list has up and down arrow controls.

- Submit and Reset buttons



## More Exercises

### Exercise 2

- Pick a topic of your interest (i.e. travel, sports, celebrities etc.)
- Create 3 pages.
- Page1 must have links to both pages 2 and 3.
- Page 3 must have an image that it's a link to page 2

### Exercise 3

Create HTML code to display the following list:

- Service car.
- Go to super market.
  - Cheese.
  - Milk.
  - Coffee.
  - Sodas.
- Call mom.

### Exercise 4

- Create the table on the right

Table Caption

Apple	Microsoft	Google
2	4	3
5		
3	6	2
10	12	12





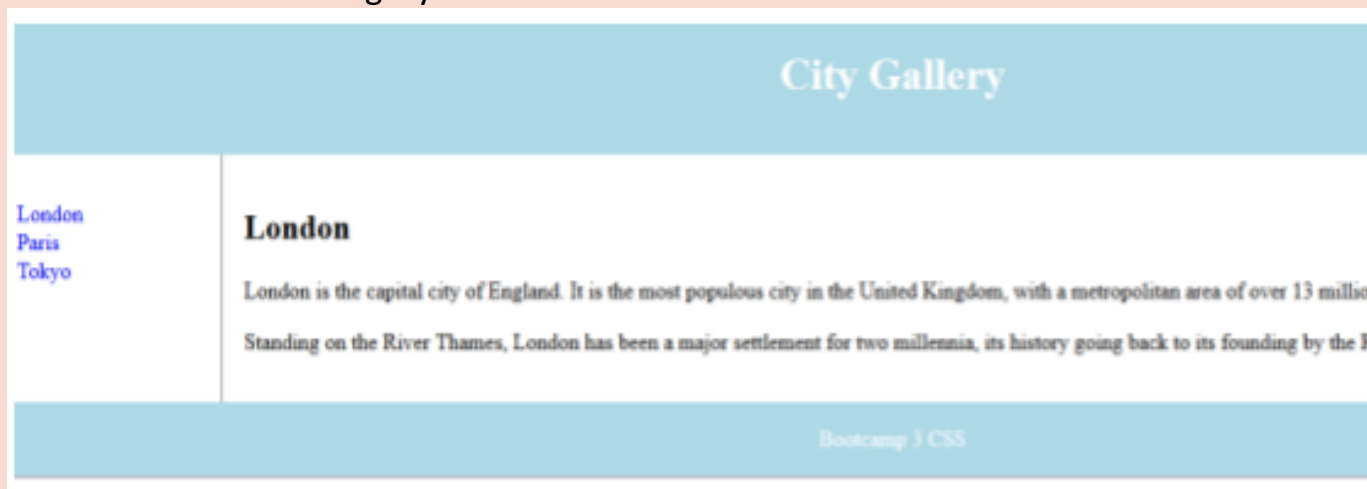
## More Exercises

### Exercise 5

- Create a paragraph with some text. Make the text size 14px and Arial font. Make the background color of the paragraph light-blue.
- Use the padding property to set the top and bottom paddings for the paragraph to "25px", and left and right paddings to "50px".

### Exercise 6

- Create the following layout



### Exercise 7

- Create a heading text. Make its background color change when the user puts the mouse on it.
- Below create a table that has different background colors for odd and even rows



# Exercise Solutions & MORE

## Exercise solutions

Sample solutions at the Student Exercise Workbook (Solutions)

## Challenge!

- Visit <https://www.w3schools.com/>
- Try several of the HTML examples
- Try to get code snippets upon which you will base your own personal website (or enhance the simple one you have already created)

## Homework

- There are extra exercises in your student workbook, which you can use to practice!
- Use the images from the **CB\_AdditionalHTML\_Exercises\_HW\_Images.zip** file provided by your instructor



## References | Further Reading

- [https://www.w3schools.com/html/html\\_form\\_input\\_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)
- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Input>
- [https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5\\_input\\_types](https://developer.mozilla.org/en-US/docs/Learn/Forms/HTML5_input_types)
- <http://www.w3schools.com>
- <http://www.htmldog.com/guides/html/beginner/forms>
- <http://www.tizag.com/htmlT/forms.php>
- <http://www.oneil.com.au/pc/html.html>
- <http://www.w3org/TR/html5/>
- [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- [https://en.wikipedia.org/wiki/Web\\_design](https://en.wikipedia.org/wiki/Web_design)
- <https://learn.onemonth.com/understanding-http-basics/>

# Coding Bootcamp

## Lessons 8-9-10

### Web Design and Development Fundamentals (Front End) Cascading Style Sheets (CSS)

---

#### Objectives:

- Define Cascading Style Sheets (CSS)
- CSS Advantages
- CSS Document Structure & Rules
- CSS Comments, Basic Selectors & Properties
- CSS and their use in HTML (Inline Style, Embedded & Linked Stylesheets)
- Create a Style Sheet

#### Syllabus Items:

- 4.3 Cascading Style Sheets (CSS)



# Syllabus

Category	Topic	Task	
<b>FSD_4 Web Design and Development Fundamentals (Front-End)</b>	<b>4.3 Cascading Style Sheets (CSS)</b>	4.3.1	Outline Cascading Style Sheets (CSS), their advantages, CSS Document Structure & Rules, Comments, Basic Selectors & Properties and their use in HTML (Inline Style, Embedded & Linked Stylesheets)
		4.3.2	Create a Style Sheet using: CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), block elements and objects, Lists and Tables, CSS Id and Class, Box Model (Introduction, Border properties, Padding Properties, Margin properties)



## Learning Objectives

- ✓ Outline Cascading Style Sheets (CSS),
- ✓ List CSS advantages, CSS Document Structure & Rules, Comments, Basic Selectors & Properties and their use in HTML (Inline Style, Embedded & Linked Stylesheets)
- ✓ Create a Style Sheet using: CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), block elements and objects, Lists and Tables, CSS Id and Class, Box Model (Introduction, Border properties, Padding Properties, Margin properties)



## What is CSS?

As you may **recall**:

- HTML was created to **describe the content** of a web page in terms of layout (i.e. headings, paragraphs etc.); thus in early editions, limited formatting was available
- **CSS** stands for **Cascading Style Sheets**
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External stylesheets are stored in **CSS files**
- When tags like **<font>**, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
- To solve this problem, the World Wide Web Consortium (W3C) created CSS.
- CSS removed the style formatting from the HTML page

Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>



# CSS Advantages & Rules

## CSS Advantages

- Provides a language that can flexibly and clearly define the manner that various elements on our website will be displayed
- Separates the content from the style
- Flexibility and ease for designing and maintaining a website (i.e. formatting is controlled by CSS, so changing formatting is simply changing the CSS)
- Less code
- Friendlier to search engines

## CSS Rules

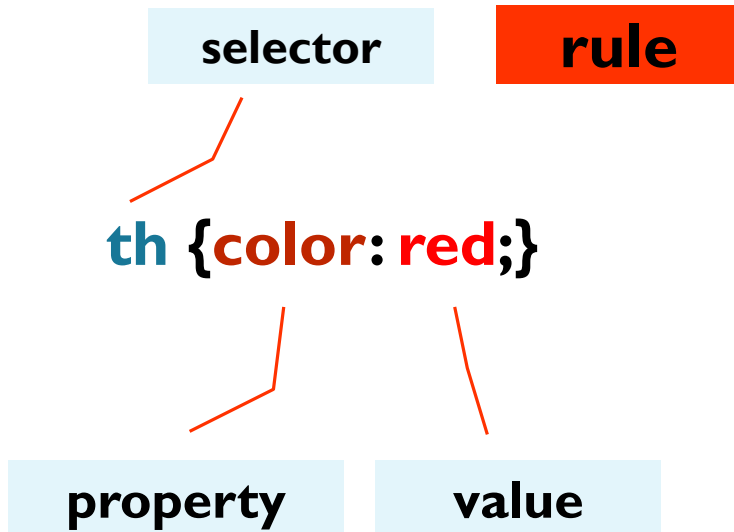
- Rules are basic elements of CSS
- Every rule has two parts:
  - **Selector**, which states which element(s) of the page will be formatted
  - **Declaration**, which defines how the selector will be displayed/formatted
- Declarations have one or more properties with respective values

Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>



# CSS Rule Structure & Syntax

## Structure



## Syntax

```
selector {
    property1: value1;
    property2: value2;
    ...
}
```

Can be in a single line too:

```
selector { property1: value1; property2: value2; ... }
```

Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>



## Examples of CSS Rules

- `body { font-family: Tahoma; font-size: 14px; color: #121212; }`
- `p { text-align: justify; color: #000000; }`
- `h2 { color: red; }`
- `div { background-color: #dddddd; color: red; }`
- `span { color: orange; font-weight: bold; }`
- `a { color: yellow; font-size: 16px; text-decoration: none; }`
- `strong { color: blue; }`

### Class Discussion

- Identify the parts of the following CSS rule?

**`h1{color:red; text-align: center;}`**

- What does the above CSS rule mean?

Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>

## Types of CSS

- **Cascading Style Sheet (CSS)** is used to set the style in web pages which contain HTML elements. It sets the background color, font-size, font-family, color, ... etc property of elements in a web pages. There are three types of CSS which are given below:
  - **Inline CSS:** contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using style attribute.
  - **Internal or Embedded CSS:** This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.
  - **External CSS:** External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, ... etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using **link** tag. This means that for each element, style can be set only once and that will be applied across web pages.

### Properties of CSS:

- Inline CSS has the highest priority, then comes Internal/Embedded followed by External CSS which has the least priority. Multiple style sheets can be defined on one page. If for an HTML tag, styles are defined in multiple style sheets then the below order will be followed.
- As Inline has the highest priority, any styles that are defined in the internal and external style sheets are overridden by Inline styles.
- Internal or Embedded stands second in the priority list and overrides the styles in the external style sheet.
- External style sheets have the least priority. If there are no styles defined either in inline or internal style sheet then external style sheet rules are applied for the HTML tags.

Sources: <https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<https://www.geeksforgeeks.org/types-of-css-cascading-style-sheet/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>



## How to Insert CSS

- **Inline**  
`<h1 style="color: red;">Heading</h1>`
- **Embedded** (<head>)  
`<style> ..... </style>`
- **Included** (<head>)  
`<link rel="stylesheet" type="text/css" href="style.css" />`

But, which is better?

Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>

## Linked Stylesheet | Example

### • HTML Code for test.htm

```
<html>
  <head>
    <title>Linked Stylesheet - Example</title>
    <link rel="stylesheet" type="text/css" href="mystyle.css">
  </head>
  <body>
    <h1>Lorem ipsum dolor</h1>
    <p>Lorem ipsum dolor sit amet, <em>consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
    <p>Ut wisi enim ad minim veniam, <strong>quis nostrud exerci tation</strong> ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat.
  </p>
</body>
</html>
```

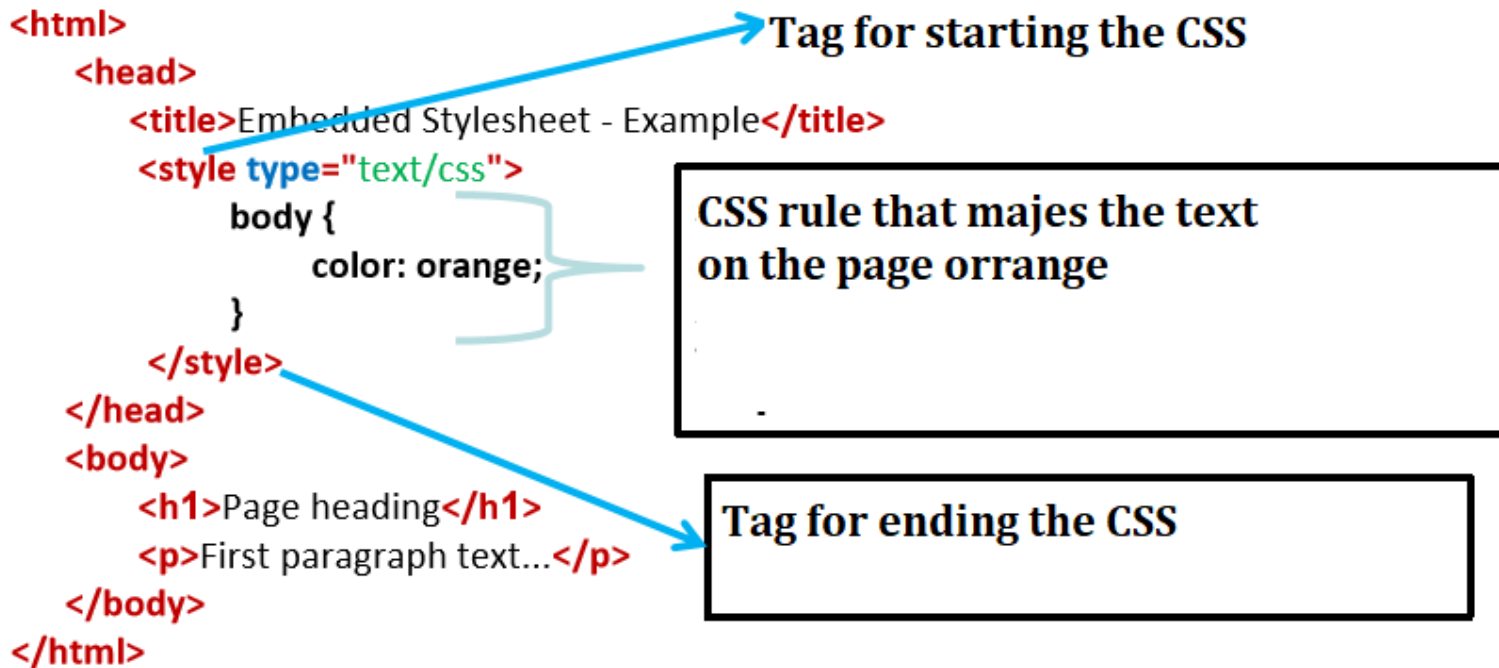
### • CSS code for stylesheet named mystyle.css

```
body { font-family: Tahoma; font-size: 14px; }
h1 { color: purple; text-align: center; }
p { text-align: justify; color: #965B30; }
strong { color: green; }
```



Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>

## Embedded Stylesheet | Example



Please note that in this case the CSS is within the `<head>...</head>` tags

Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>

## Inline Stylesheet | Example

```
<html>
  <head>
    <title>Inline Style - Example</title>
  </head>
  <body>
    <h1 style="color: orange; font-size: 18px">Page heading</h1>
    <p>First paragraph text...</p>
  </body>
</html>
```

Declaration

**Makes the font color of heading 1 <h1> orange and of size 18 points (pt.)**

Sources: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)  
<https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>  
<http://ctis.okcareertech.org/development/training/css/css-101/css-theory>



## Class Exercise | Discussion

- **Exercise 1**

Using one of the webpages created in the previous lessons apply:

1. An inline CSS that formats the text of the body to a blue color
2. An embedded CSS that formats the text of the headings at level 1 to a color of your choice, aligned to the right

- **Exercise 2**

1. Create a simple style sheet as follows:

```
body {  
    background-color: lightblue;  
}
```

```
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

2. Save as **mystyle.css**
3. Apply this style sheet (as an external CSS) to one of the pages you have created in a previous lesson
4. Make changes to the style sheet and reopen the page





## CSS Selectors

- CSS selectors **are patterns used to select** the element(s) you want to style.
- Common:
  - #id
    - Used to select elements based on their id attribute value
  - .class
    - Used to select elements based on their class attribute value
  - Element
    - Used to select elements based on their tag name
- Visit [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp) for a full list of CSS selectors

Sources: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

## ID selector: # & Example

- The **#id** selector is very useful as it allows us to set once an element for the whole page and simply refer to it by its ID from the CSS to apply formatting to it
- Syntax:  
**#idvalue { property1: value1; property2: value2; ... }**

### Example:

The id selector uses the id attribute of an HTML element to select a specific element

<style>

**#redText {color: red;}**

<head>

</style>

...

<body>

...

<span id="redText">Κείμενο</span>

<html>

<head>

<title>Class Selector - Example</title>

<link rel="stylesheet" type="text/css" href="mystyle.css">

</head>

<body>

<h1>Lorem ipsum dolor</h1>

<p id="mainparag">Lorem ipsum <em>consectetuer</em> aliquam erat.</p>

<p>Ut veniam, <strong>quis nostrud</strong> ullamcorper a commodo consequat.</p>

</body>

</html>

**#mainparag { color: green;}**

**Lorem ipsum dolor**

Lorem ipsum consectetur aliquam erat.

Ut veniam, quis nostrud ullamcorper a commodo consequat.

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

## Class selector: .class & Example

- The **.class** selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.
- You can also specify that only specific HTML elements should be affected by a class. To do this, start with the element name, then write the period (.) character, followed by the name of the class.
- Syntax:

```
.class {
  css declarations;
}
```

### Example:

The class selector selects elements with a specific class attribute.

```
<style>
```

```
.redText {color: red;}
```

```
</style>
```

```
...
```

```
<span class="redText">Κείμενο</span>
```

An element can have many classes

```
<span class="redText bold">
```

```
<html>
<head>
  <title>Class Selector - Example</title>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
  <h1 class="paint_red">Lorem ipsum dolor</h1>
  <p>Lorem ipsum <em class="paint_red">consectetuer</em> aliquam erat.</p>
  <p>Ut veniam, <strong class="paint_red">quis nostrud</strong> ullamcorper a commodo
consequat.</p>
</body>
</html>
```

```
.paint_red { color: red; }
```

Lorem ipsum dolor

Lorem ipsum ~~consectetuer~~ aliquam erat.

Ut veniam, ~~quis nostrud~~ ullamcorper a commodo consequat.

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)



## Element Selector

- The element selector selects **elements** based on the element name.
- You can select **all** <p> elements on a page like this (in this case, all <p> elements will have yellow background):

```
p {  
  background-color: yellow;  
}
```

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

## Grouping Selectors & Examples

If our page has more than one tags that we want to format with the same formatting, we can write a single CSS rule that will address all tags at once.

### Example:

Suppose we want to have all headings be of a blue font and with center alignment.

The rule would be:

```
h1, h2, h3, h4, h5, h6 { color: blue; text-align: center; }
```

### More examples:

```
b, strong { color: purple; }
```

```
h1, h2, p, li { color: #121212; text-align: left; }
```

```
h1, p {
  background-color: yellow;
}
```

← Styles ALL h1 and p

```
p .center {
  text-align: center;
  color: red;
}
```

← Styles ONLY <p> elements with class="center"

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)



## Class Discussion

- How would we style all paragraph elements with a class named “BC3” that would have white font and a red background?
- How would we style all paragraph elements according to class “center” and “BC3”?  
BC3 does what in (1) above and center applies center alignment, 200% font size and blue font to text.



## CSS Combinators

- **descendant selector (space)**

The descendant selector matches **all** elements that are descendants of a specified element.

**Example:**

```
p span {color: red;}
```

All span elements that are descendants of p elements

- **child selector >**

The child selector selects all elements that are the **immediate** children of a specified element.

**Example:**

```
div > p {  
  background-color: yellow;  
}
```

Selects all <p> elements that are **immediate** children of a <div> element

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)



# Class Exercises

## Exercise 1

What do the following selectors mean?

- `#p2 span {color: red;}`
- `.redText p span {color: red;}`

## Exercise 2

What is the difference between:

- `#header.callout {color: red;}`
- `#header .callout {color: red;}`

## Exercise 3

Using one of your webpages, apply all the CSS methods you've learned so far

## Exercise 4

- Visit <https://www.w3schools.com/cssref/tryse1.asp>
- Click on any selector to see which element(s) will get selected (and formatted) if you create a style for them





## Sibling Selectors

### General Sibling Selector

- The general sibling selector selects all elements that are siblings of a specified element.

#### Example:

```
p ~ ul {  
    background: #ff0000;  
}
```

Set a background color for all <ul> elements that are **preceded** by a <p> element with the **same parent**

### Adjacent Sibling Selector

- The adjacent sibling selector selects all elements that are the **adjacent siblings** of a specified element.
- Sibling elements must have the same parent element, and "adjacent" means "immediately following".

#### Example:

```
div + p {  
    background-color: yellow;  
}
```

Selects all <p> elements that are placed immediately after <div> elements

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

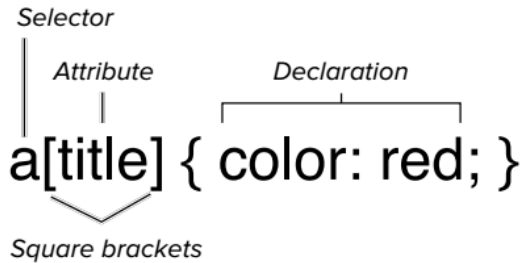


## Combinator Selectors

Format	Selector Name	Elements Are Styled If...	Compatibility
<b>a b c</b>	Descendent	<b>c</b> descendent of <b>b</b> descendent of <b>a</b>	IE4, FF1, O3.5, S1, C1, CSS1
<b>a * b</b>	Universal	<b>b</b> within a regardless of <b>b</b> 's parents	IE7, FF1, O4, S1, C1, CSS2
<b>a &gt; b</b>	Direct Child	<b>b</b> direct child of <b>a</b>	IE7 FF1, O3.5, S1, C1, CSS1
<b>a + b</b>	Adjacent Sibling	sibling <b>b</b> immediately after <b>a</b>	IE7, FF1, O5, S1, C1, CSS2
<b>a ~ b</b>	General Sibling	sibling <b>b</b> anywhere after <b>a</b>	IE8, FF1, O5, S1, C1, CSS2

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

## Selecting by Attribute



**A** The general syntax of an attribute selector.

Attribute Selectors			
Format	Name	Elements Are Styled If That Element:	Compatibility
[attr]	Attribute	has specified attribute	IE7, FF1.5, O5, S2, CSS2
[attr="value"]	Exact value	has specified attribute equal to exact value	IE7, FF1.5, O5, S2, CSS2
[attr~="value"]	Spaced List	has specified attribute equal to exact value within space-separated list	IE7, FF1.5, O5, S2, CSS2
[attr ="value"]	Hyphenated List	has specified attribute equal to exact value within hyphen-separated list	IE7, FF1.5, O5, S2, CSS2
[attr^="value"]	Begins with	has specified attribute equal to exact value at beginning	CSS3
[attr\$="value"]	Ends With	has specified attribute equal to exact value at end	CSS3
[attr*="value"]	Contains	has specified attribute equal to exact value anywhere	CSS3

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

## CSS Rules & Conflicts | Styles & Rules Priority

- As we have seen there are several ways to use CSS on our web page, so sometimes there may be more than one rules per element
- For **example**, a CSS rule in an **external CSS** may state paragraph font RED while an Embedded CSS may state paragraph font GREEN
- What will happen??
- Styles
- priorities take care of this...

### Styles priority

Rules that are more specific comes first

1. Inline styles
2. Internal style sheet
3. External style sheet
4. Browser default

`<style>`

`#redText {color: green;}`

`</style>`

...

`<span id="redText" style="color: red;">Hello</span>`

What will be the color of the text? **RED!**

### Rules priority

General list of the internal priorities for CSS

1. #id
2. .class
3. Element

`<style>`

`p { color: red; }`

`.myclass { color: green; }`

`</style>`

...

`<p class="myclass">Hello</p>`

Sources: <https://www.geeksforgeeks.org/types-of-css-cascading-style-sheet>  
[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

## CSS Comments

- We can import comments in our CSS (as in any other code)
- CSS comments are ignored by the browser
- A CSS comment starts with `/*` and ends with `*/`

```
div {
  text-align: center;
  /* This is a css comment, browser will ignore it */
  color: #de0023;
  font-family: Tahoma;
  /* ...This is another comment... */
}
```

## Firefox Developer Tools

- **Page Inspector:** [https://developer.mozilla.org/en-US/docs/Tools/Page\\_Inspector](https://developer.mozilla.org/en-US/docs/Tools/Page_Inspector)
  - Assists in tracing CSS rules
  - Allows to change and experiment with CSS properties and the values virtually (without actually changing our web page)
  - Very useful while working with HTML & CSS
- **Web Console:** [https://developer.mozilla.org/en-US/docs/Tools/Web\\_Console](https://developer.mozilla.org/en-US/docs/Tools/Web_Console)
  - Checks and traces errors on our pages
  - Displays HTTP and Ajax requests etc.
  - Very useful while working with JavaScript (& CSS)

Source: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)



## Typography

The CSS properties that allow you to define the color, size, spacing, and shape of text are:

- font
- font-size
- font-family
- font-weight
- font-style
- text-align
- text-indent
- text-decoration
- text-transform
- text-shadow
- line-height
- letter-spacing
- word-spacing
- color

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## Font Properties

There are seven font sub-properties, including:

- **font-size:** sets the height of the font.
- **font-family:** defines the font that is applied to the element.
- **font-weight:** sets the weight or the thickness of the font.
- **font-style:** makes the text appear italicized or oblique.
- **font-variant:** changes target text to small caps.
- **font-stretch:** this property sets the font width, such as condensed or expanded.
- **line-height:** defines the amount of space above and below inline elements.

CSS Font Properties			
property	type	default	comments
font	URI	inherited (-)	overrides the next 6 below
font.family	string+	inherited (-)	
font.weight	..., -2, -1, 0, 1, 2,...	inherited (0)	1=bold, 2=extra-bold, etc.
font.expansion	..., -2, -1, 0, 1, 2,...	inherited (0)	0=normal, 1=expanded, etc.
font.style	(italic slanted roman small-caps)+	inherited (roman)	
font.size	..., -2, -1, 0, 1, 2,... length	0 (-)	n=1.2^n times parent's size; length in pt
encoding	encoding	inherited (none)	Postscript-like encoding vector

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## Font Size

- Set via **font-size** property.
- Sets the height of the font.
  - `<size>` i.e. **14px** ή **2em**
  - `<%>` i.e. **80%** percentage
  - small
  - medium
  - large
  - x-small xx-small
  - x-large xx-large
  - smaller larger
  - percentage
  - inherit

Unit	Description
<b>em</b>	Relative to the font-size of the element (gem means 2 times the size of the current font)
<b>ex</b>	Relative to the x-height of the current font (rarely used)
<b>ch</b>	Relative to width of the "0" (zero)
<b>rem</b>	Relative to font-size of the root element
<b>vw</b>	Relative to 1% of the width of the viewport*
<b>vh</b>	Relative to 1% of the height of the viewport*
<b>vmin</b>	Relative to 1% of viewport's* smaller dimension
<b>vmax</b>	Relative to 1% of viewport's* larger dimension
<b>cm</b>	centimeters
<b>mm</b>	millimeters
<b>in</b>	inches (lin = 96px = 2.54cm)
<b>px *</b>	pixels (ipx = 1/96th of lin)
<b>pt</b>	points (lpt = 1/72 of lin)
<b>pc</b>	picas (lpc = 12 pt)

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>



## Font Family

- Set via **font-family** property
- Defines the font that is applied to the element.

**h1, h2, h3, h4, h5, h6 {**

**font-family:** "Times New Roman", **Serif;**

**}**

### font-family (generic families)

Name	Example
Serif	Times New Roman
Sans-serif	Helvetica and Arial
Monospace	Courier New
Cursive	<i>Brush Script MT</i>
Fantasy	Papyrus

## Font Weight

Set via **font-weight** property

Sets the weight or the thickness of the font.

Values:

- normal
- bold
- lighter
- bolder
- 100 - 900

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## Font Variant

- Set via **font-variant** property
- Changes target text to small caps.
- Values
  - normal
  - small-caps

Normal

SMALLCAPS

## CSS Text Properties

Property	Values
<b>text-align</b>	left, right, center, justify
<b>text-decoration</b>	none, overline, underline, line-through
<b>text-indent</b>	length (in pixels), or percentage %
<b>text.transform</b>	none, capitalize, uppercase, lowercase
<b>vertical-align</b>	baseline, top, middle, bottom, length (in pixels)
<b>word-spacing</b>	normal or length (in pixels)
<b>line-height</b>	normal, number (in pixels), % of font size
<b>letter-spacing</b>	normal or length (in pixels)
<b>color</b>	Color name, hexadecimal code

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>



## Text Alignment

Set via **text-align** property.

Sets alignment of the text.

Values:

- left (default)
- right
- center
- Justify

## Text Indent

- Set using **text-indent** property.
- Indent of first line in paragraph
- Possible values
  - Size **2em 30px**
  - Percentage **10%** (refer to the width of the containing block)

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

# Vertical Align & White Space

## Vertical Align

Set via vertical-align property

Values:

- baseline
- super
- sub
- top
- middle
- bottom
- text-top
- text-bottom
- Percentage or length

Mad	Hatter	baseline
Mad	Hatter	superscript
Mad	Hatter	subscript
Mad	Hatter	top
Mad	Hatter	middle
Mad	Hatter	bottom
Mad	Hatter	text-top
Mad	Hatter	text-bottom

## White Space

The white-space property specifies how white-space inside an element is handled.

```
p {
  white-space: nowrap;
}
```

Value	Description
normal	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary. This is default
nowrap	Sequences of whitespace will collapse into a single whitespace. Text will never wrap to the next line. The text continues on the same line until a   tag is encountered
pre	Whitespace is preserved by the browser. Text will only wrap on line breaks. Acts like the <pre> tag in HTML
pre-line	Sequences of whitespace will collapse into a single whitespace. Text will wrap when necessary, and on line breaks
pre-wrap	Whitespace is preserved by the browser. Text will wrap when necessary, and on line breaks

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

# Text Decoration, Transformation & Shadow

## Text Decoration

Set via **text-decoration** property

Values:

- none
- underline
- overline
- line-through

underline

overline

~~line-through~~

underline overline line-through

## Text Transformation

Set via **text-transform** property

Values:

- none
- capitalize (capitalize first letter)
- lowercase
- uppercase

## Text Shadow

Set via text-shadow property.

Example:

```
{ text-shadow: horizontal vertical blur color; }
```

```
{ text-shadow: 2px 2px 2px #000000; }
```

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## Line Height

- Set via **line-height** property
- Defines height between lines
- Possible values:

Value	Description
<b>normal</b>	A normal line height. This is default
<b>number</b>	A number that will be multiplied with the current font size to set the line height
<b>length</b>	A fixed line height in px, pt, cm, etc.
<b>%</b>	A line height in percent of the current font size

## CSS Background Properties

property	values
<b>background-color</b>	transparent, color
<b>background-image</b>	none, url("URL")
<b>background-repeat</b>	repeat, repeat-x, repeat-y, no-repeat
<b>background-position</b>	left top, left center, left bottom, right top, right center, right bottom, center top, center center, center bottom, number (px or %)
<b>background-attachment</b>	scroll, fixed

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

# Background

## Background Image

- Set via background-image property.
- Specifies the background of an element.

### Examples:

- **background-image:** `url(image.gif);`
- **background-repeat:**  
`repeat;` (default)  
`repeat-x;`  
`repeat-y;`  
`no-repeat;`

## Background Attachment

The **background-attachment** property sets whether a background image will scroll with the rest of the page, or be fixed.

### Values:

`scroll;`  
`fixed;`  
`local;`

## Background Position

Set via **background-position** property.

Values:

`top;`  
`bottom;`  
`left;`  
`right;`  
`center;`  
`<<length>>`  
`<<percentage>>`

## Background Size

The **background-size** property specifies the size of the background images.

### Values:

`cover;`  
`contain;`  
`<<length>>`  
`<<percentage>>`

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

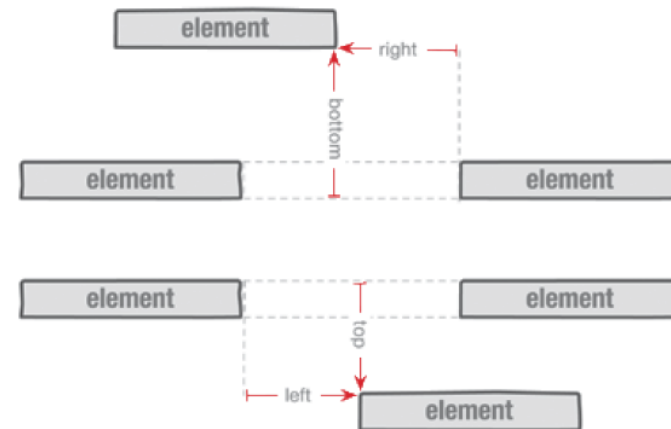
# Position Property

- The **position** property specifies the type of positioning method used for an element

Value	Description
<b>static</b>	Default value. Elements render in order, as they appear in the document flow
<b>absolute</b>	The element is positioned relative to its first positioned (not static) ancestor element
<b>fixed</b>	The element is positioned relative to the browser window
<b>relative</b>	The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position
<b>sticky</b>	The element is positioned based on the user's scroll position A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed). Note: Not supported in IE/Edge 15 or earlier. Supported in Safari from version 6.1 with a -webkit- prefix.

## Relative Position

- Set via **position: relative;**
- An element whose position property has the value relative is first laid out just like a **static** element. The rendered box is then shifted vertically (according to the top or bottom property) and/or horizontally (according to the left or right property)



Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>



## Absolute Position

- Set via `position: absolute;`
- Completely removed from the document flow
- Positioned with respect to its containing block

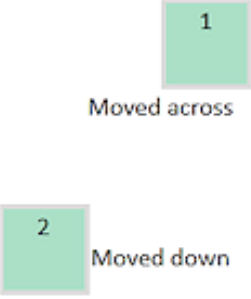

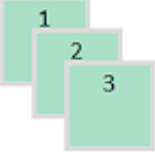


## Fixed Position

- Set via `position: fixed;`
- Fixed positioning is a subcategory of absolute positioning.
- An element whose position property is set to fixed always has the **viewport** as its containing block

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

# Positioning Examples

 <p>Default position</p>	 <p>Moved across</p> <p>Moved down</p>	 <p>Centered horizontally automatically</p>
	<code>position:relative;</code> <code>1. left:100px;</code> <code>2. top:50px;</code>	<code>margin:0 auto;</code>
 <p>Side-by-side</p>	 <p>Overlapping</p>	 <p>Out of order</p>
<code>1: position:absolute;</code> <code>top:40px;left:40px;</code> <code>2: position:absolute;</code> <code>top:40px;left:100px;</code>	<code>position:absolute;</code> <code>1: top:10px;left:30px;</code> <code>2: top:30px;left:50px;</code> <code>3: top:50px;left:70px;</code>	<code>position:absolute;</code> <code>1: top:30px;left:50px;</code> <code>2: top:50px;left:70px;</code> <code>3: top:10px;left:30px;</code>

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>



## Visibility

Set via **visibility** property.

- Values:
  - **visible**: the object is visible on the page
  - **hidden**: the object is **NOT** visible on the page but still holds space on the page

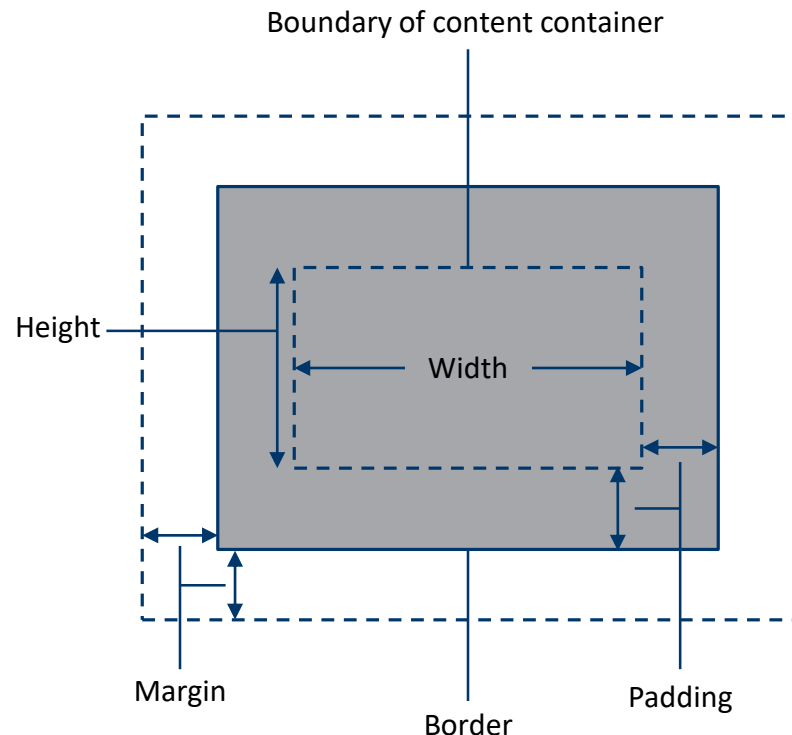
### Class Exercise 5

- Apply all you have learned so far to create an external CSS
- Save as myFirstCss.css and apply it to one of your webpages

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## CSS Box Model

- In CSS, the term "box model" is used when talking about design and layout.
- All HTML elements can be considered as boxes.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.
- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent



Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
[https://www.w3schools.com/Css/css\\_boxmodel.asp](https://www.w3schools.com/Css/css_boxmodel.asp)  
<https://cssreference.io/>

# Border

- The border is defined as the boundary from the top, bottom, left and right of the box of an element
- With the border property we can set the width, style, color of the border

**Example:** `p { border: 1px solid red; }`

sets the border of each page paragraph to a red 1pixel line

**Set via `border: <width> <style> <color>`**

- Syntax for border settings in all 4 directions
- We can set the parameters for each side differently/separately by using the following:

- `border-top`
- `border-bottom`
- `border-left`
- `border-right`

`border-width`

- In pixels: `3px` or
- `thin`
- `medium`
- `thick`

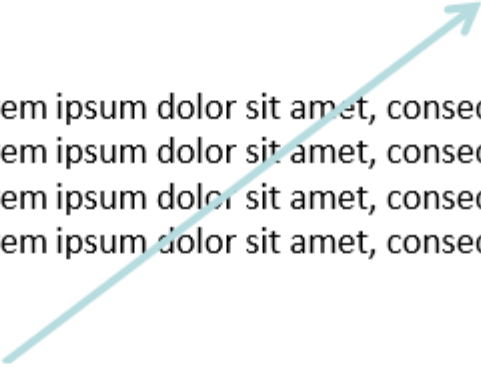
`border-style`

none	
dotted	.....
dashed	- - -
solid	_____
double	=====
groove	_____
ridge	_____
inset	_____
outset	_____

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## Border Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS border Property - Example 8</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="css/example8.css">
  </head>
  <body>
    <p class="border1">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <p class="border2">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <p class="border3">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <p class="border4">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </body>
</html>
```



example8.css

```
.border1 { border: 1px solid #000099; }
.border2 { border: 3px solid #999900; }
.border3 { border: 4px double #999900; }
.border4 { border: 2px dashed #006633; }
```

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>



## Class Exercise | Discussion

- Try the previous border example in one of your web pages (create the css style sheet and use it as an external css to one of your pages)

### example8.css

```
.border1 { border: 1px solid #000099; }  
.border2 { border: 3px solid #999900; }  
.border3 { border: 4px double #999900; }  
.border4 { border: 2px dashed #006633; }
```

# Padding Property

## Definition:

**Padding** is defined as the space that exists between the border and the content of an element

- Padding can be set for every side (`padding-top`, `padding-right`, `padding-bottom` and `padding-left` )
- Padding is affected by the `background-color`
- The value for padding can be a pixel number or a percentage of the width of the element (parent)

## Example 1:

```
p { padding: 8px; }
div { padding: 2%; }
```

## Example 2:

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS - Παράδειγμα 9</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="css/example9.css" >
  </head>
  <body>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy
    nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud
    exerci tation ullamcorper suscipit.
    </p>
    <p class="yellowbg">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod
    tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
    ullamcorper suscipit.
    </p>
  </body>
</html>
```

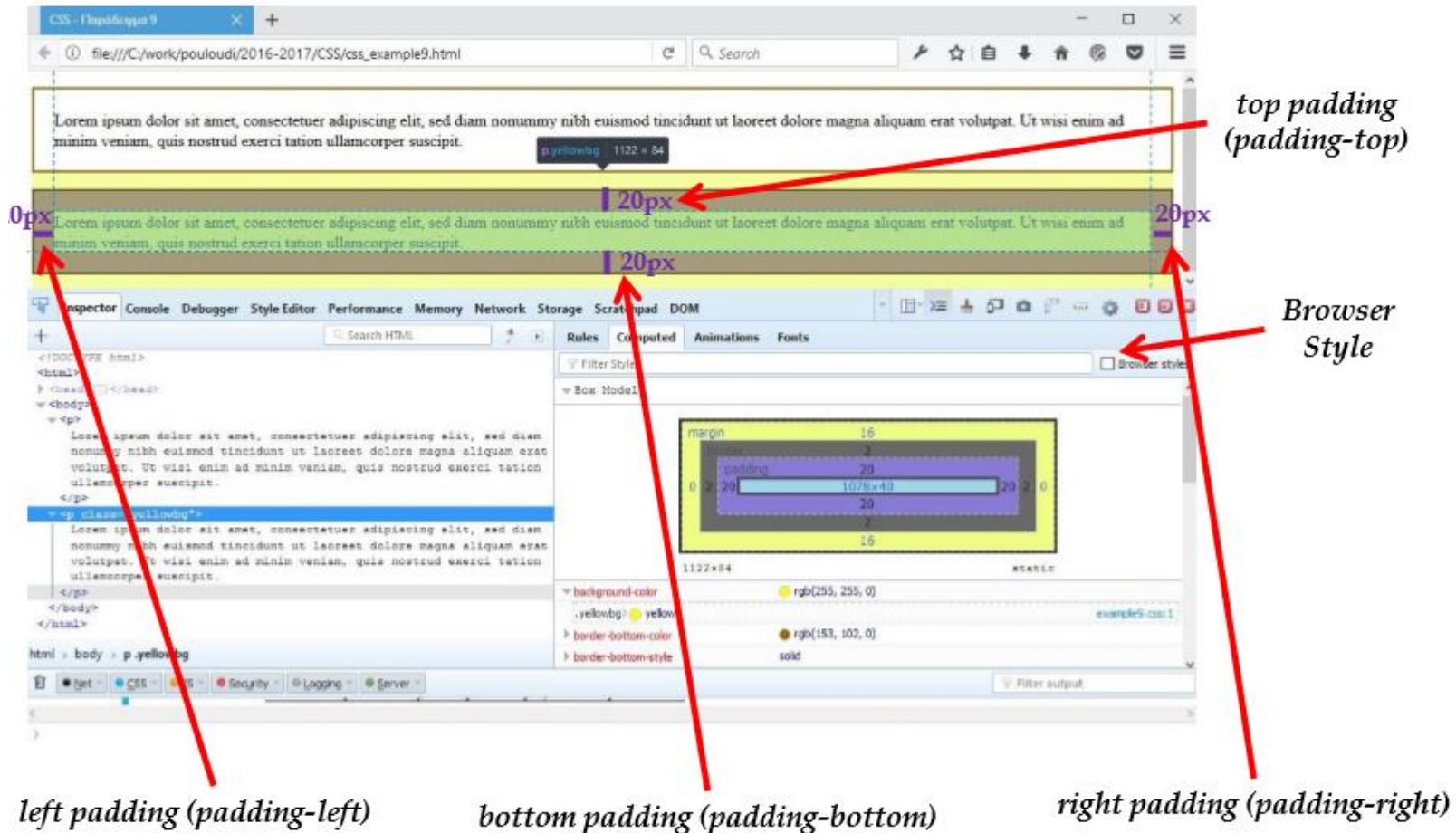
example9.css

```
.yellowbg { background-color: yellow; }
p { border: 2px solid #996600; padding: 20px; }
```

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>



# Padding **Example** in Inspector



# CSS Margin Property

## Definition:

**Margin** is defined as the empty space (top, bottom, left, right) that exists outside the border around an element

- Margin sets the space that will remain empty around the element
- Margin can be set for every side ([margin-top](#), [margin-right](#), [margin-bottom](#) and [margin-left](#). )
- Margin is **not** affected by the [background-color](#)
- The value for padding can be a pixel number or a percentage of the width of the element or auto

## Examples:

```
p { margin: 5px; }
div { margin: 2%; }
h1 { margin: auto; }
```

example10.html

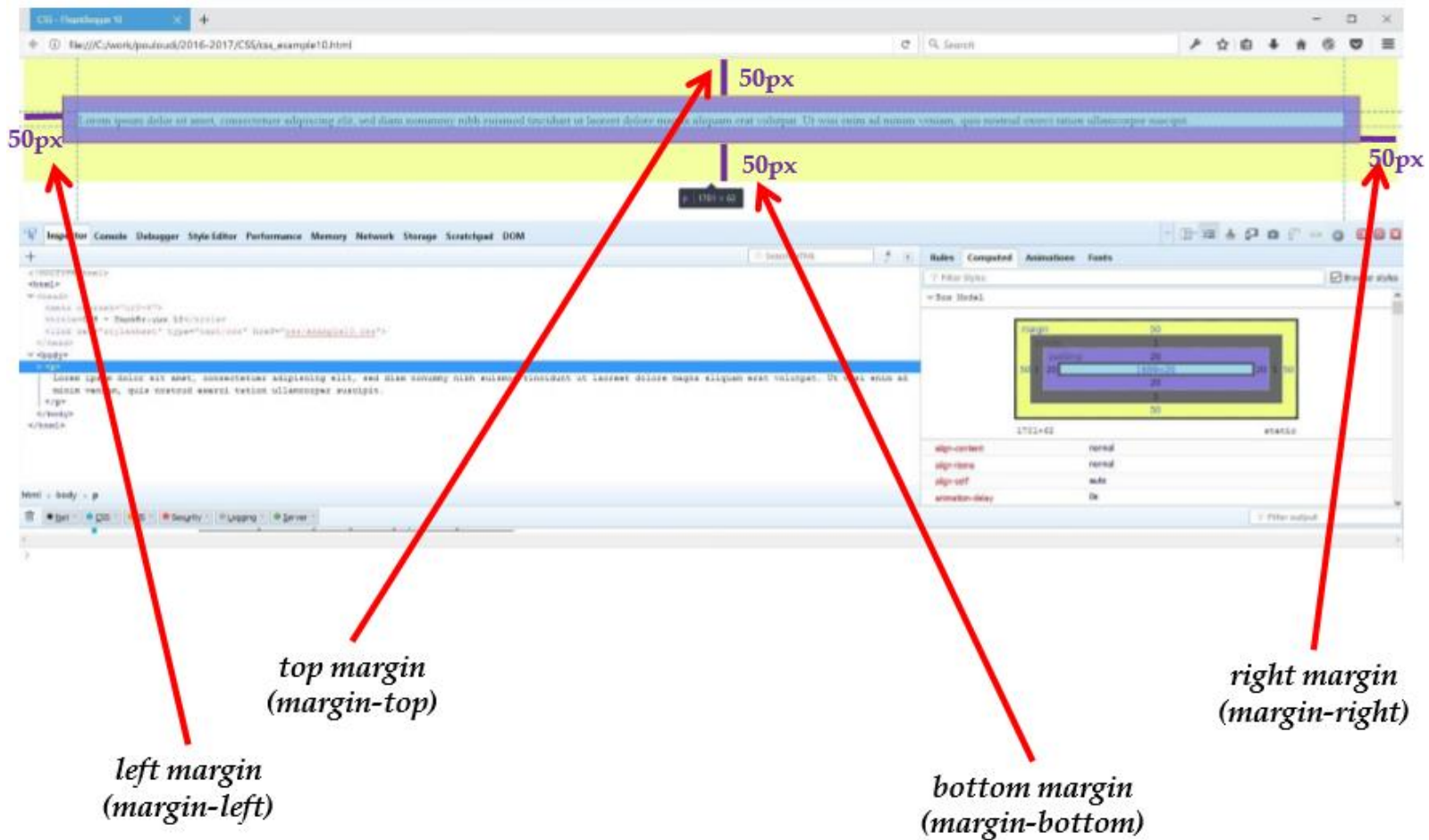
```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS - Example 10</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="css/example10.css">
  </head>
  <body>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod
    tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci
    tation ullamcorper suscipit.
    </p>
  </body>
</html>
```

example10.css

```
p {
  border: 1px solid red;
  padding: 20px;
  margin: 50px;
  background-color: #dddddd;
}
```

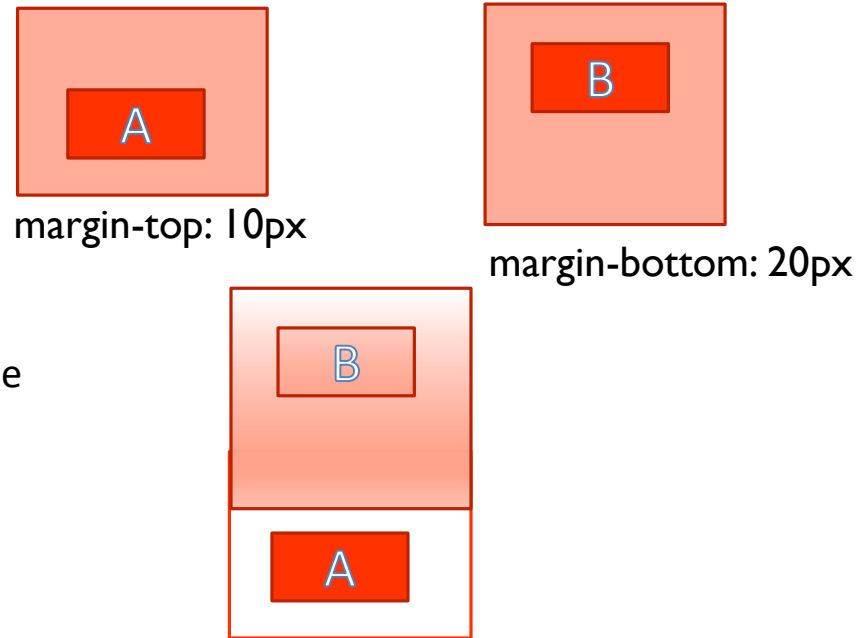
Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## Margin Example in Inspector



## Margin | Padding

- **margin**
  - `margin-top`
  - `margin-right`
  - `margin-bottom`
  - `margin-left`
- Set via **margin** property
- margins of 2 neighbor elements merge
- **padding**
  - `padding-top`
  - `padding-right`
  - `padding-bottom`
  - `padding-left`



Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>

## Width & Height Properties

- Define the size of the element
- The height and width properties do not include padding, borders, or margins; they set the height/width of the area inside the padding, border, and margin of the element!
- Block level elements occupy 100% of the length of the element that contains them, by default, while they have height auto
- Inline elements have auto width and height
- Width and height does not apply to inline elements, except for images (<img>)
- We can use height and width to size any block level element

### Examples:

```
div { width: 960px; }
div { width: 90%; }
```

### Links

**a:visited**

**a:hover**

**a:active**

```
a:visited {
  color: #00FF00;
}
```

Note **a:hover** MUST come after **a:link** and **a:visited** in the CSS definition in order to be effective!  
**a:active** MUST come after **a:hover** in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.

Sources: [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)  
<https://cssreference.io/>



**It is time for a Knowledge Check!**



## Sample Questions

1. \_\_\_\_\_ is the space between the content of an element and its border.

- A. Margin
- B. Box
- C. Width
- D. Padding

2. What would happen if we used

**li { text-align: left; }**

**li:nth-child(even) { text-align: right; }**

in a CSS?

- A. Every lists' items of our website would have alternate text alignment (left, right, left etc.)
- B. Every paragraph of our website would have alternate text alignment (left, right, left etc.)
- C. Every lists' items of our website would have alternate text alignment (right, left, right etc.)
- D. Every paragraph of our website would have alternate text alignment (right, left, right etc.)

3. Consider the code to the right:

What color would the word "Hello" be shown in?

- A. Green
- B. Red
- C. Black
- D. Blue

```
<style>
    #redText {color: green;}
</style>
.....
<span id="redText" style="color: blue;">Hello</span>
<span id="redText">HEY!</span>
</body>
</html>
```



## Sample Questions | Answers

1. \_\_\_\_\_ is the space between the content of an element and its border.

- A. Margin
- B. Box
- C. Width
- D. **Padding**

2. What would happen if we used

```
li { text-align: left; }
```

```
li:nth-child(even) { text-align: right; }
```

in a CSS?

- A. **Every lists' items of our website would have alternate text alignment (left, right, left etc.)**
- B. Every paragraph of our website would have alternate text alignment (left, right, left etc.)
- C. Every lists' items of our website would have alternate text alignment (right, left, right etc.)
- D. Every paragraph of our website would have alternate text alignment (right, left, right etc.)

3. Consider the code to the right:

What color would the word "Hello" be shown in?

- A. Green
- B. Red
- C. Black
- D. **Blue**

```
<style>
    #redText {color: green;}
</style>
....
<span id="redText" style="color: blue;">Hello</span>
<span id="redText">HEY!</span>
</body>
</html>
```





## Additional Exercises

1. Using on of the web pages you have created so far, create an external style sheet that will format your elements to your liking
2. Create a web page that will an online resume with
  - Title of the page: CV – Your name
  - Top of the page Your name
  - Heading 1 is left aligned, bold, Blue font, Arial 16pt
  - Bullets are indented, bold and black
  - Your projects are listed in a table
3. Create a sample project for your application
  - Create a main index.html file
  - Specify the needed image and styling folders, in order to apply file structure to your project
  - Include at least an external CSS file into your project
  - You are free to create more than one HTML or CSS files and use them into your page



**Any Questions?**



## References | Further Reading

- [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)
- <http://www.htmldog.com/guides/css/>
- <http://www.tizag.com/cssT/index.php>
- <http://www.w3.org/Style/CSS/>
- <https://css-tricks.com/>
- <https://marksheet.io/css-selectors.html>
- <https://vanseodesign.com/css/css-positioning/>
- <https://quizlet.com/10764751/css-tutorial-css-basic-flash-cards/>
- <https://scrimba.com/g/gintrotocss>
- [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Introduction](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Introduction)
- <https://developers.google.com/web/tools/chrome-devtools>
- <http://ctis.okcareertech.org/development/training/css/css-101/css-theory>
- [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)
- <https://www.geeksforgeeks.org/types-of-css-cascading-style-sheet>
- [https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling\\_text/Fundamentals](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals)
- <https://cssreference.io/>
- [https://www.w3schools.com/Css/css\\_boxmodel.asp](https://www.w3schools.com/Css/css_boxmodel.asp)



## PeopleCert Values Your Feedback

Like the course?

Have something to say?

Send us your comments at:  
**[academic@peoplecert.org](mailto:academic@peoplecert.org)**

# Thank You!

For latest news and updates follow us.



[linkedin.com/company/peoplecert-group/](https://linkedin.com/company/peoplecert-group/)

[twitter.com/peoplecert](https://twitter.com/peoplecert)

[youtube.com/channel/UCLBidKZS9Xk08f\\_PM5Edtfg](https://youtube.com/channel/UCLBidKZS9Xk08f_PM5Edtfg)

[facebook.com/peoplecert.org](https://facebook.com/peoplecert.org)