

ITEd2331: Fundamentals of Web System & Technologies



Today's schedule

- Syllabus
- Course Info
- Browsers! The Internet!
- A little bit about HTML and CSS

Syllabus

What this course is all about ?

"Fundamentals" of Web Systems and Technologies

- You will learn basics of the web ecosystem
- An introduction to web programming

Q: What does that mean, exactly?

What this course is all about ?

Web Programming "Fundamentals"

- An **opinionated, hopefully frustration-free** introduction to web programming

Opinionated:

- There are many ways to do things on the web: can't learn them all at once!
- We will do what I think you need to know as a beginner

Hopefully frustration-free:

- We will go slowly through the essential concepts and speed

ITEd2331 Goals

If you never take another web programming class again, you will leave the course with the following skills:

- Have the **foundation** to pursue the areas of web programming that you're interested in (if you choose)
- Create **attractive, small scale web sites or apps** that at least mostly work on phones
- Have the **vocabulary and background knowledge** to understand technical writing/discussions about the web (e.g. web API documentation; random blog posts)

(ITEd2331 Non-goals)

This course is **not** a class to take to learn how to code.

This course is **not** a class that will turn you into a senior frontend/backend developer.

- ❑ Nor is any class; software takes years of experience to develop expertise.

This course is **not** a class that will teach you all there is to know about web programming.

- ❑ For example, we will **not** teach how to support old browsers,

ITEd2331, in detail

HTML

CSS

JavaScript

~~Backend basics~~

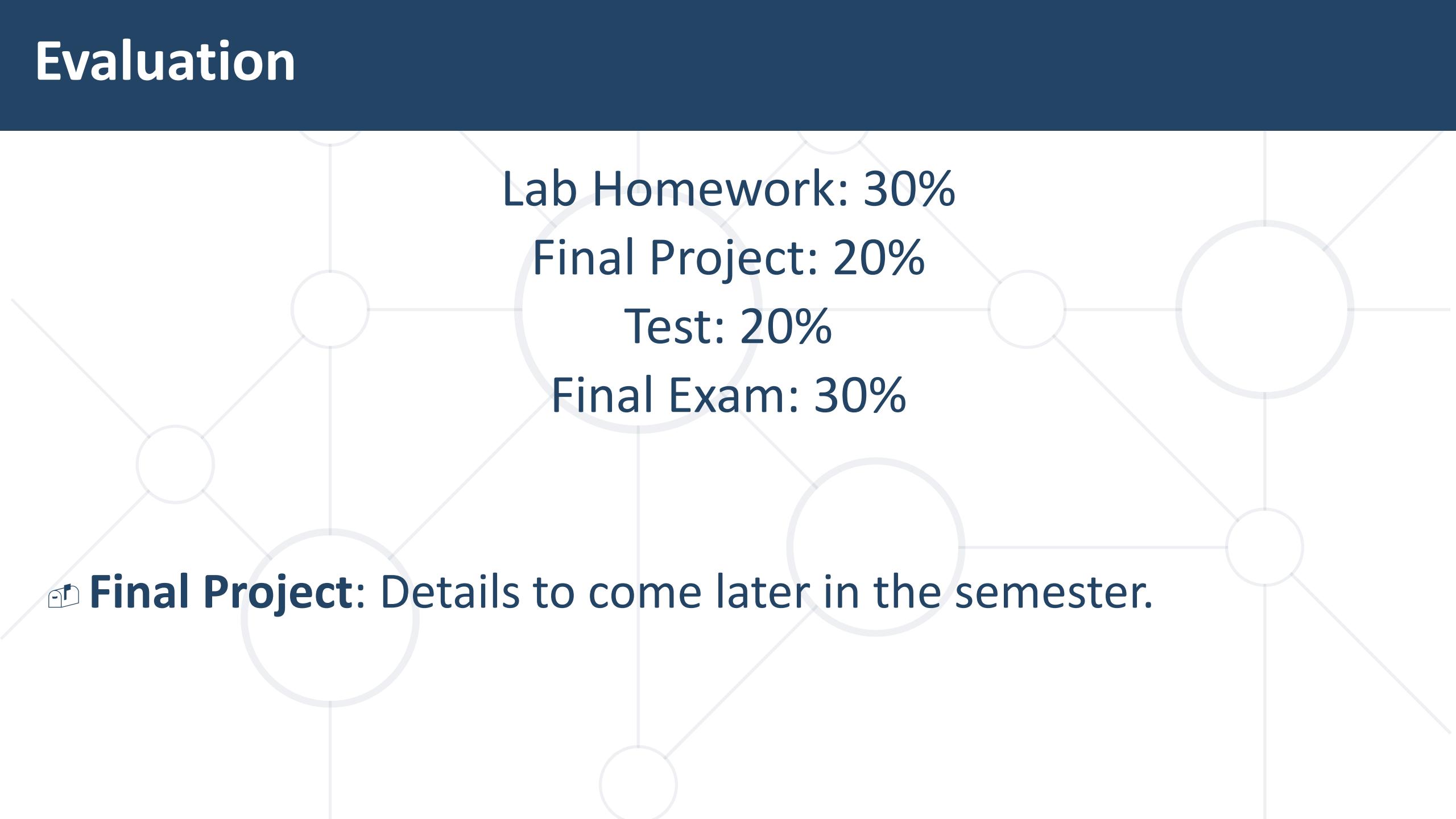
~~Server on NodeJS + Express~~

~~Database via MongoDB and Mongoose~~

Course info



Evaluation



Lab Homework: 30%
Final Project: 20%
Test: 20%
Final Exam: 30%

☞ **Final Project:** Details to come later in the semester.

Lateness policy

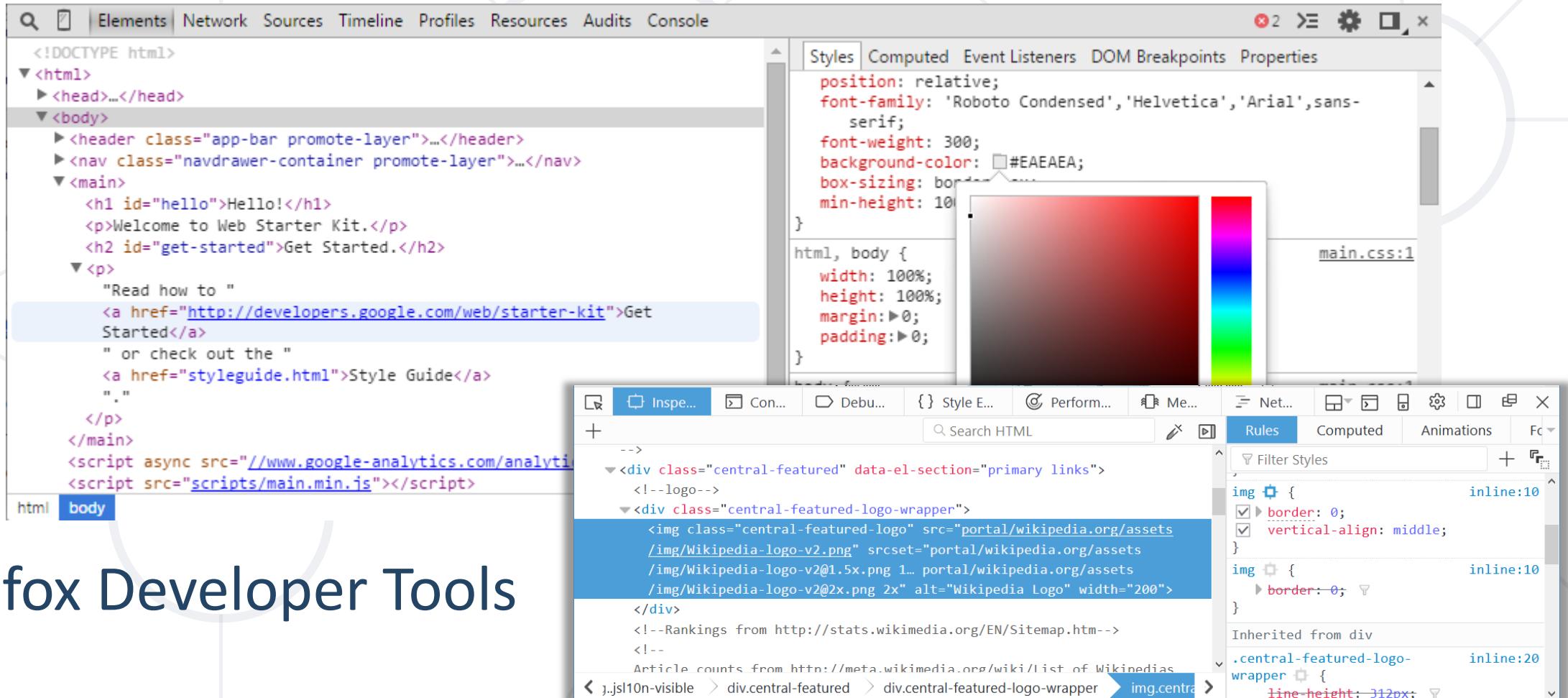
- Every homework may be submitted up to 48 hours after the deadline, without penalty.
- Homework submitted on time will receive a small bonus to their homework score.
- Submissions are **not accepted** beyond the 48-hour grace period. The grace period is strictly enforced.
- The final project must be turned in on time.

Browser and Text editor/IDE

- **Text editor:** You can use whatever you want. We recommend Visual Studio Code.
- **Browser:** Your code must work on Chrome, as that is what your TAs will use when grading your homework. It will not be tested in any other browser.

Browser Developer Tools

- Chrome DevTools – just press [F12] in Chrome



- Firefox Developer Tools

Schedule

Lecture:

- Tuesday 1 & 2

Lab:

- Tuesday 5-6

- We usually do not take attendances

- But please come!

- There will no be makeups if you miss in class assessments

Disclaimer

- ☞ **Everything is subject to change.** Including everything I've just told you and everything I'm about to tell you.
- ☞ **You might encounter:**
 - ☞ Bugs in homework
 - ☞ Awkward lectures
 - ☞ Things that are too hard / too easy

Today's schedule

~~- Syllabus~~

~~Course Info~~

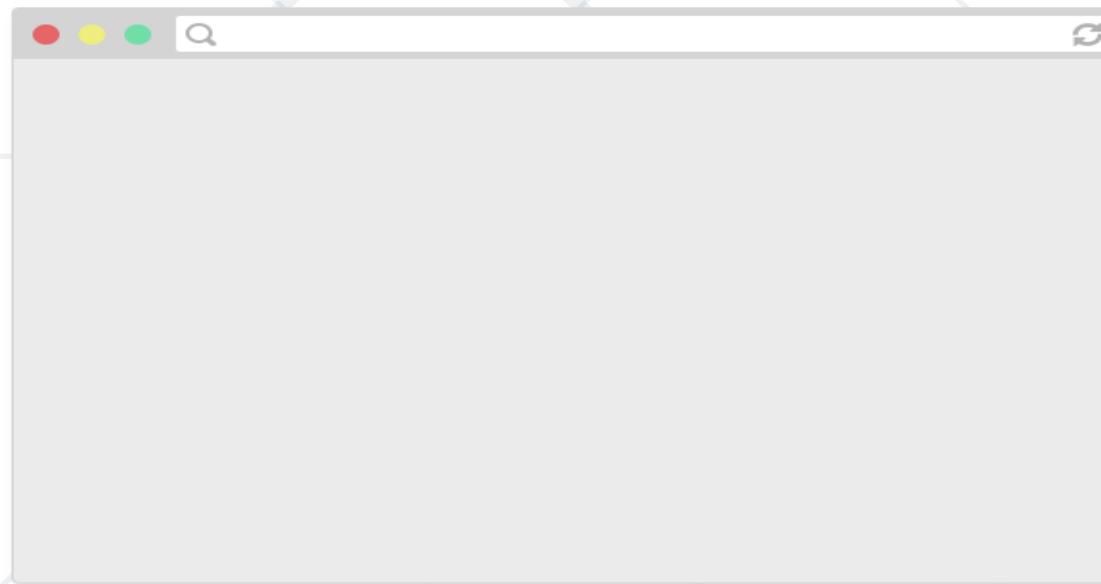
~~Browsers! The Internet!~~

~~A little bit about HTML and CSS~~

Chapter - 1

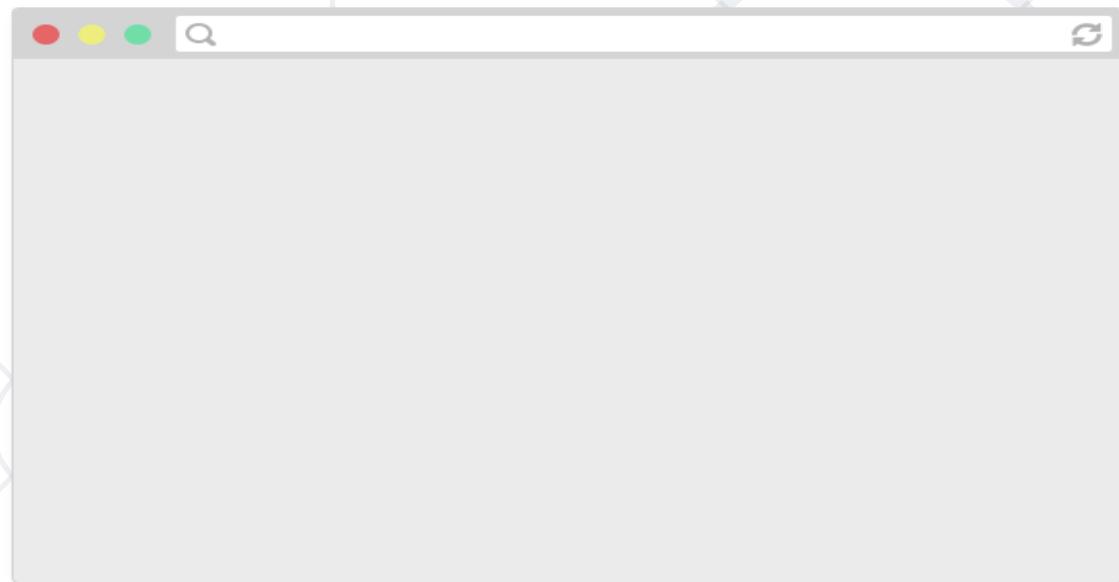
- **Browsers!**
- **The Internet!**
- **The web!**

How do web pages work?



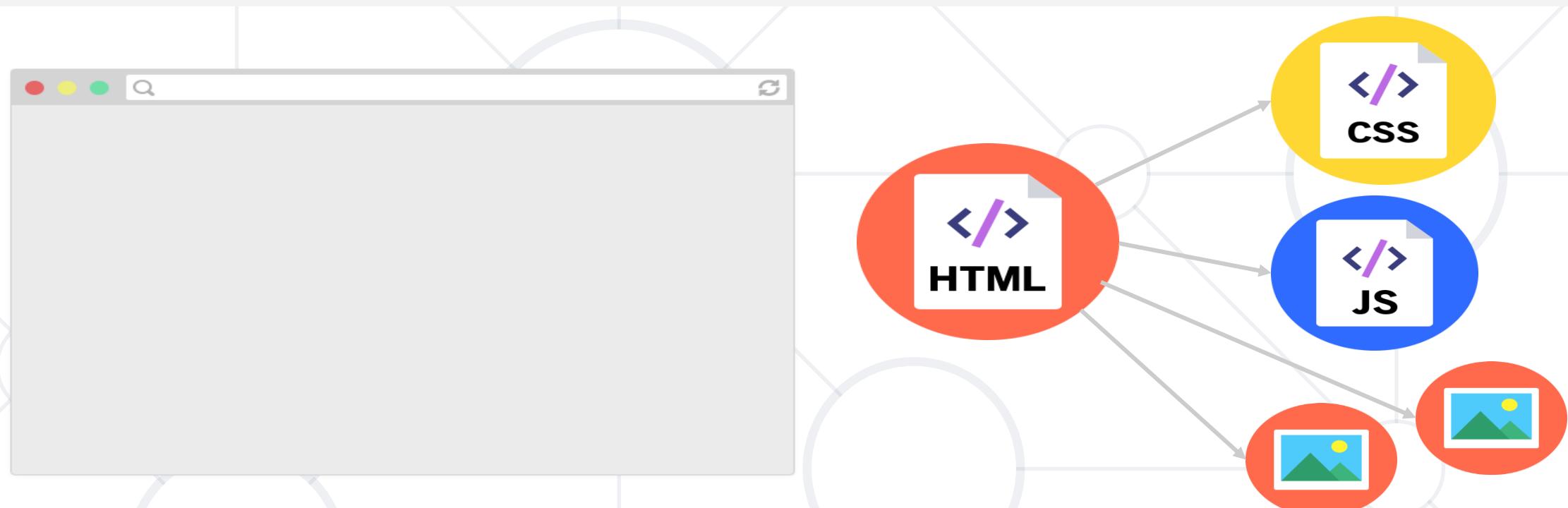
Browsers are applications that can display web pages.
E.g. Chrome, Firefox, Safari, Internet Explorer, Edge, etc.

How do web pages work?



Web pages are written in a markup language called **HTML**, so browsers display a web page by reading and interpreting its HTML.

How do web pages work?



The HTML file might link to other resources, like images, videos, as well as **JavaScript** and **CSS** (stylesheet) files, which the browser then also loads.

How do web pages work?

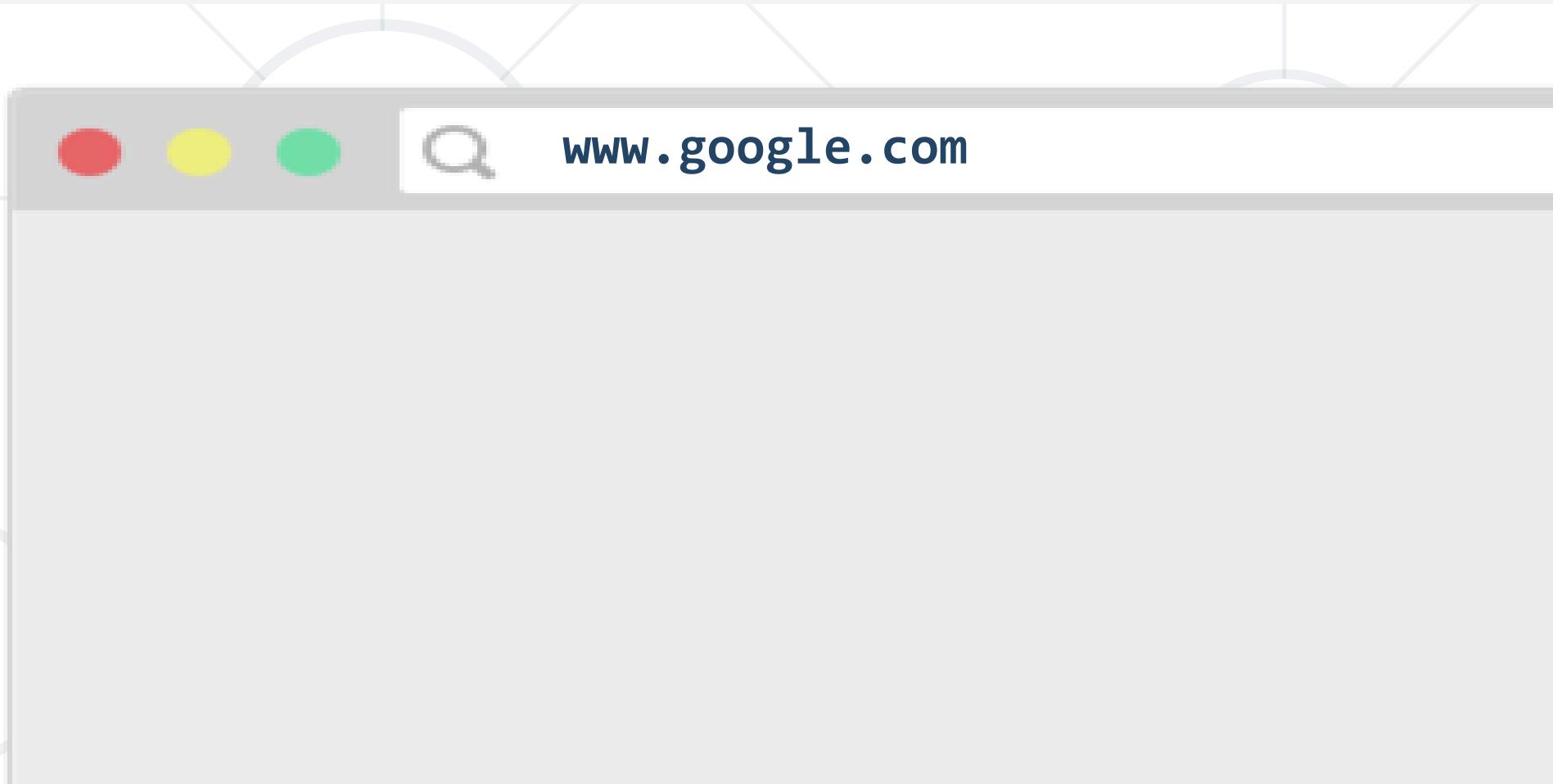


A **web server** is a program running on a computer that delivers web pages in response to requests.

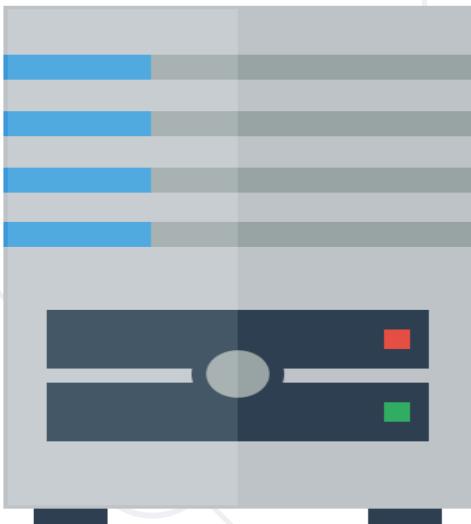
It either stores or generates the web page returned.

How do web pages work?

1. You type in a URL, which is the address of the HTML file on the internet.

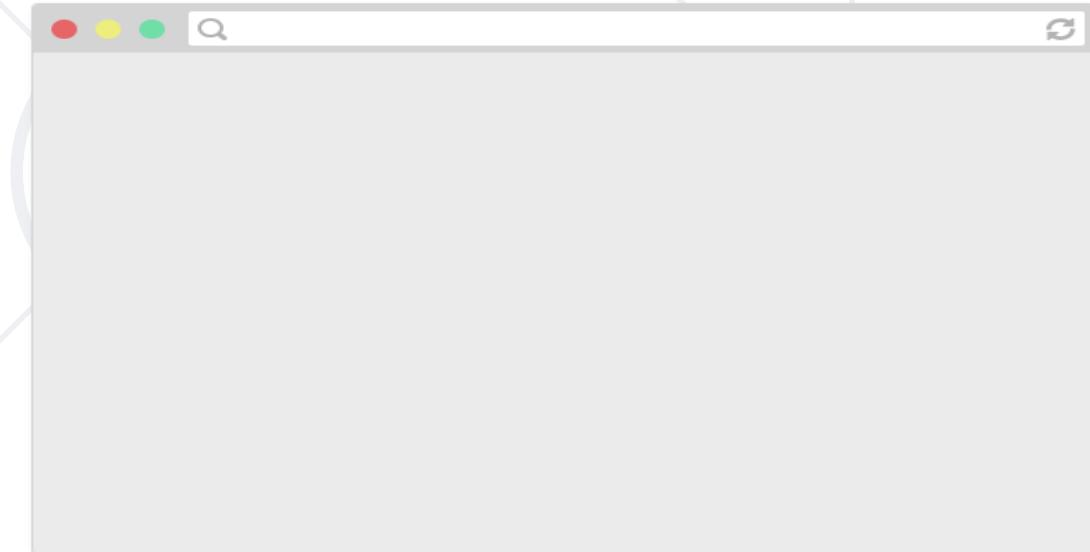


How do web pages work?

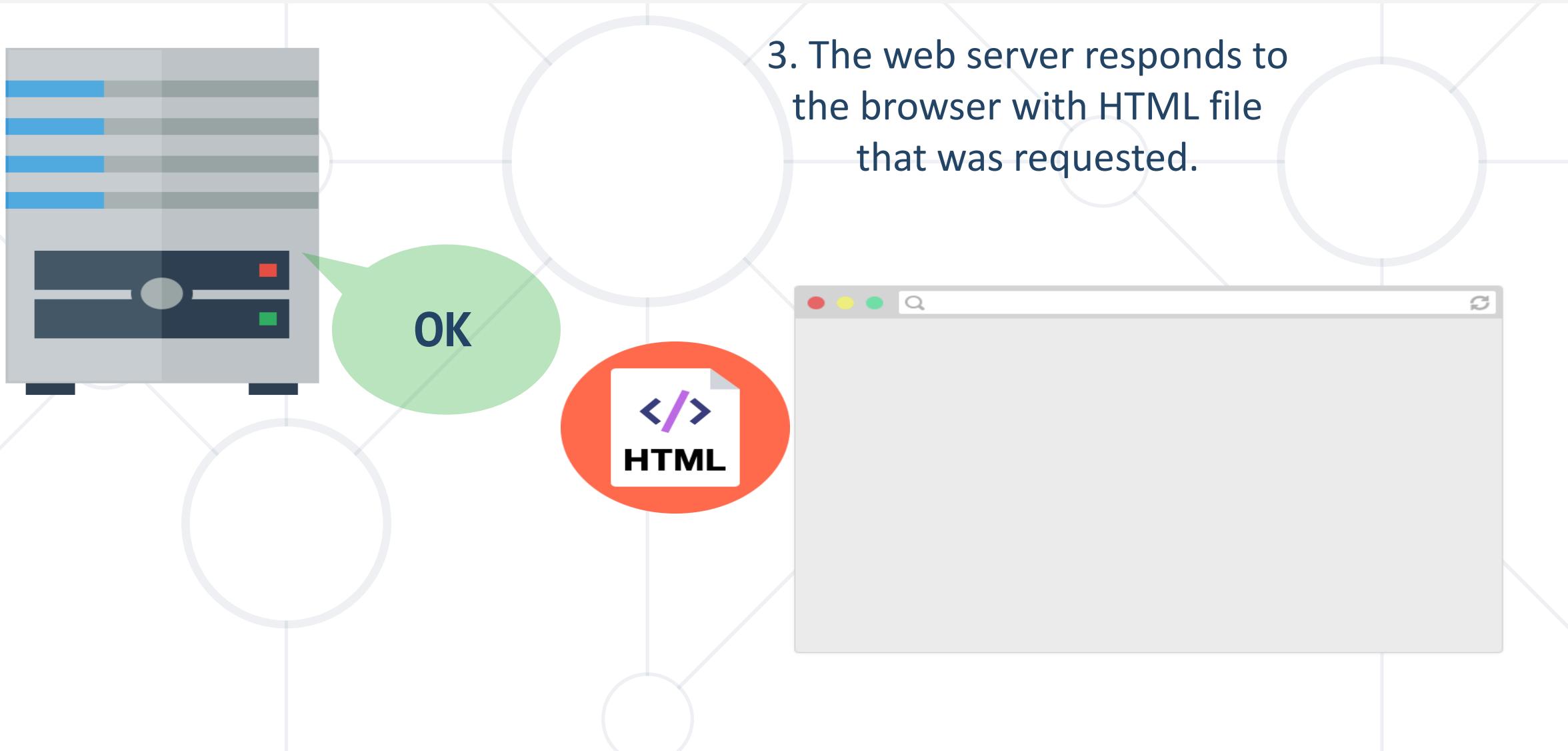


2. The browser asks the web server that hosts the document to send that document.

GET

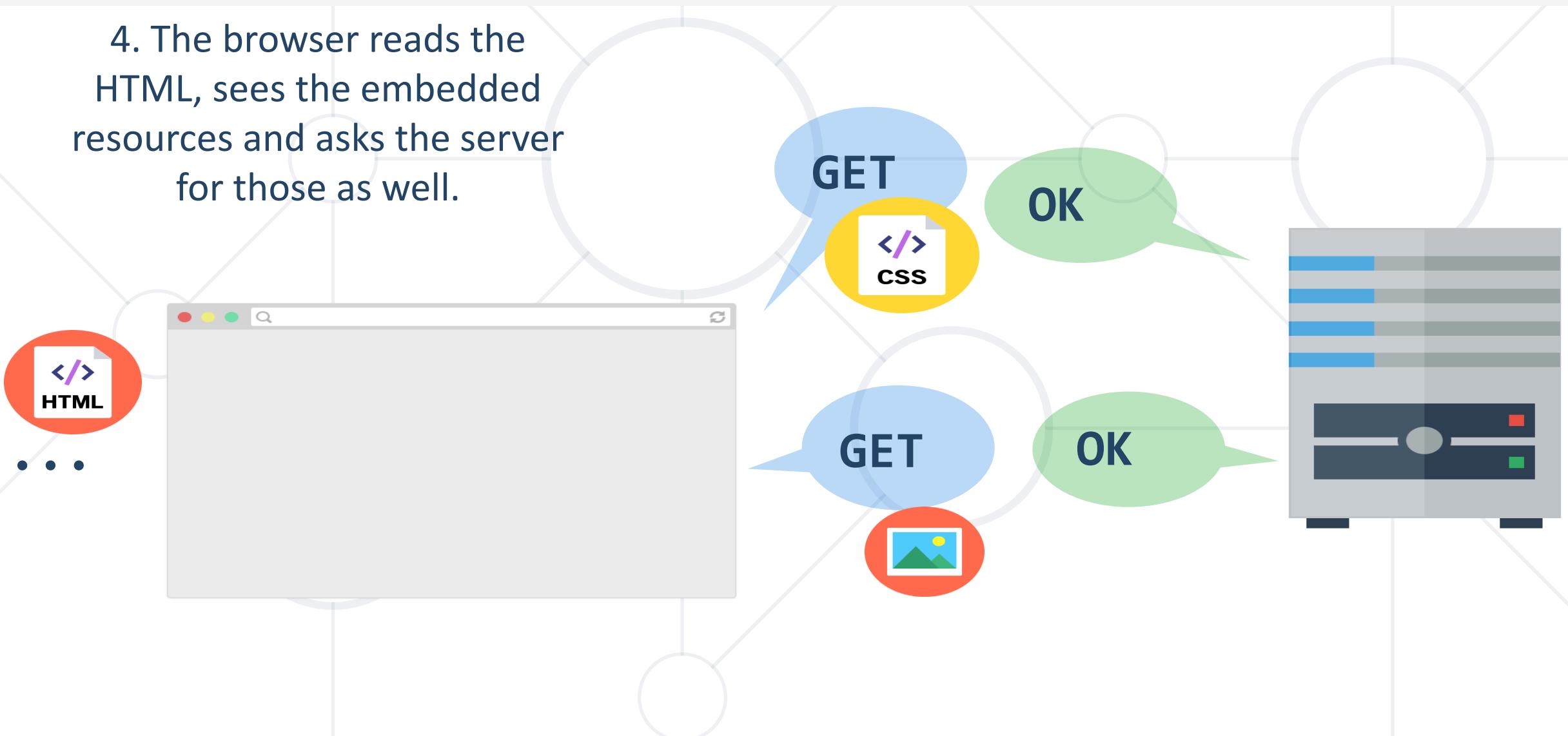


How do web pages work?



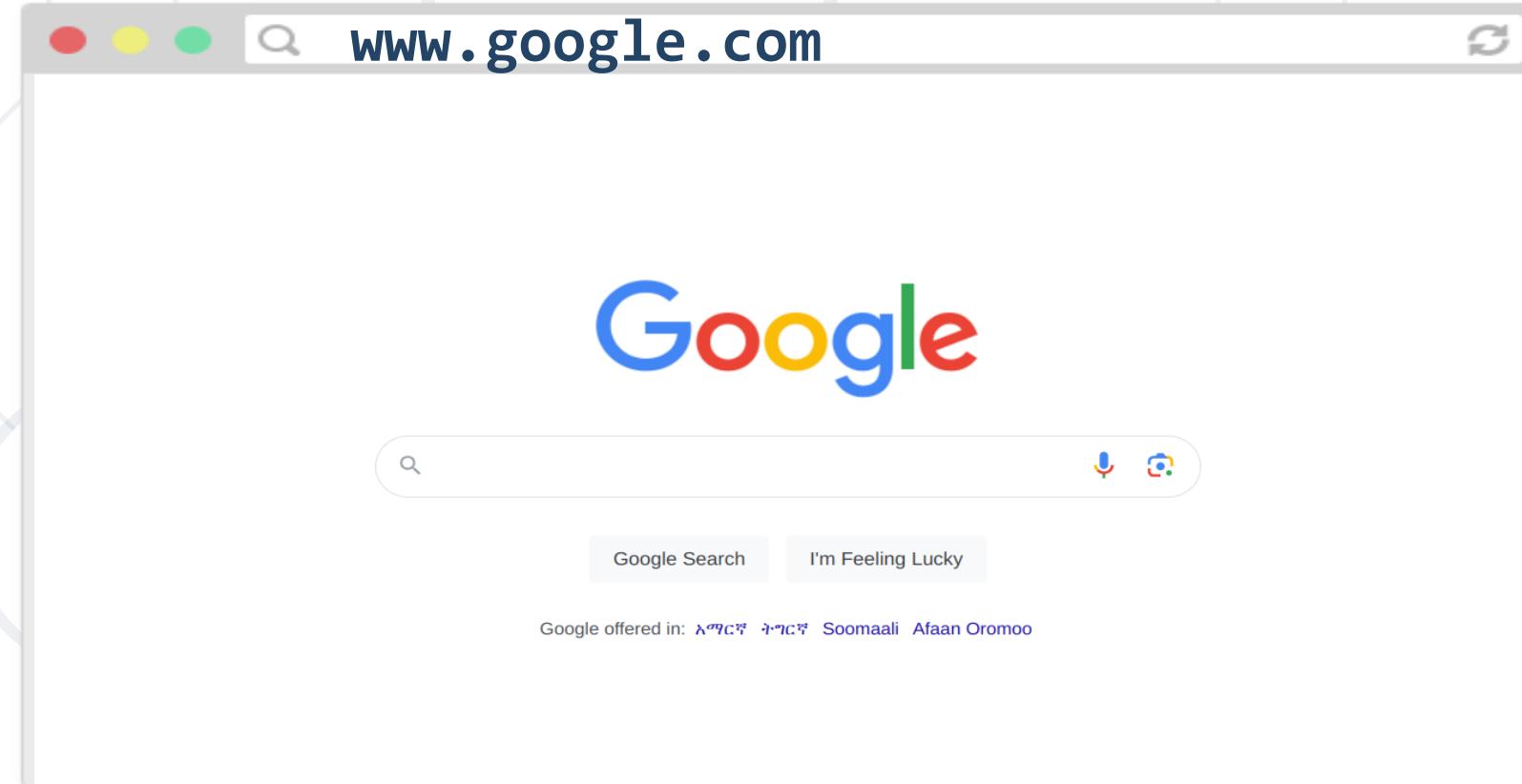
How do web pages work?

4. The browser reads the HTML, sees the embedded resources and asks the server for those as well.



How do web pages work?

5. The web page is loaded when all the resources are fetched and displayed.



Introduction to HTML and CSS

HTML



CSS



Table of Contents

1. HTML Introduction

2. Basic HTML Tags

- Headings, Paragraphs, Images, Lists, Hyperlinks, Divs and Spans, Forms

3. Basic CSS Styling

- Selectors, Rules, Fonts, Colors, Borders, Margins, Paddings, Blocks and Inline Elements





HTML

What is Hypertext Markup Language?

What is HTML?

- HTML – Hypertext Markup Language
 - A text-based notation for describing
 - document structure (semantic markup)
 - document content (text + images + others)
 - formatting (presentation markup)
 - A HTML document consists of many tags (with nesting)
 - A web site consists of many HTML documents (cross-linked)
+ images + CSS styles + scripts + other assets



Basic HTML page structure

(i.e. copy/paste boilerplate)

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS </title>
  </head>

  <body>
    ... contents of the page...
  </body>
</html>
```

Saved in a *filename.html* file.

Basic HTML page structure

(i.e. copy/paste boilerplate)

Metadata that doesn't appear in the viewport of the browser

Contents that render in the viewport of the browser

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS </title>
  </head>
  <body>
    ... contents of the page...
  </body>
</html>
```

E.g. **<title>** shows up as the name of the tab

HTML elements

<**p**>

HTML is <**emem**>

<**img src="puppy.png"** />

</**p**>

- ❖ An element usually has start and ending tags (<**p**> and </**p**>)
 - ❖ **content:** stuff in between start and end tags
- ❖ An element can be self-closing (**img**)
- ❖ An element can have attributes (**src="puppy.jpg"**)
- ❖ Elements can contain other elements (**p** contains **em** and **img**)

HTML Terminology

- **Tags** – keywords describing the document structure
- **Attributes** – properties of the tag, e.g. size, color, etc...
- **Elements** – opening + closing tag + attributes + content

Opening tag

Attribute: **key = "value"**

```
<a href="/home">
```

Navigate to

```
<b>home page</b>
```

```
</a>
```

Element

Closing tag

Some HTML elements

(to place within <body>)

Top-level heading h1, h2, ... h6	<code><h1>Moby Dick</h1></code>
Paragraph	<code><p>Call me Ishmael.</p></code>
Line break	<code>since feeling is first
</code> who pays any attention
Image	<code></code>
Link	<code>click here!</code>
Strong (bold)	<code>Be BOLD</code>
Emphasis (italic)	He 's my <code>brother</code> and all

Some HTML elements

Top-level heading: h1, h2, ... h6

```
<h1>Moby Dick</h1>  
<h2>Or, the Whale</h2>
```

Moby Dick
Or, the Whale

Paragraph: p

```
<p>Call me Ishmael.</p>
```

Call me Ishmael.

Line break: br

```
since feeling is first<br/>  
who pays any attention<br/>  
to the syntax of things
```

since feeling is first
who pays any attention
to the syntax of things

Some HTML elements

Image: `img`

```

```



Link: `a` (note: not `link`)

```
<a href="google.com">click here!</a> click here!
```

Strong (bold): `strong` (note: don't use `b`)

```
<strong>Be BOLD</strong>
```

Be BOLD

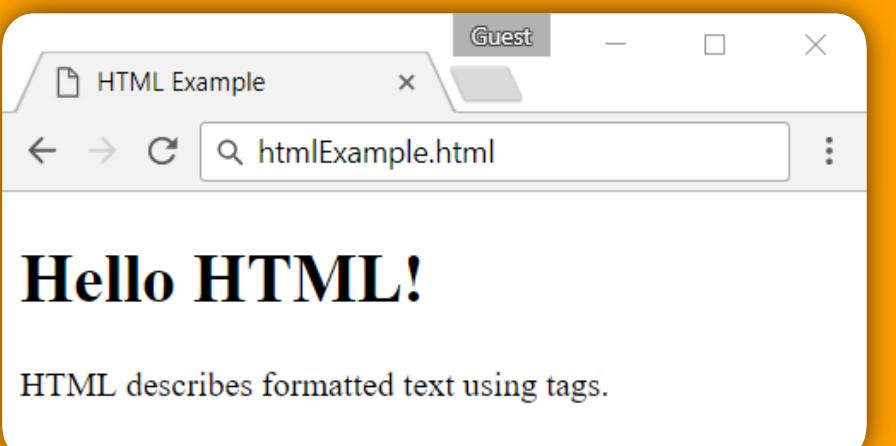
Emphasis (italic): `em` (note: don't use `i`)

He's my `brother` and all

He's my *brother* and all

HTML Page – Example

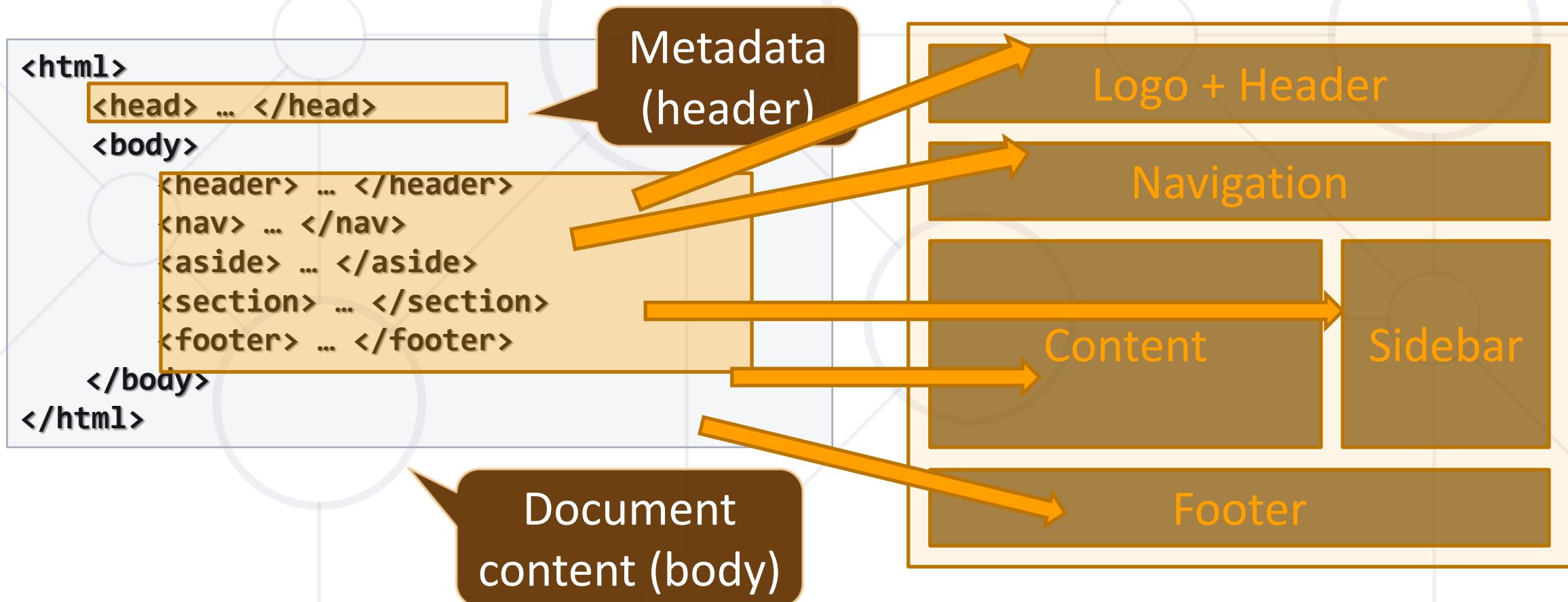
```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Example</title>
  </head>
  <body>
    <h1>Hello HTML!</h1>
    <p>HTML describes formatted text using tags.</p>
  </body>
</html>
```



The screenshot shows a web browser window titled "HTML Example". The address bar contains "htmlExample.html". The main content area displays the text "Hello HTML!" in a large, bold, black font. Below it, a smaller paragraph reads "HTML describes formatted text using tags.".

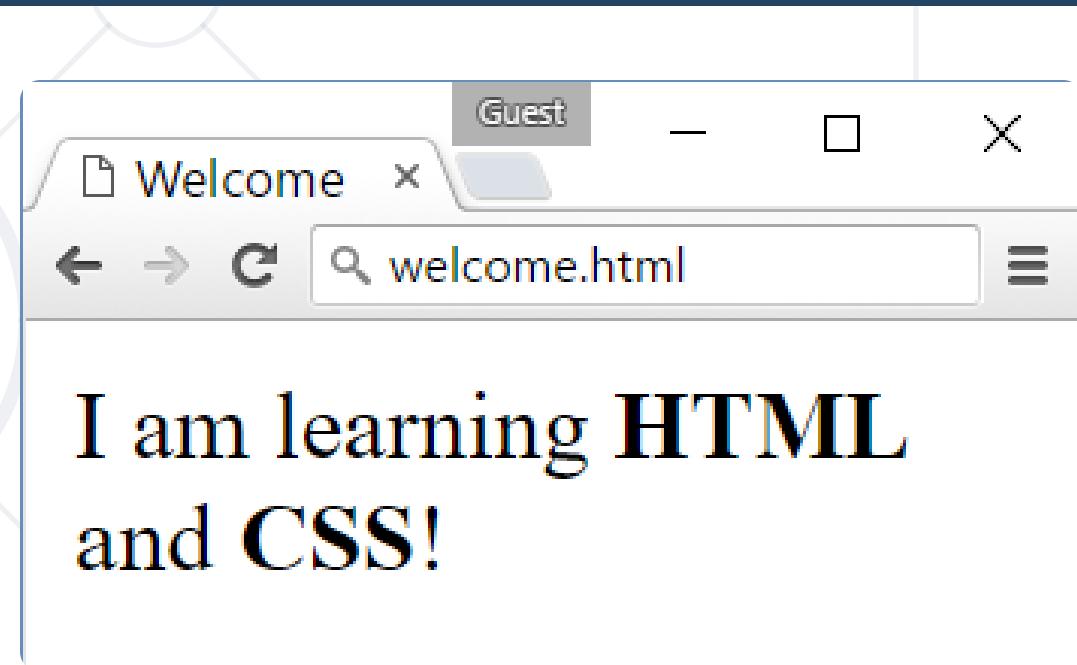
Document Structure

- HTML 5 defines semantic tags for layout
 - **<header>, <footer>, <nav>, <aside>, <section>**



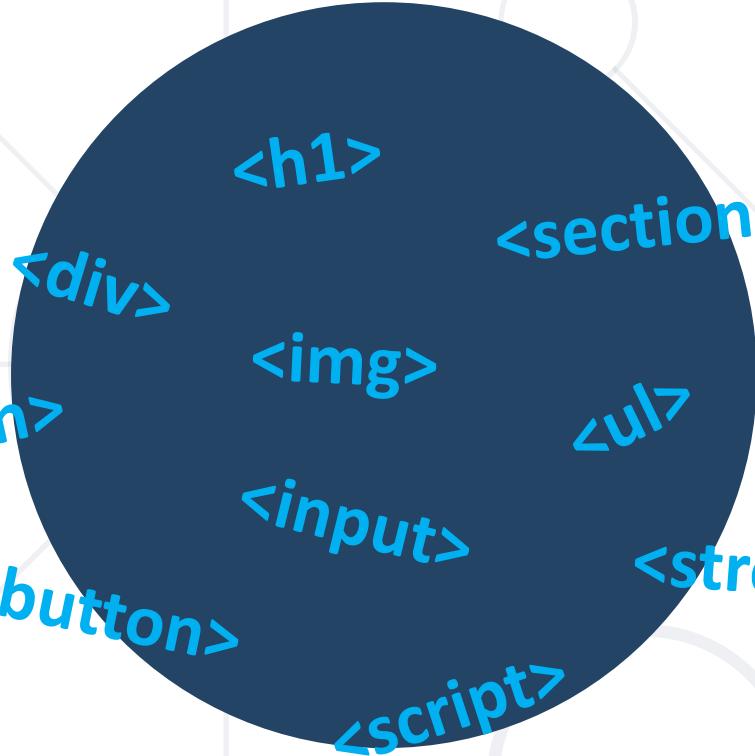
Exercise: Welcome to HTML

- Create your first Web page:
 - File name: **welcome.html**
 - Title: Welcome
 - Paragraph of text:
I am learning HTML and CSS!
- Hint:
 - You can use as a basis the html structure from the previous slides
 - For bolder text use the **** tag



HTML Common Tags

Widely Used on All Websites



<h1>
<section>

<input>

<script>
<button>

<div>

Headings

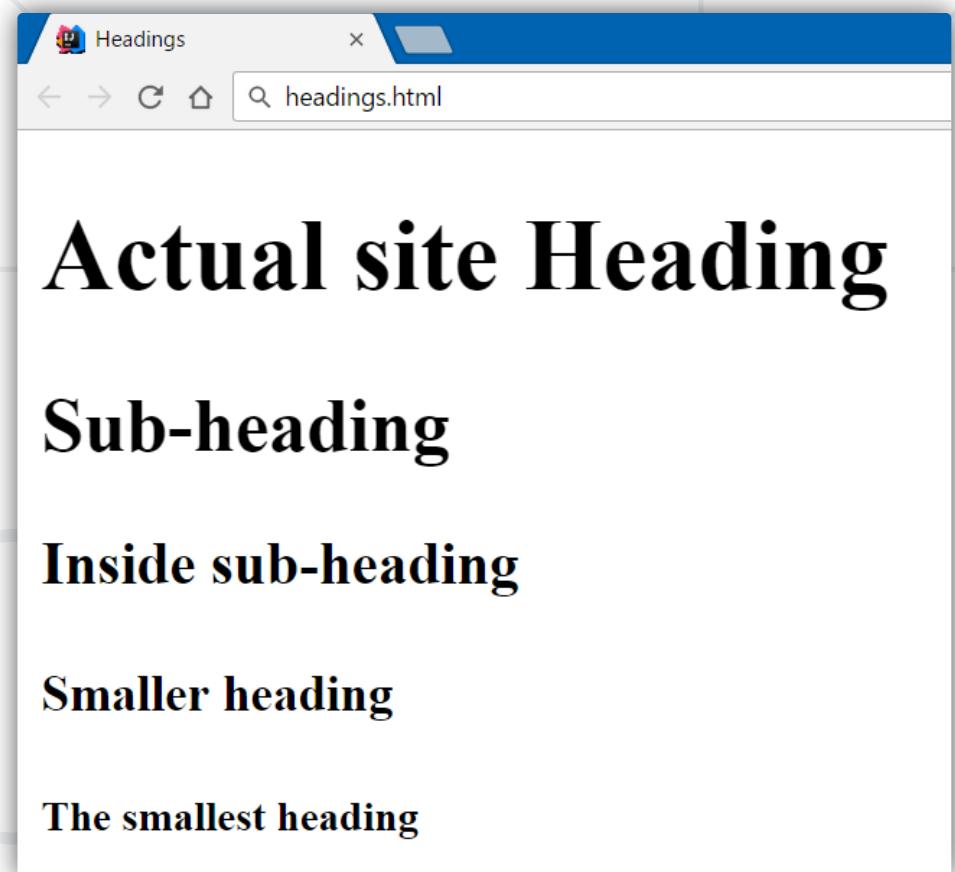
- Headings help with page structure, just like in Microsoft Word
- HTML has six different headings
 - **<h1>** defines the most important heading
 - **<h6>** defines the least important heading
- Example:

```
<h1>First Heading (Biggest)</h1>
<h2>Second Heading (Smaller)</h2>
<h3>Third Heading (Even Smaller)</h3>
<h4>Fourth Heading (Smallest)</h4>
```

First Heading (Biggest)
Second Heading (Smaller)
Third Heading (Even Smaller)
Fourth Heading (Smallest)

Problem: Headings

- Create a Web page with five headings:
 - Heading for the whole page `<h1>`
 - Four smaller headings `<h2>` to `<h5>`
- Hint:
 - Use `<h1>` to `<h5>` tags



Solution: Headings

```
<html>
  <head>
    <title>Headings</title>
  </head>
  <body>
    <h1>Actual site Heading</h1>
    <h2>Sub-heading</h2>
    <h3>Inside sub-heading</h3>
    <h4>Smaller heading</h4>
    <h5>The smallest heading</h5>
  </body>
</html>
```

Set the page title

Page content

Paragraphs

- The **<p>** tag defines a paragraph
- The **
** tag defines a line break
- Example:

```
<p>First paragraph</p>
<p>Second paragraph</p>
<br> <!-- empty Line -->
<p>Third paragraph</p>
```

Comment

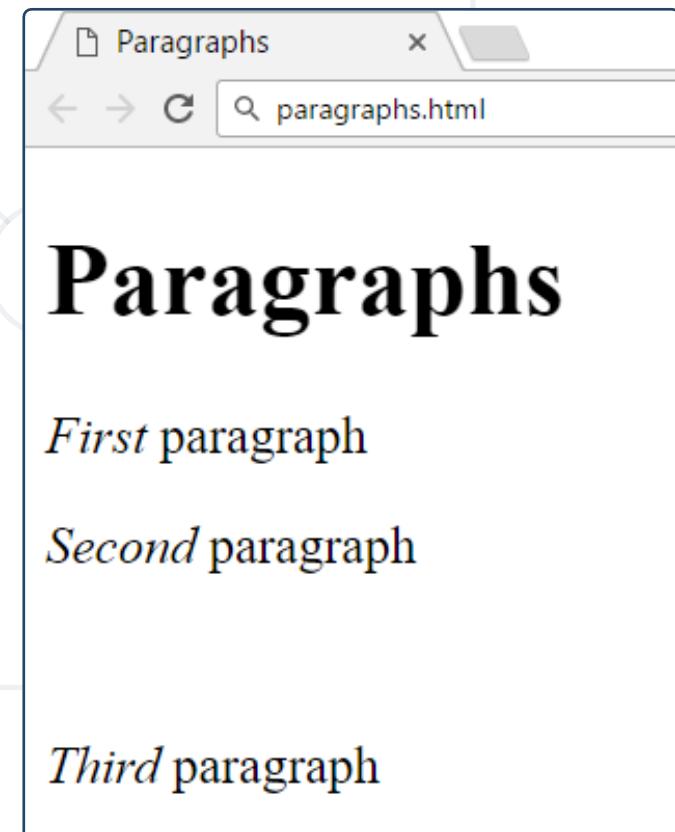
First paragraph

Second paragraph

Third paragraph

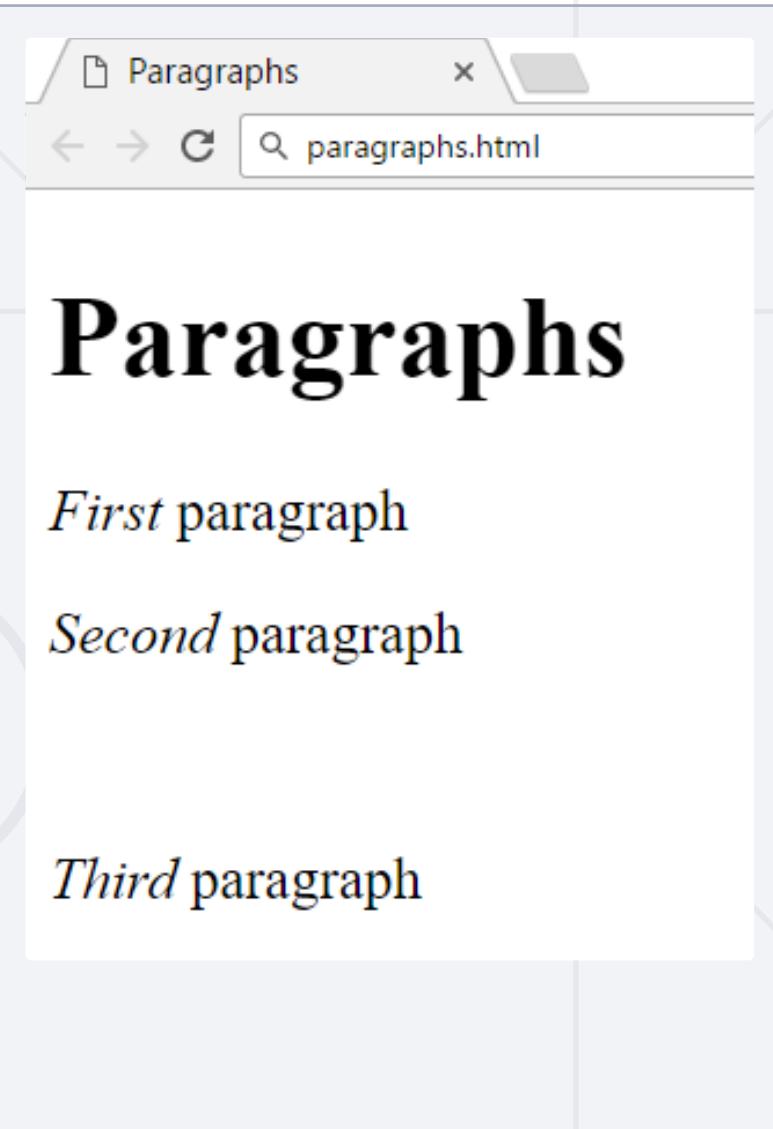
Exercise: Paragraphs

- Create a Web page holding three paragraphs and a blank line:
 - Heading for the whole page **<h1>**
 - Three paragraphs **<p>**
 - Blank line **
**
- Hint:
 - Use **** tag for emphasizing text (*italic font*)



Solution: Paragraphs

```
<html>
  <head>
    <title>Paragraphs</title>
  </head>
  <body>
    <h1>Paragraphs</h1>
    <p><em>First</em> paragraph</p>
    // TODO
    <br>
    <p><em>Third</em> paragraph</p>
  </body>
</html>
```



Bullets and Numbered Lists

```
<ul>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ul>
```

- First item
- Second item
- Third item

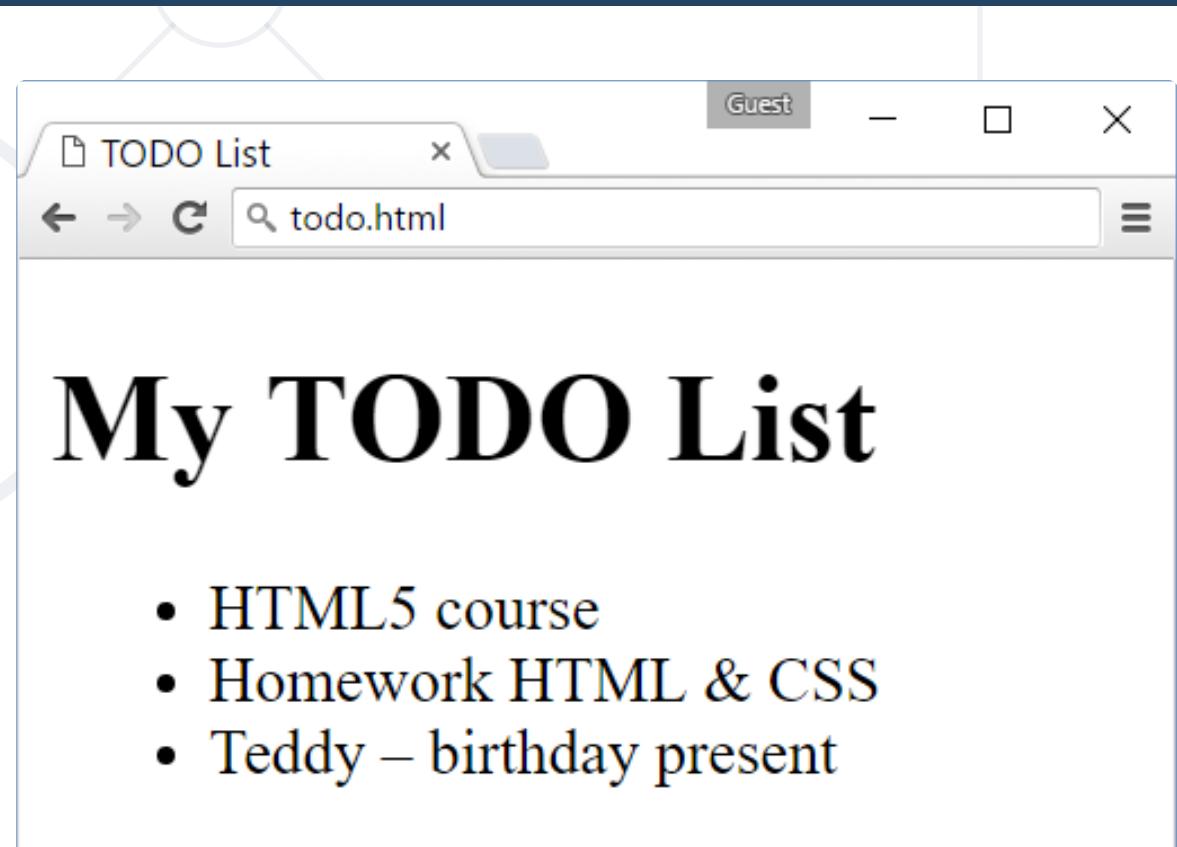
```
<ol>
  <li>One</li>
  <li>Two</li>
  <li>Three</li>
</ol>
```

1. One
2. Two
3. Three

Exercise: My TODO List

- Create a Web page:

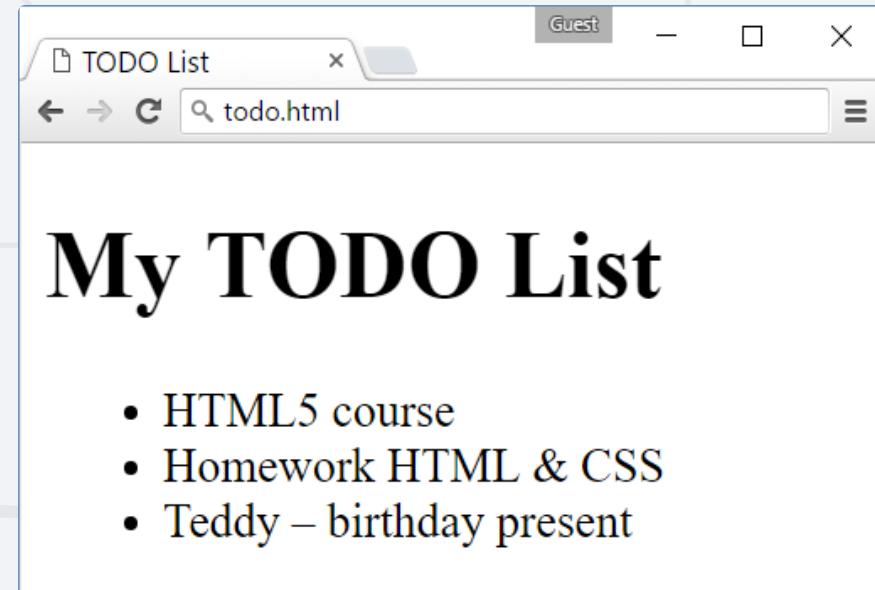
- File name: **todo.html**
- Title: **TODO List**
- Large heading: **My TODO List**
- List of items:
 - HTML5 course @ SoftUni
 - Homework HTML & CSS
 - Teddy – birthday present



Hint: use – to display the long hyphen –

Solution: My TODO List

```
<html>
  <head>
    <title>TODO List</title>
  </head>
  <body>
    <h1>My TODO List</h1>
    <ul>
      <li>HTML5 course </li>
      <li>Homework HTML & CSS</li>
      <li>Teddy – Birthday present</li>
    </ul>
  </body>
</html>
```



Hyperlinks

- Created by using the `<a>` tag

```
<a></a>
```

- The actual address is specified in the `href=""` attribute

```
href="https://softuni.bg"
```

- External hyperlink

```
<a href="https://www.google.com">Google</a>
```

Exercise: Hello HTML

- Create a Web page:

- File name: **hello.html**
- Title: Hello HTML
- Large heading: Hello HTML!

- Paragraph of text:

I am ***<your name (bold)>***. I am from *<your town as link to your town's Web site>*.

- Paragraph of text:

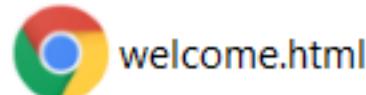
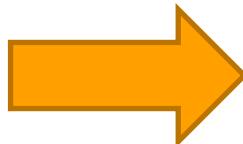
I study *<specialty (italic)>* at *<link to SoftUni>*.



Hyperlinks – Local and Internal

- Local hyperlink – link to the same web site

```
<a href="welcome.html">Review "welcome.html"</a>
```



- Internal hyperlink (to the same page)

```
<h1 id="top">Heading</h1>
```

```
... <!-- some Long text -->
```

```
Go to <a href="#top" target="_self">top</a>
```

Heading

"Lorem ipsum dolor sit amet,

...

deserunt mollit anim id es. laborum."

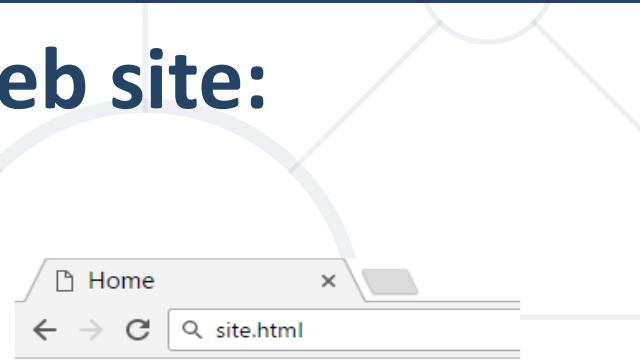
Go to [top](#)



Exercise: Website

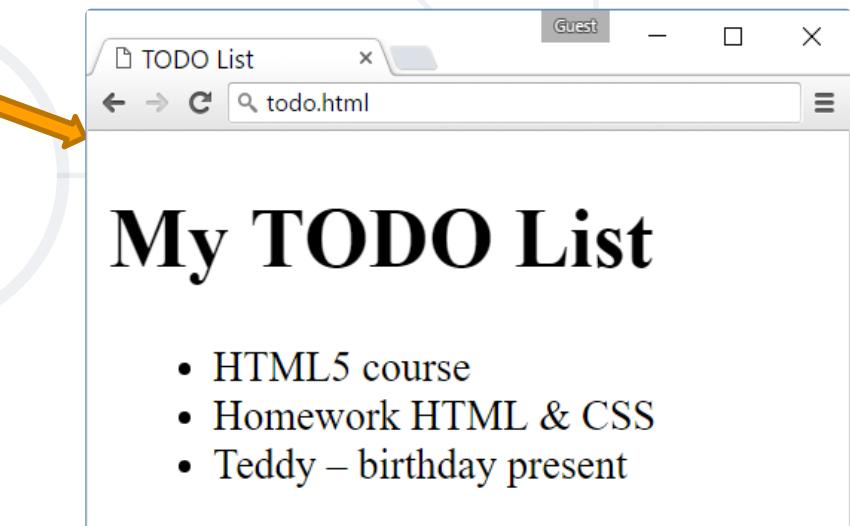
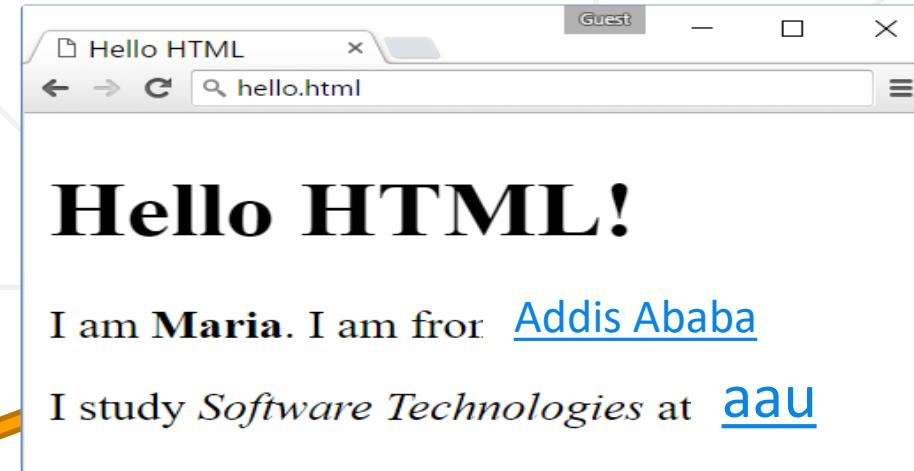
- Create a three page Web site:

- Create page:
 - **home.html**
- Link it with both:
 - **hello.html**
 - **todo.html**
- In both files create:
 - **link "back to home"**



Home

- [hello.html](#)
- [todo.html](#)



Images

- Images are inserted using the **** tag

```

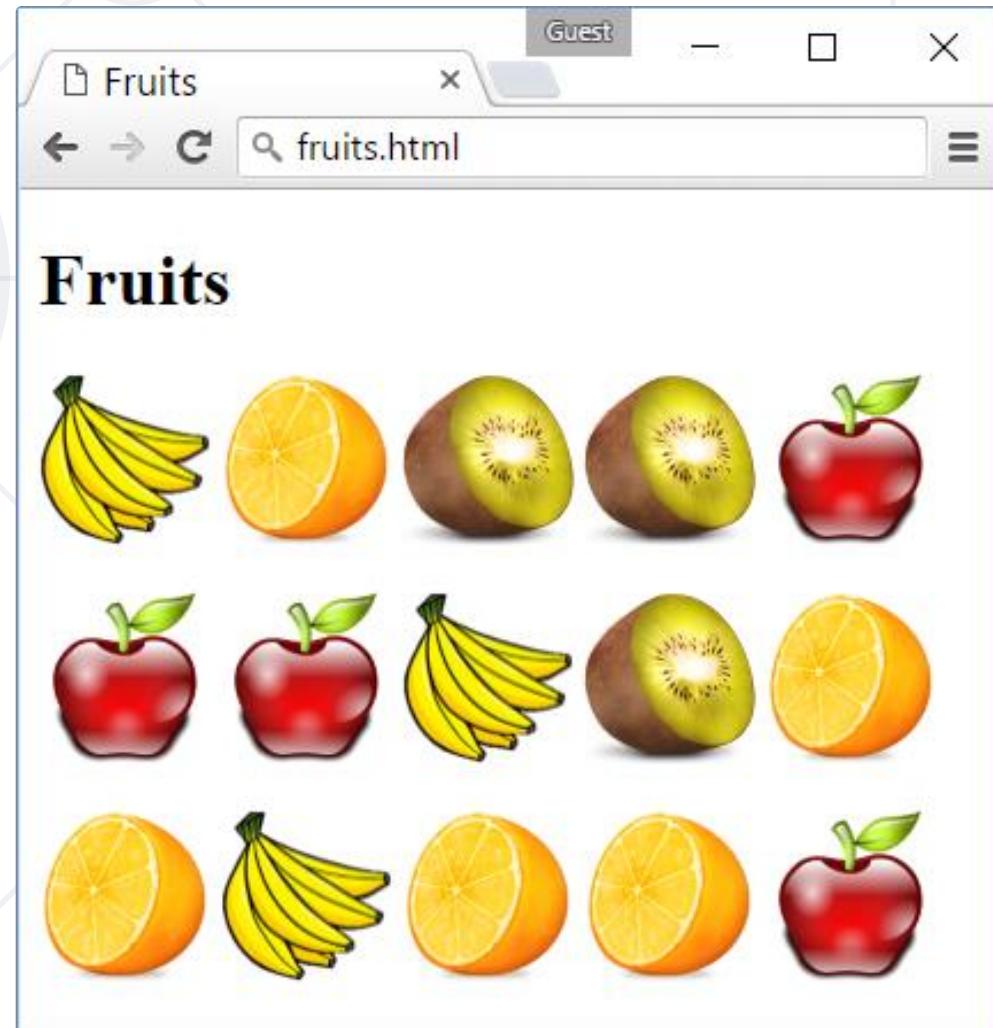
```



- **** tag attributes:
 - **src="..."** – sets the location of the image
 - **alt="..."** – sets an alternative text
 - **width / height="..."** – sets dimensions

Exercise: Fruits

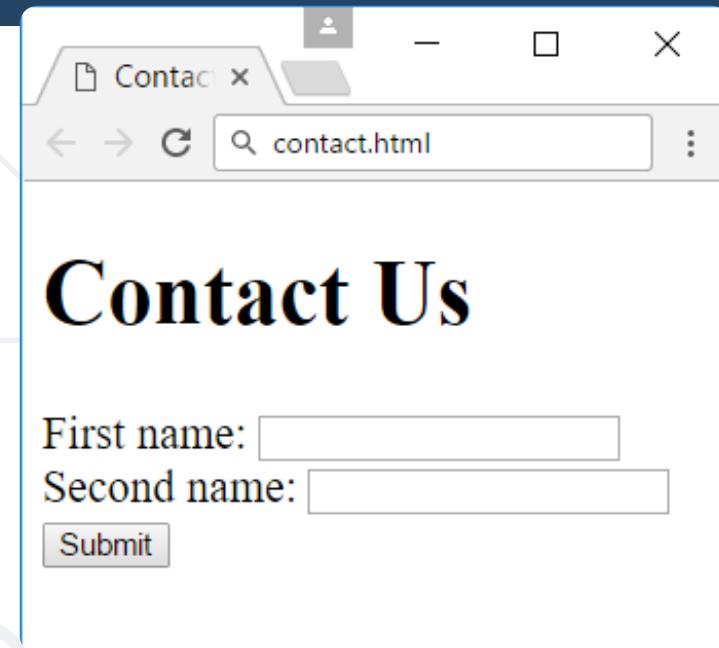
- You are given 4 image files:
 - **apple.png**
 - **banana.png**
 - **kiwi.png**
 - **orange.png**
- Create a Web page like the screenshot on the right
 - Hint: use 3 paragraphs, each holding 5 images



HTML Forms

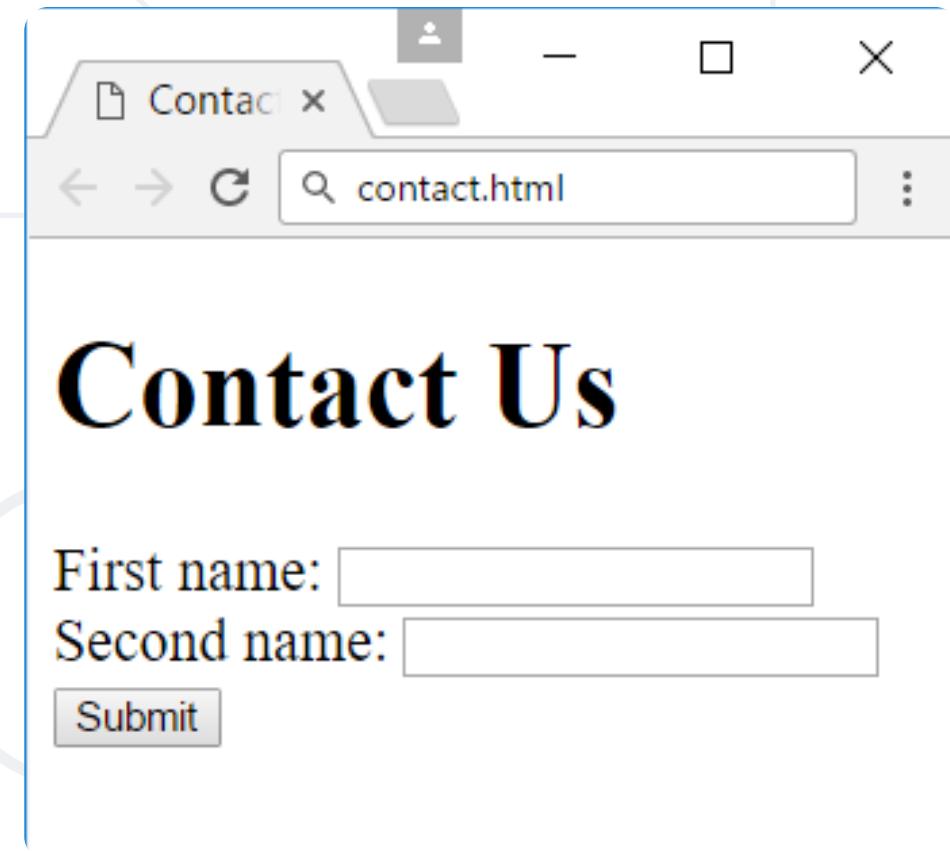
- HTML forms allow user to fill data and send it to the server
 - Input fields can hold text, number, date, radio button, checkbox, ...
- Creating a contact form:

```
<form>
  First name: <input type="text" name="firstname"><br>
  Last name: <input type="text" name="lastname"><br>
  <input type="submit" value="Submit">
</form>
```



Exercise: Contact Form

- Create a form:
 - First Name – text box
 - Last Name – text box
 - [Submit] button
- Create a Web page like the screenshot on the right
 - Hint: use `<input type="text"/>` and `<input type="submit"/>`



Home work: Basic HTML Tags

- Implement all the basic HTML tags we covered sofar

```
.tagcloud {  
  @include flexbox();  
  @include flex-wrap(wrap);  
  @include align-items(center);  
  
  a {  
    display: inline-block;  
    margin: 0.125rem;  
    padding: 0.4375rem;  
    background: rgba($link-color, 0.2);  
    border: 1px solid rgba($link-color, 0.2);  
    transition: all 0.1s ease-in-out;  
  
    &:hover,  
    &:focus {  
      background: $link-color;  
      color: $white;  
      transform: scale(1.1);  
    }  
  }  
}
```

```
/*Dialog*/  
.dialog, .dialog__overlay {  
  width: 100%;  
  height: 100%;  
  top: 0;  
  left: 0;  
}  
  
&__content {  
  width: 50%;  
  max-width: 750px;  
  min-width: 550px;  
  background: #fff;  
  @include breakpoint(medium)  
    max-width: 90%;  
    min-width: 50%;  
  width: 90%;  
}
```



CSS

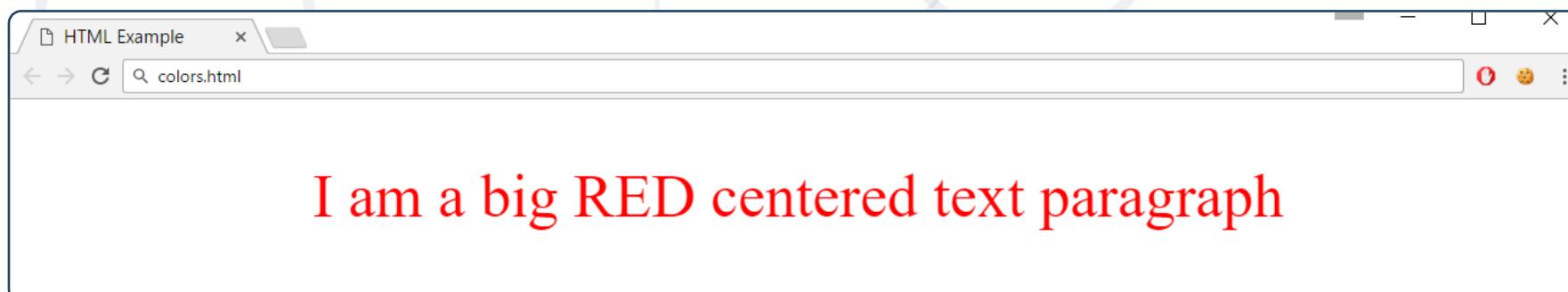
Cascading Style Sheets

Separating Content from Presentation

What is CSS?

- CSS defines styling of the HTML elements
 - CSS specifies fonts, colors, margins, sizes, positioning, floating, ...
 - Uses CSS declarations in format: **property:value**
- Inline CSS defines formatting rules for a certain HTML element:

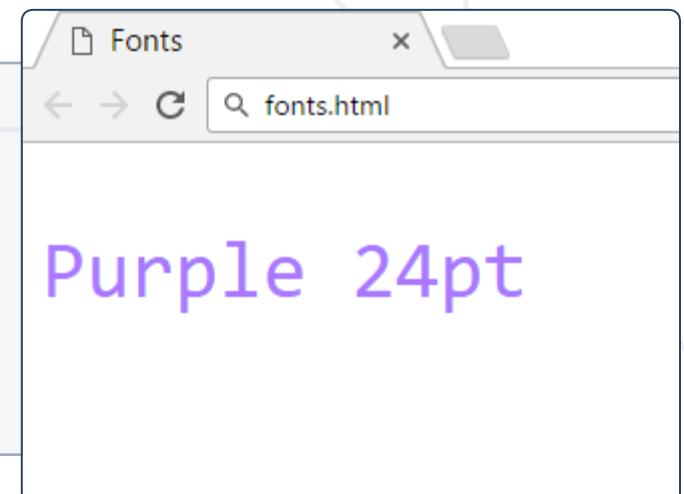
```
<p style="color: red; text-align: center; font-size: 30pt">I am a big RED centered text paragraph</p>
```



Fonts – Font Family, Size and Colors

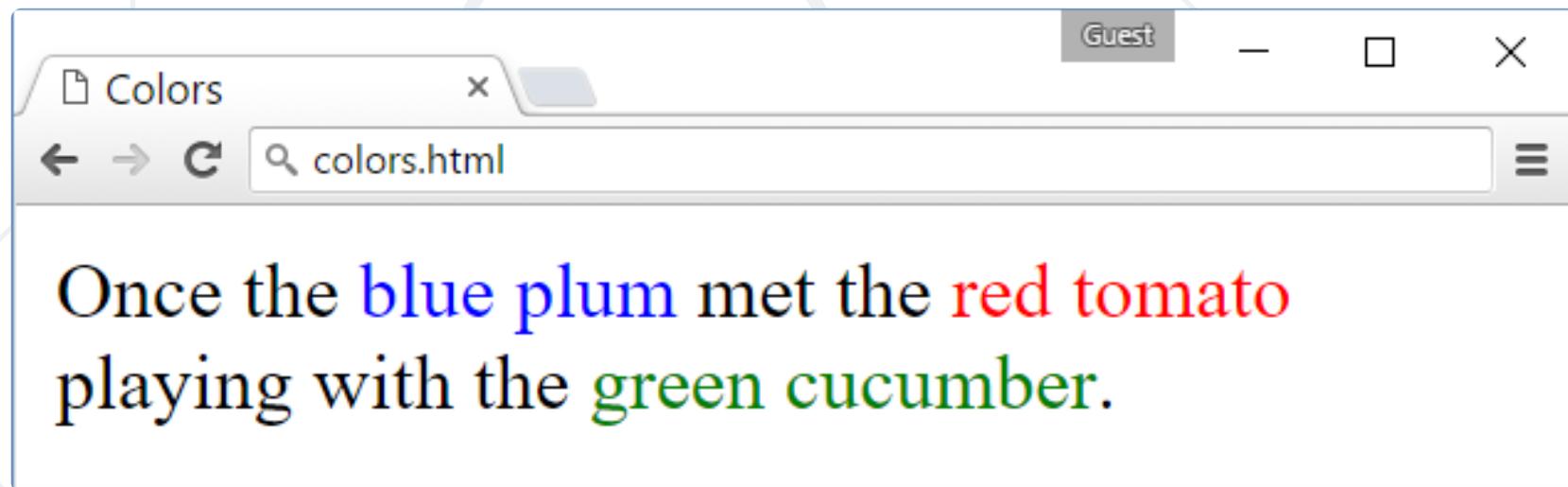
- **color**: specifies the color of the letters
- **font-family**: should hold several fonts. If the browser does not support the first one, it tries the next, and so on
- **font-size**: sets the size

```
<p style="color: #AA77FF;  
font-family: Consolas, monospace;  
font-size: 24pt; Example</p>
```



Exercise: Colors

- Create a Web page like the screenshot below:

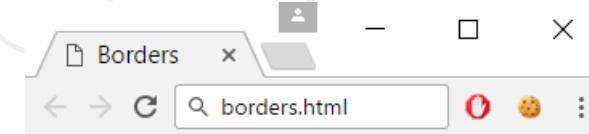


- Hints:
 - Use a paragraph of text
 - Use `text` for the colored text

Borders, Backgrounds

- **border**: specifies the type, thickness, color
- **border-radius**: rounds border edges
- **background**: sets the background

```
<p style=" border: 2px solid red;  
border-radius: 10px;  
background: lightgray;">red border</p>
```



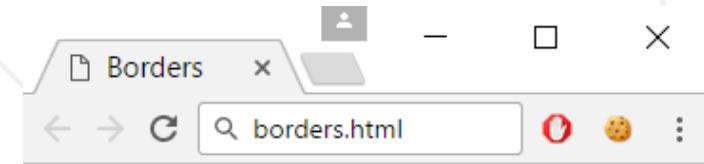
Red Border

Exercise: Colors

- Create a Web page like the screenshot:

- Title: Borders
 - Content:
 - Red bordered paragraph
 - Green bordered paragraph
 - Blue bordered paragraph

- Hints:
 - Use a paragraphs of text
 - Use `text` for the colored text



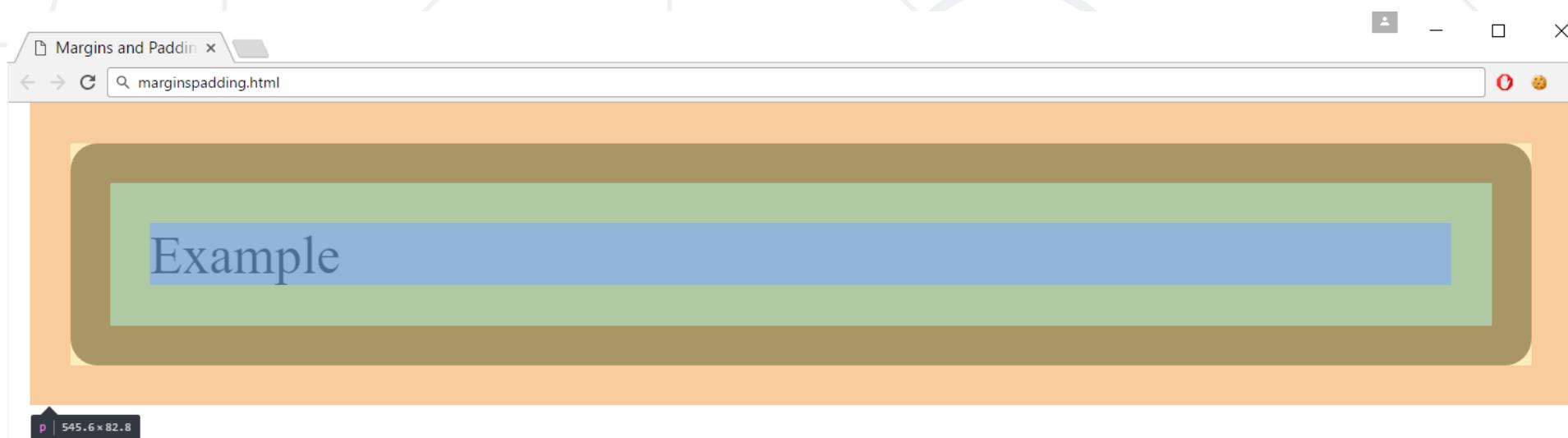
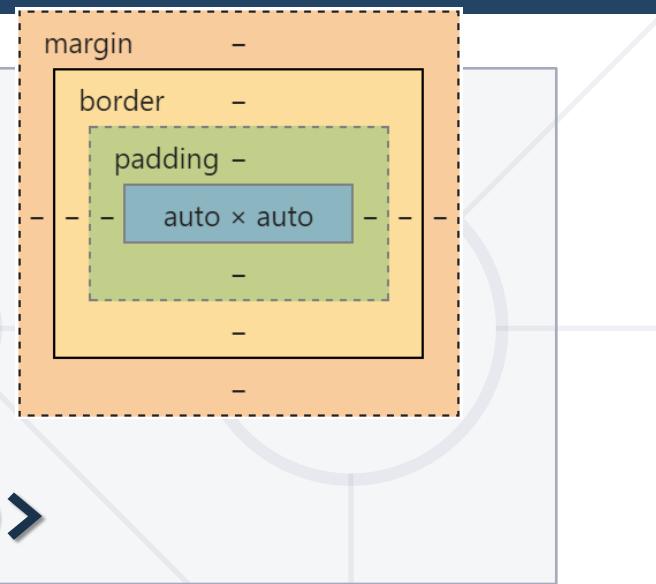
Red Border

Green Border

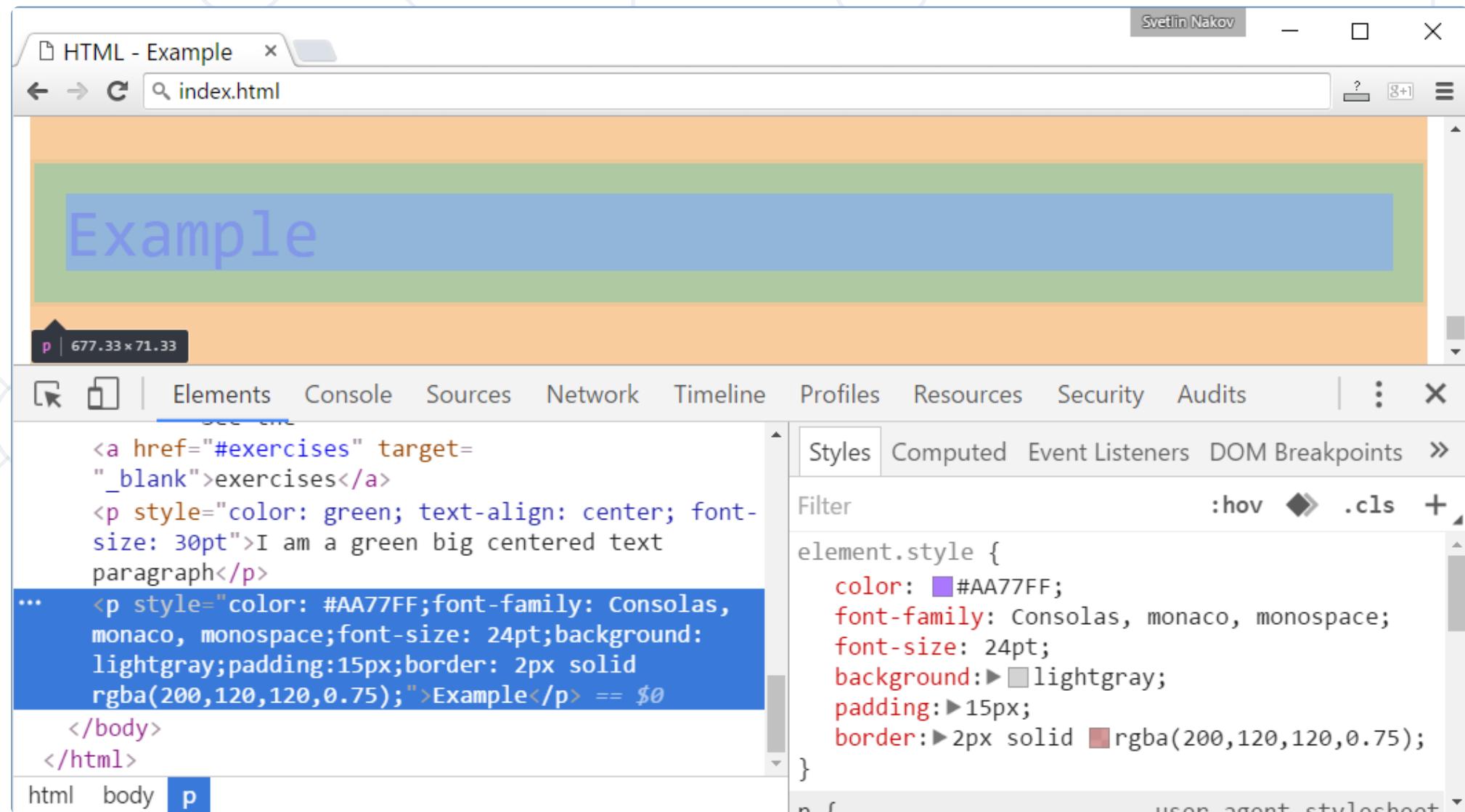
Blue Border

Margins, Padding

```
<p style="margin: 15px;  
padding: 15px;  
border-radius: 10px;  
border: 15px solid black">Example</p>
```

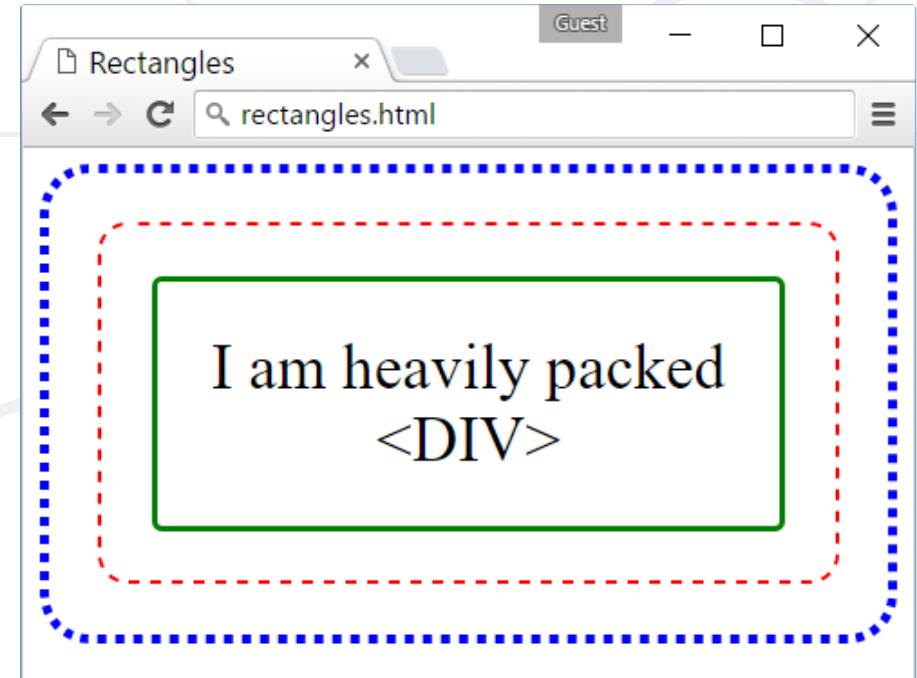


The Dev Tools / Styles Inspector / [F12]



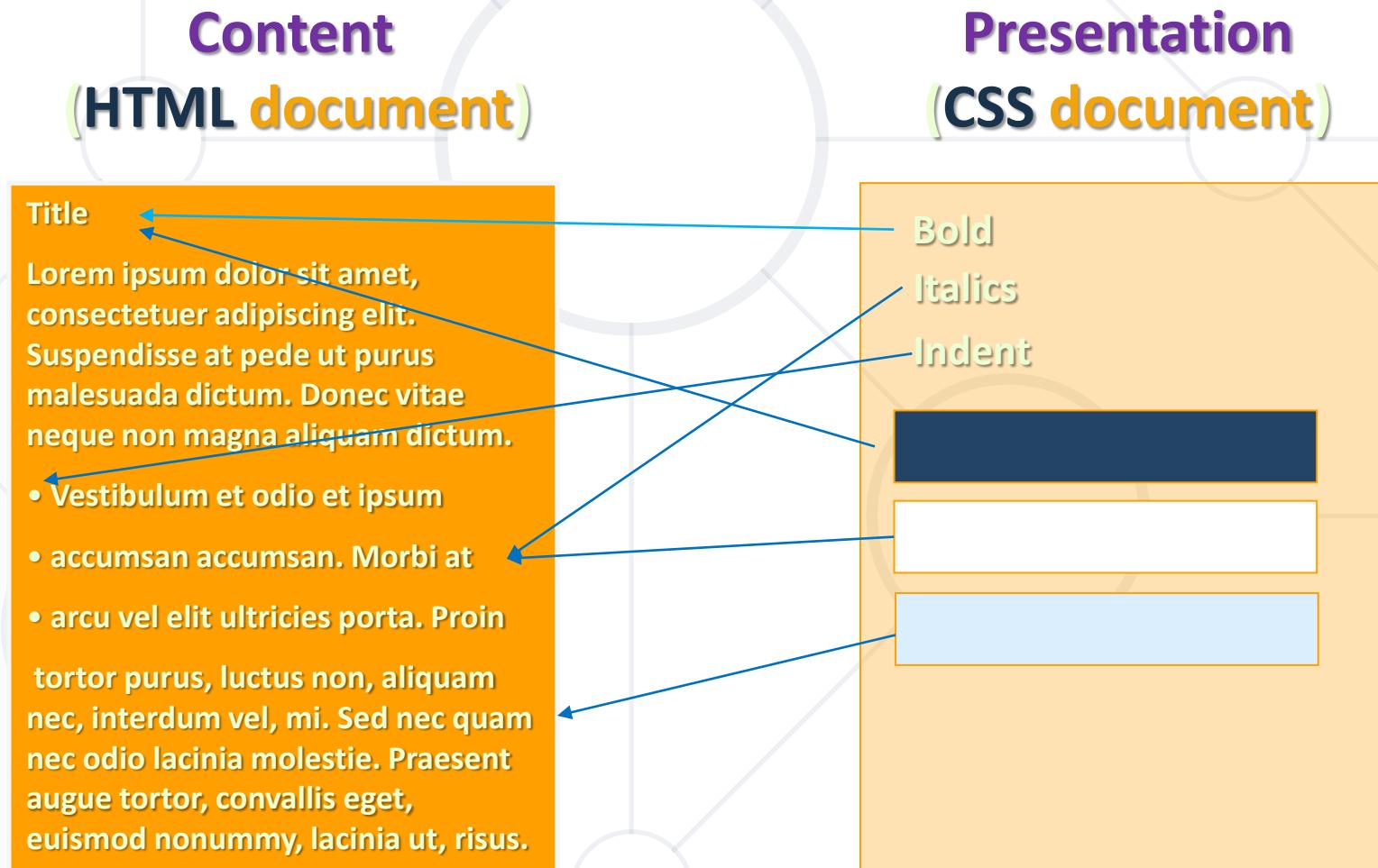
Exercise: Rectangles

- Create a Web page like at the screenshot. Hints:
 - Use 3 nested `<div>` elements
 - Outside div: blue dotted border + `border-radius` + padding
 - Middle div: red dashed border + `border-radius` + padding
 - Inner div: green solid border + `border-radius` + padding + `text-align` + `font-size`
 - Use `<` and `>` to escape the `<` and `>` characters in the text



CSS Philosophy: Content/Presentation

- Separate content from presentation!



CSS Philosophy: Result

Title

**Lorem ipsum dolor sit amet,
 consectetuer adipiscing elit.
 Suspendisse at pede ut purus
 malesuada dictum. Donec vitae neque
 non magna aliquam dictum.**

- *Vestibulum et odio et ipsum*
- *accumsan accumsan. Morbi at*
- *arcu vel elit ultricies porta. Proin*

**Tortor purus, luctus non, aliquam nec,
interdum vel, mi. Sed nec quam nec
odio lacinia molestie. Praesent augue
tortor, convallis eget, euismod
nonummy, lacinia ut, risus.**

Combining HTML and CSS Files

using-css.html

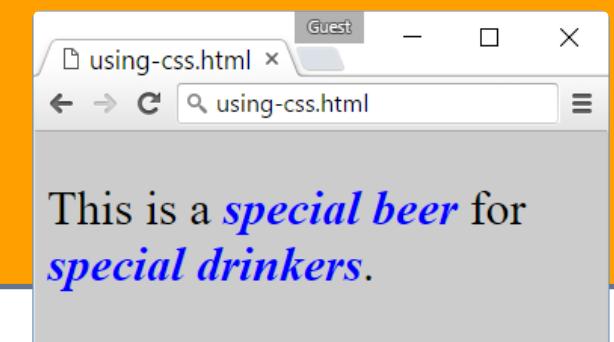
```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css"
          href="styles.css">
  </head>
  <body id="content"><p>
    This is a <span class="special"> special
    beer</span> for <span class= "special">special
    drinkers</span>.</p>
  </body>
</html>
```

styles.css

```
.special {
  font-style: italic;
  font-weight: bold;
  color: blue;
}

#content {
  background: #EEE;
}

p {
  font-size: 24pt;
}
```



CSS Selectors

- **.class** – selects a group of elements with the specified class
- **#id** – selects a unique element by ID
- **tag** – selects all specified tags
- ***** - selects everything

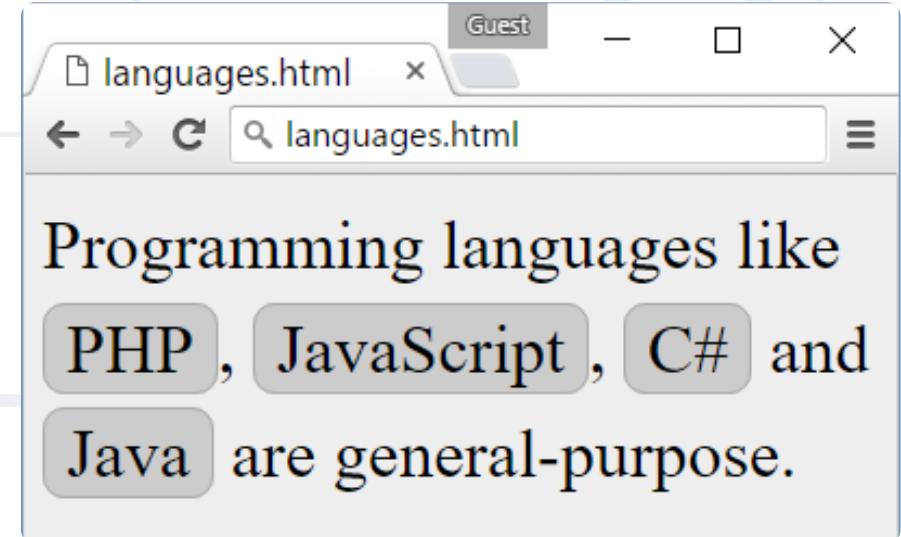


styles.css

```
.special {  
    font-style: italic;  
    font-weight: bold;  
    color: blue;  
}  
  
#content {  
    background: #EEE;  
}  
  
p {  
    font-size: 24pt;  
}  
  
* {  
    border: 0;  
}
```

Exercise – Languages

- Create a Web page like at the screenshot. Hints:
 - Create HTML file **langs.html**
 - Create CSS file **langs.css**
 - Link the CSS file in the HTML header
`<link rel="..." href="langs.css">`
 - In the **body** CSS selector define:
 - `background:#EEE; line-height:1.5em; font-size:24pt;`
 - Define and use a CSS class **.lang** for styling the languages:
 - Specify `border:1px #AAA, border-radius, background:#CCC, padding`



Solution – Languages (HTML + CSS)

languages.html

```
<!DOCTYPE html>
<html>
  <head> <title>...</title>
    <link rel="stylesheet" type=
      "text/css" href="languages.css">
  </head>
  <body>
    Programming ... <span class="lang">
      PHP</span>, <span class="lang">
      JavaScript</span>, ... purpose.
  </body>
</html>
```

languages.css

```
body {
  font-size: 24pt;
  background: #EEE;
  line-height: 1.5em;
}

.lang {
  padding: 2px 10px;
  border: 1px solid #AAA;
  background: #CCC;
  border-radius: 10px;
}
```

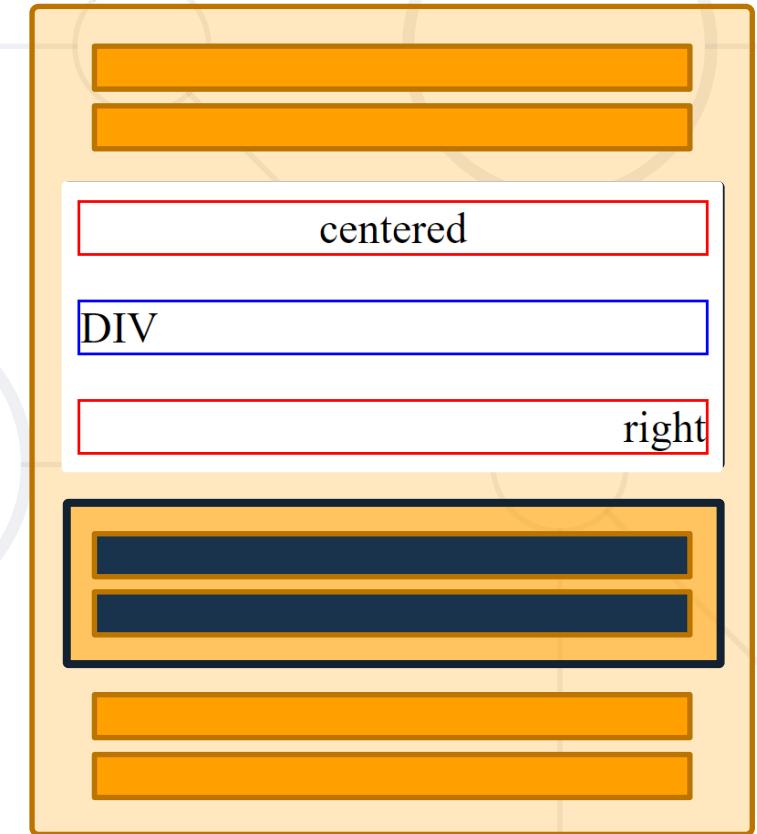
Block Elements

- **<div>** and **<p>** are **block elements** (rectangles)

- Fill the entire container width
- Stack vertically one after another

```
<p style="border:1px solid red;  
text-align:center">centered</p>  
<div style="border:1px solid  
blue">DIV</div>  
<p style="border:1px solid red;  
text-align:right">right</p>
```

display: block

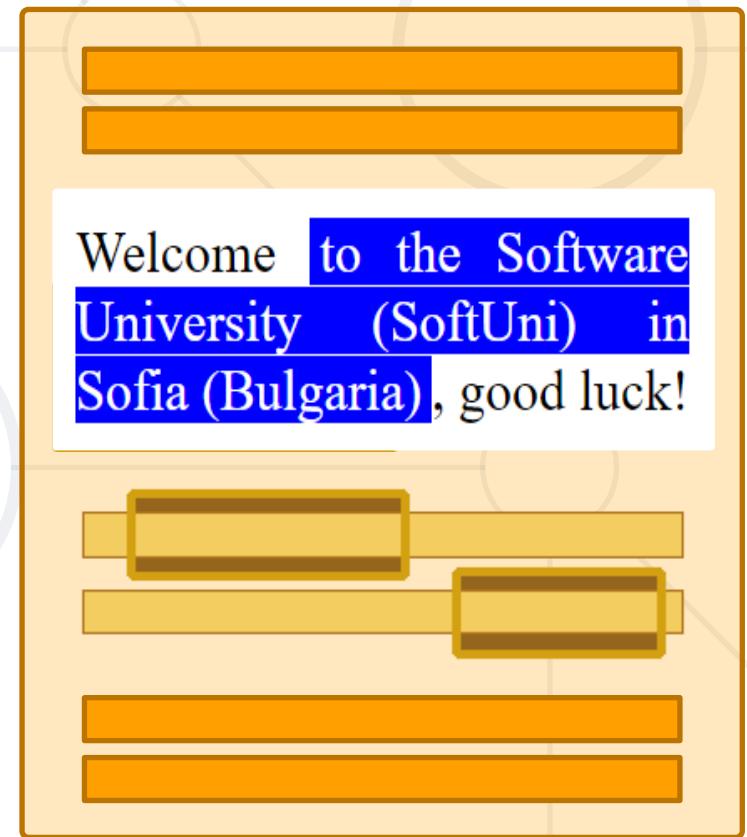


Inline Elements

- **** is inline element
 - Its shape is not always rectangular
 - Can be split across multiple lines

```
<p style="text-align:justify">  
Welcome <span style="color:white;  
background:blue; padding-right:3px;  
padding-left:3px;">to the Software  
University (SoftUni) in Sofia  
(Bulgaria)</span>, good luck!</p>
```

display: inline



Inline-Block Elements

- Elements can be also **inline-block**
 - Rectangles arranged one after another
 - Just like words in a sentence

```
<div style="text-align:justify;">  
  <div style="display:inline-block;  
background:green">green</div>  
  <div style="display:inline-block;  
background:red">red block</div>  
  ...  
</div>
```

display: inline-block



Exercise – Contact Us Form (HTML)

- Create a Web page (HTML + CSS) like at the screenshot:

```
<form>
  <span>First name:</span>
  <input type="text" name="firstname"><br>
...
  <span>Town:</span>
  <select name="town">
    <option value="1">Sofia</option>
...
  </select>
  <input type="submit" value="Submit">
</form>
```

The screenshot shows a web browser window titled "Contact Form" with the URL "form-contact-us.html". The page has a title "Contact Us" and contains a form with the following fields:

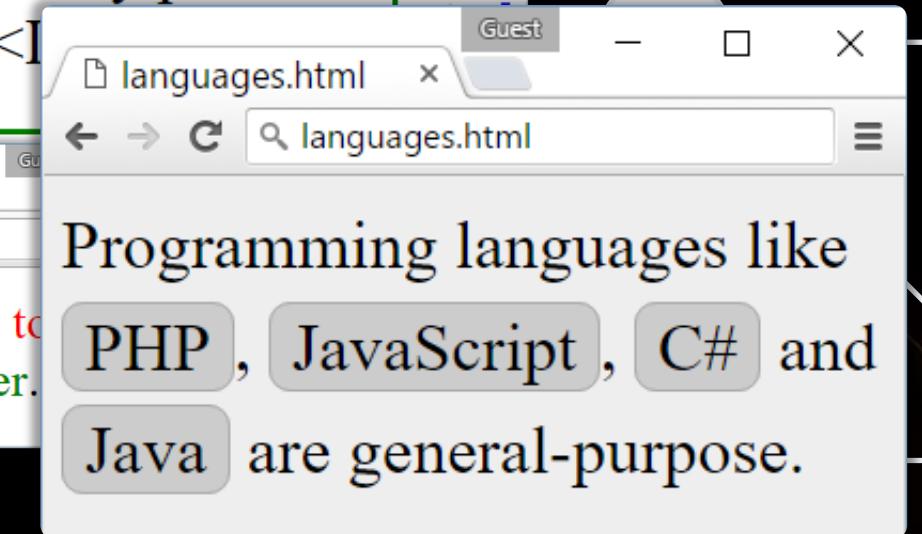
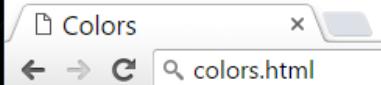
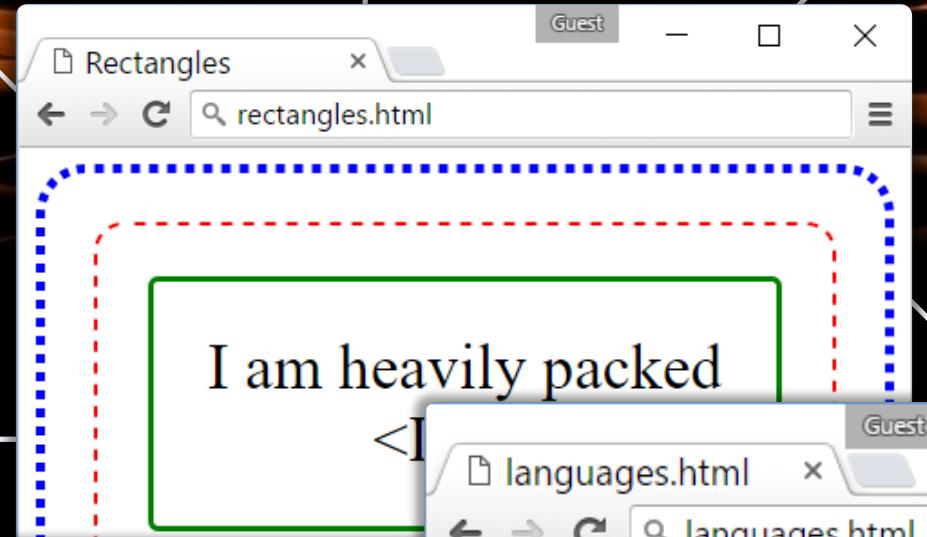
First name:	<input type="text"/>
Last name:	<input type="text"/>
Email:	<input type="text"/>
Town:	<input type="text" value="Sofia"/> ▼

At the bottom right of the form is a "Submit" button.

Exercise – Styling the Contact Us Form (CSS)

```
form {  
    background: #eee;  
    display: inline-block;  
}  
  
form span {  
    display: inline-block;  
    width: 80px;  
    background: #eee;  
    margin: 10px 3px 3px 8px;  
}  
  
form input, form select {  
    width: 150px;
```

```
height: 20px;  
margin: 0px 5px;  
box-sizing: border-box;  
}  
  
form select {  
    margin: 1px;  
}  
  
form input[type='submit'] {  
    display: block;  
    margin: 10px 0 12px 98px;  
    width: 80px; height: 30px;  
}
```



Styling with CSS

Live Exercises in Class (Lab)

Summary

- HTML describes text with formatting, images, tables, forms, etc.
 - Uses tags like `<p>`, `` and ``
- CSS adds styling to the HTML documents
 - Font, color, background, alignment, ...
 - Layout, position, size, margins, paddings, ...
- Web sites consist of HTML + CSS + images
 - May hold JavaScript code and other assets



Questions?

