University of Miskolc
Faculty of Mechanical Engineering and Informatics

Java Web Application Development Technology

N13020008

HTML, CSS, JS

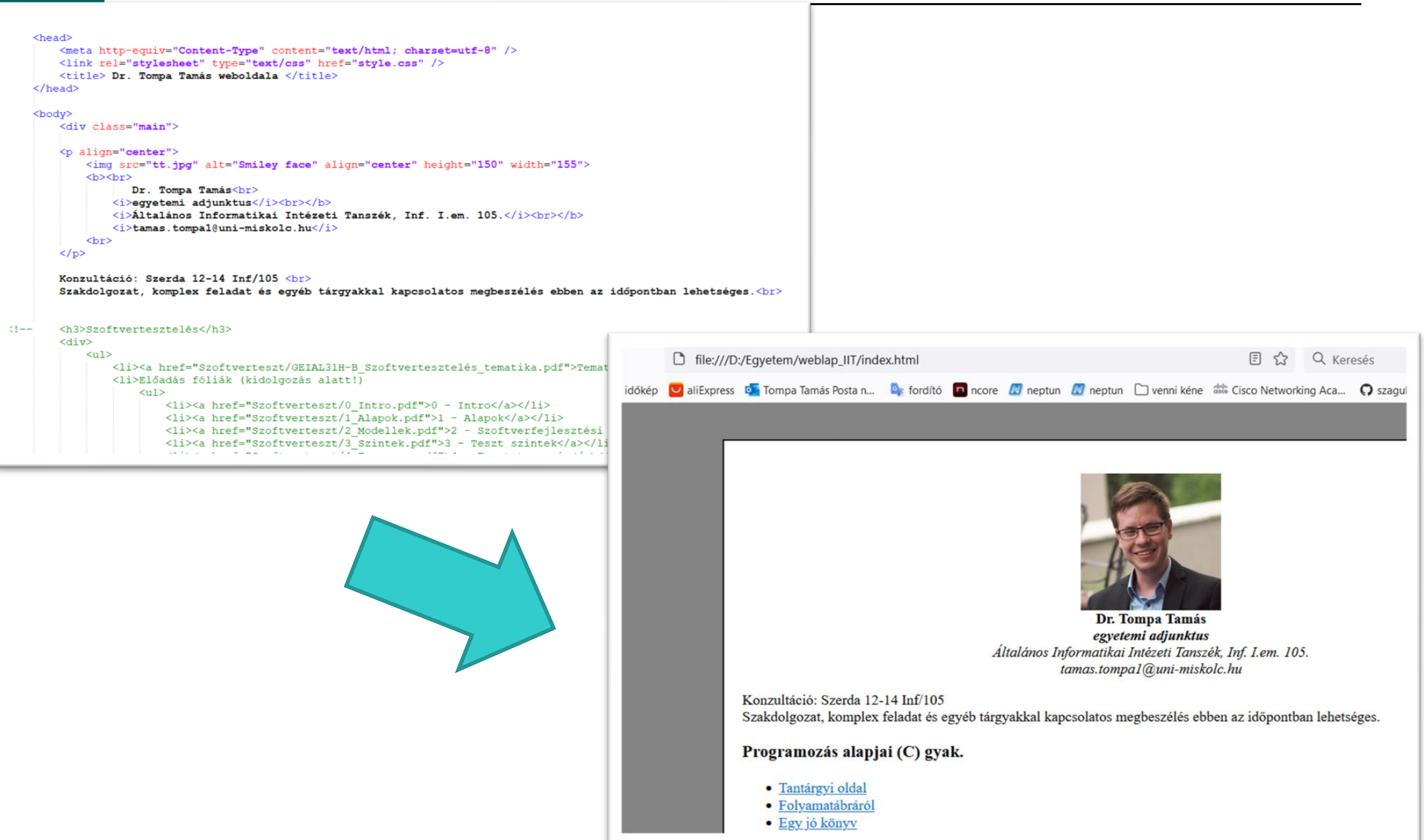
Tamás Tompa, PhD
assistant professor
Department of Information Technology
University of Miskolc

2024.

What is HTML?

- HTML stands for **Hyper Text Markup Language**
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
- File type: .HTML -> Can be edited in a text editor

What is HTML?



```
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css" />
    <title> Dr. Tompa Tamás weboldala </title>
</head>

<body>
    <div class="main">

    <p align="center">
        <img src="tt.jpg" alt="Smiley face" align="center" height="150" width="155">
        <b><br>
              Dr. Tompa Tamás<br>
            <i>egyetemi adjunktus</i><br></b>
            <i>Általános Informatikai Intézeti Tanszék, Inf. I.em. 105.</i><br></b>
            <i>tamas.tompa1@uni-miskolc.hu</i>
        <br>
    </p>

    Konzultáció: Szerda 12-14 Inf/105 <br>
    Szakdolgozat, komplex feladat és egyéb tárgyakkal kapcsolatos megbeszélés ebben az időpontban lehetséges.<br>


:!--    <h3>Szoftvertesztelés</h3>
    <div>
        <ul>
            <li><a href="Szoftverteszt/GEIAL31H-B_Szoftvertesztelés_tematika.pdf">Temat
            <li>Előadás fóliák (kidolgozás alatt!)
                <ul>
                    <li><a href="Szoftverteszt/0_Intro.pdf">0 - Intro</a></li>
                    <li><a href="Szoftverteszt/1_Alapok.pdf">1 - Alapok</a></li>
                    <li><a href="Szoftverteszt/2_Modellek.pdf">2 - Szoftverfejlesztési
                    <li><a href="Szoftverteszt/3_Szintek.pdf">3 - Teszt szintek</a></li
```

Dr. Tompa Tamás
egyetemi adjunktus
Általános Informatikai Intézeti Tanszék, Inf. I.em. 105.
tamas.tompa1@uni-miskolc.hu

Konzultáció: Szerda 12-14 Inf/105
Szakdolgozat, komplex feladat és egyéb tárgyakkal kapcsolatos megbeszélés ebben az időpontban lehetséges.

Programozás alapjai (C) gyak.

- Tantárgyi oldal
- Folyamatábráról
- Egy jó könyv

HTML history

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

A Simple HTML Document

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document

- The <html> element is the root element of an HTML page

- The <head> element contains meta information about the HTML page

- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

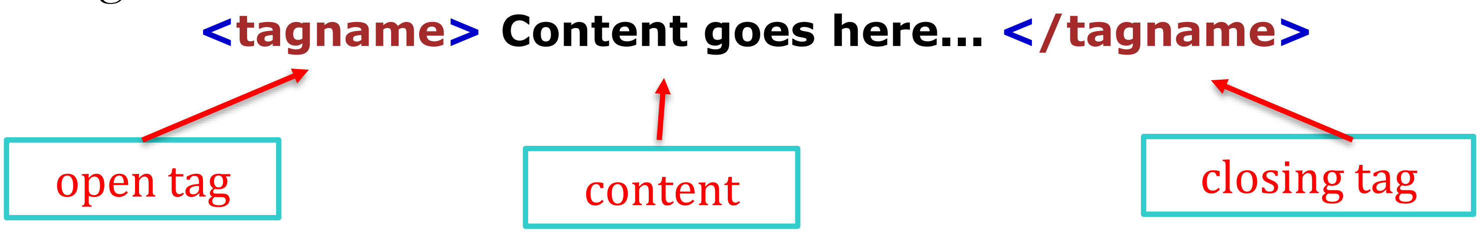
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

- The <h1> element defines a large heading

- The <p> element defines a paragraph

HTML tag

- An **HTML element** is defined by a start tag, some content, and an end tag:



<tagname> Content goes here... </tagname>

open tag

content

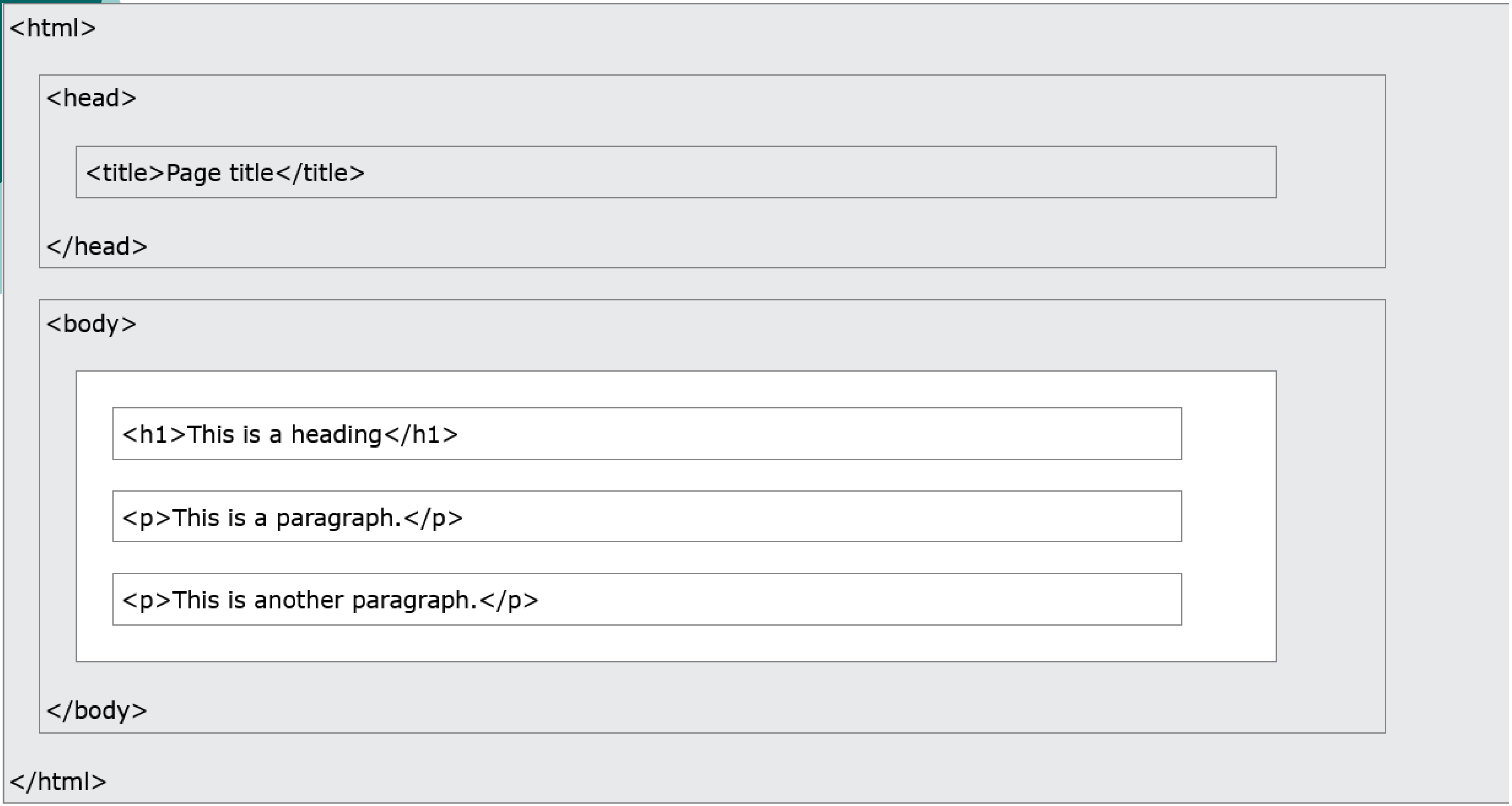
closing tag

- The HTML **element** is everything from the start tag to the end tag:

```
<h1> My First Heading </h1>
<p> My first paragraph. </p>
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
<br>	none	none

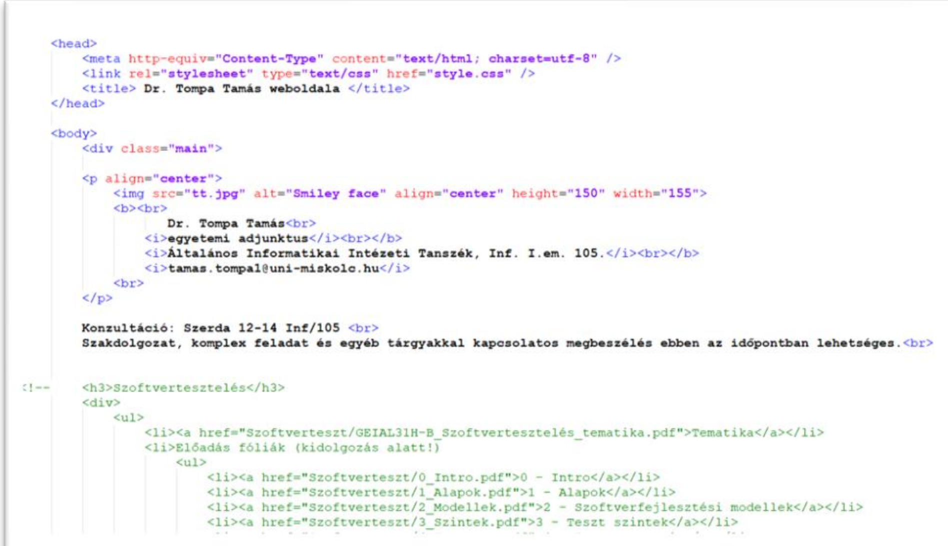
HTML Page Structure



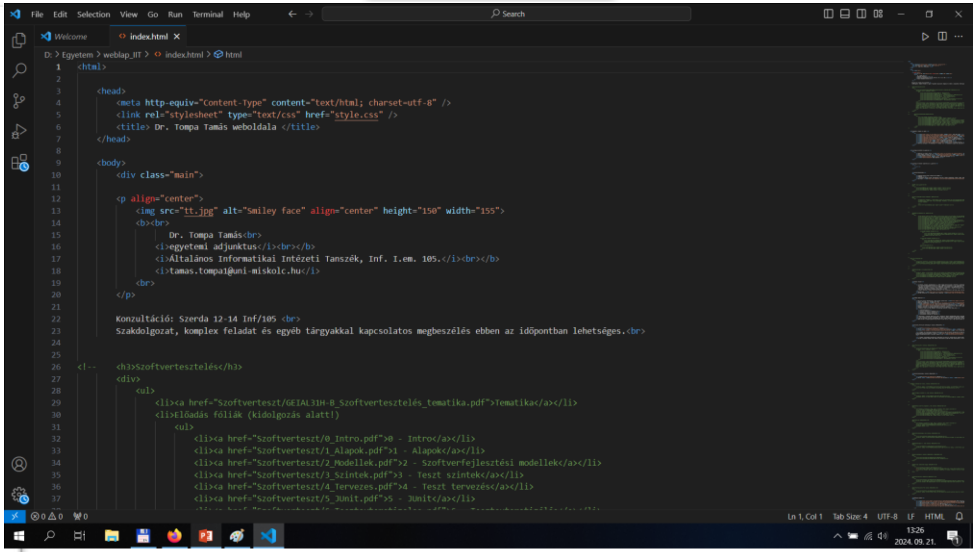
```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
      <h1>This is a heading</h1>
      <p>This is a paragraph.</p>
      <p>This is another paragraph.</p>
  </body>
</html>
```

HTML editors

- A simple text editor is all you need to edit HTML domcument
  - **Notepad++** (https://notepad-plus-plus.org/downloads/)
  - **VSCode** (https://code.visualstudio.com/)



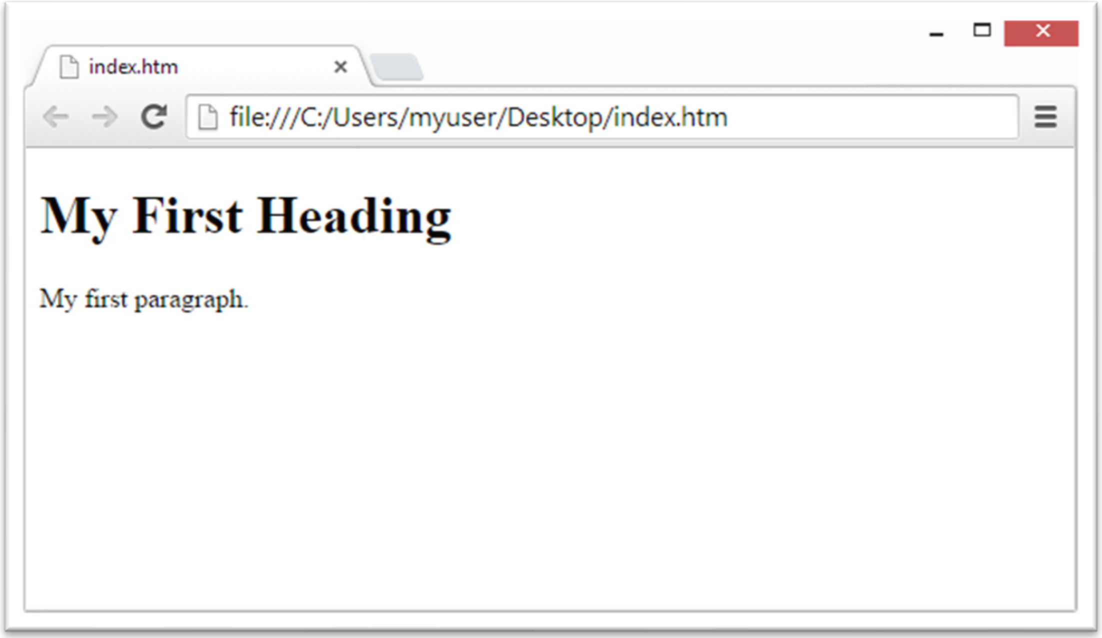
```
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css" />
    <title> Dr. Tompa Tamás weboldala </title>
</head>

<body>
    <div class="main">

    <p align="center">
        <img src="tt.jpg" alt="Smiley face" align="center" height="150" width="155">
        <b><br>
                Dr. Tompa Tamás<br>
            <i>egyetemi adjunktus</i><br></b>
            <i>Általános Informatikai Intézeti Tanszék, Inf. I.em. 105.</i><br></b>
            <i>tamas.tompa1@uni-miskolc.hu</i>
        <br>
    </p>

    Konzultáció: Szerda 12-14 Inf/105 <br>
    Szakdolgozat, komplex feladat és egyéb tárgyakkal kapcsolatos megbeszélés ebben az időpontban lehetséges.<br>


<!--    <h3>Szoftvertesztelés</h3>
    <div>
        <ul>
            <li><a href="Szoftverteszt/GEIAL31H-B_Szoftvertesztelés_tematika.pdf">Tematika</a></li>
            <li>Előadás fóliák (kidolgozás alatt!)
                <ul>
                    <li><a href="Szoftverteszt/0_Intro.pdf">0 - Intro</a></li>
                    <li><a href="Szoftverteszt/1_Alapok.pdf">1 - Alapok</a></li>
                    <li><a href="Szoftverteszt/2_Modellek.pdf">2 - Szoftverfejlesztési modellek</a></li>
                    <li><a href="Szoftverteszt/3_Szintek.pdf">3 - Teszt szintek</a></li>
```



```
<html>

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <link rel="stylesheet" type="text/css" href="style.css" />
        <title> Dr. Tompa Tamás weboldala </title>
    </head>

    <body>
        <div class="main">

        <p align="center">
            <img src="tt.jpg" alt="Smiley face" align="center" height="150" width="155">
            <b><br>
                    Dr. Tompa Tamás<br>
                <i>egyetemi adjunktus</i><br></b>
                <i>Általános Informatikai Intézeti Tanszék, Inf. I.em. 105.</i><br></b>
                <i>tamas.tompa1@uni-miskolc.hu</i>
            <br>
        </p>

        Konzultáció: Szerda 12-14 Inf/105 <br>
        Szakdolgozat, komplex feladat és egyéb tárgyakkal kapcsolatos megbeszélés ebben az időpontban lehetséges.<br>


<!--    <h3>Szoftvertesztelés</h3>
        <div>
            <ul>
                <li><a href="Szoftverteszt/GEIAL31H-B_Szoftvertesztelés_tematika.pdf">Tematika</a></li>
                <li>Előadás fóliák (kidolgozás alatt!)
                    <ul>
                        <li><a href="Szoftverteszt/0_Intro.pdf">0 - Intro</a></li>
                        <li><a href="Szoftverteszt/1_Alapok.pdf">1 - Alapok</a></li>
                        <li><a href="Szoftverteszt/2_Modellek.pdf">2 - Szoftverfejlesztési modellek</a></li>
                        <li><a href="Szoftverteszt/3_Szintek.pdf">3 - Teszt szintek</a></li>
                        <li><a href="Szoftverteszt/4_Tervezes.pdf">4 - Teszt tervezés</a></li>
                        <li><a href="Szoftverteszt/5_JUnit.pdf">5 - JUnit</a></li>
```

HTML viewer

- **Web Browsers**
- The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.
- A browser does not display the HTML tags, but uses them to determine how to display the document:

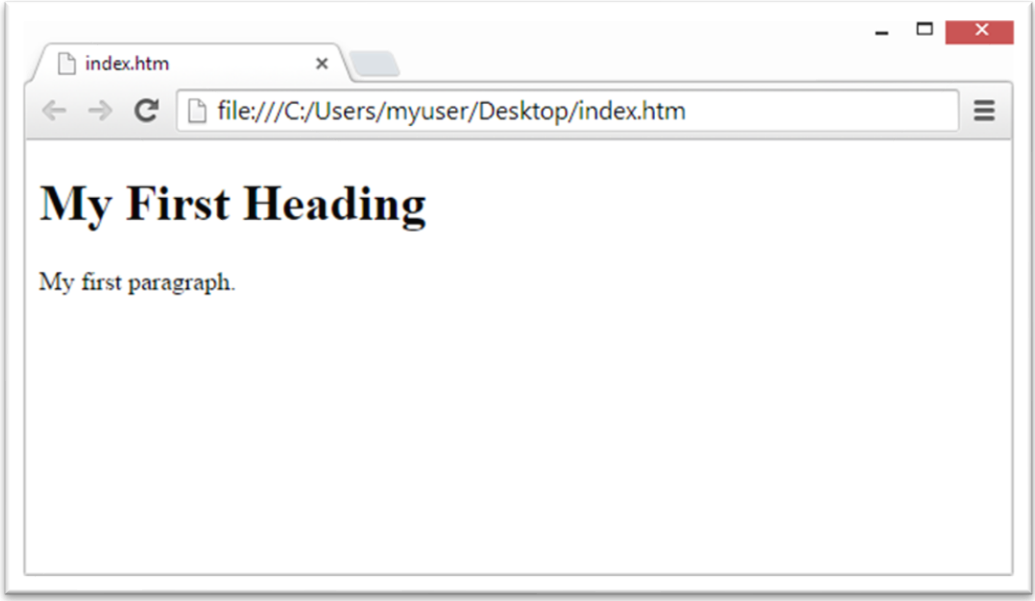


My First Heading

My first paragraph.

Simple HTML example

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```



My First Heading

My first paragraph.

- Save the file on your computer. Select File > Save as in the Notepad menu
- Name the file "index.htm" and set the encoding to UTF-8 (which is the preferred encoding for HTML files)

HTML documents

- All HTML documents must start with a document type declaration: <!DOCTYPE html>
  - represents the document type, and helps browsers to display web pages correctly

- The HTML document itself begins with <html> and ends with </html>

- The visible part of the HTML document is between <body> and </body>

Empty HTML Elements

- HTML elements with no content are called empty elements:
  - <br>
  - <p>

- The <br> tag defines a line break, and is an empty element without a closing tag:

```
<p>This is a <br> paragraph with a line break.</p>
```

- HTML is Not Case Sensitive
  - HTML tags are not case sensitive: <P> means the same as <p>

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in the start tag
- Attributes usually come in name/value pairs like: **name="value"**
- Examples:
  - The `<a>` tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to:

    ```
    <a href="https://www.google.com">The first link</a>
    ```

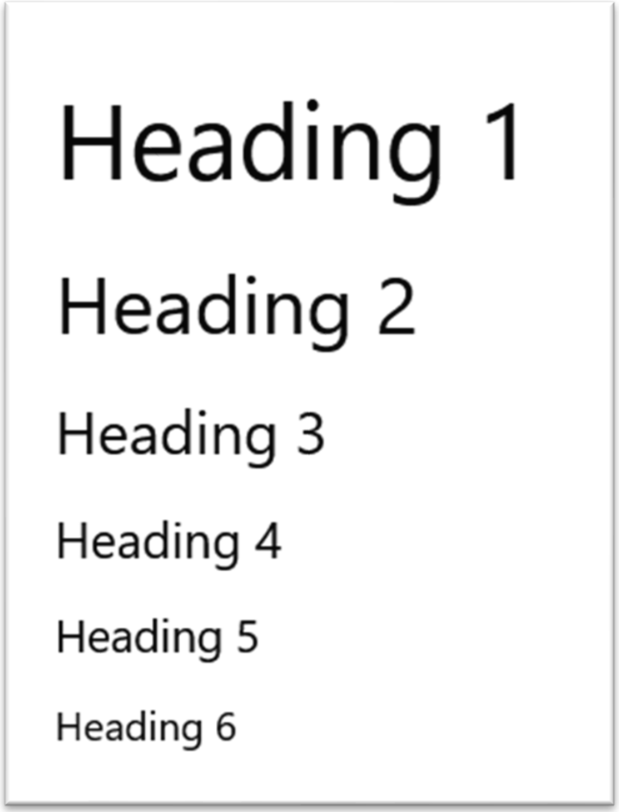
  - The `<img>` tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed:

    ```
    <img src="img_girl.jpg" width="500" height="600">
    ```

HTML Headings

- HTML headings are titles or subtitles that you want to display on a webpage

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```



Heading 1

Heading 2

Heading 3

Heading 4

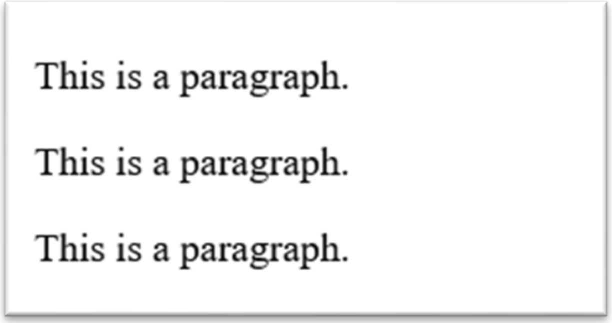
Heading 5

Heading 6

HTML Paragraphs

- A paragraph always starts on a new line, and is usually a block of text
- The HTML <p> element defines a paragraph

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```



This is a paragraph.

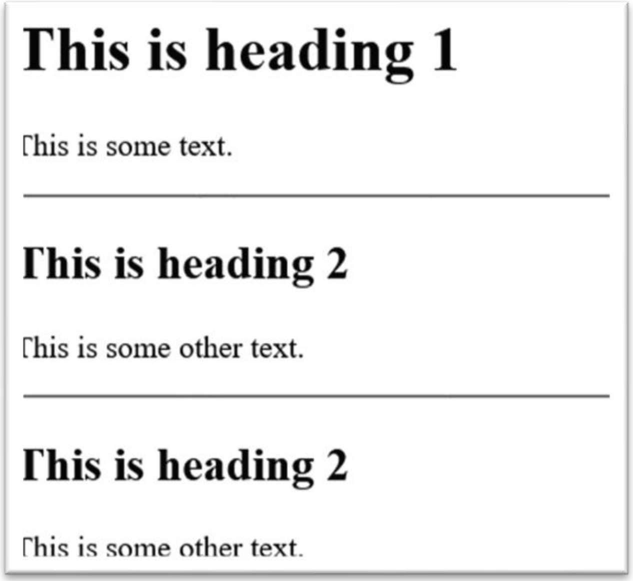
This is a paragraph.

This is a paragraph.

HTML Horizontal Rules

- The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.
- The <hr> element is used to separate content (or define a change) in an HTML page:

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```



This is heading 1

This is some text.

This is heading 2

This is some other text.

This is heading 2

This is some other text.

HTML Tag Reference

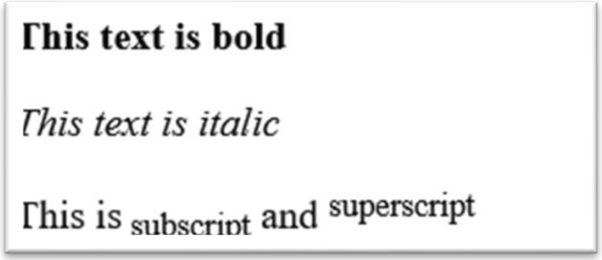
- The tag reference contains additional information about HTML elements and their attributes

Tag	Description
<p>	Defines a paragraph
<hr>	Defines a thematic change in the content
<br>	Inserts a single line break
<pre>	Defines pre-formatted text

HTML formatting

- HTML contains several elements for defining text with a special meaning

```
<!DOCTYPE html>
<html>
<body>

<p><b>This text is bold</b></p>
<p><i>This text is italic</i></p>
<p>This is<sub> subscript</sub>
and <sup>superscript</sup></p>

</body>
</html>
```



This text is bold

This text is italic

This is $_{\text{subscript}}$ and $^{\text{superscript}}$

HTML formatting elements

- Formatting elements were designed to display special types of text:

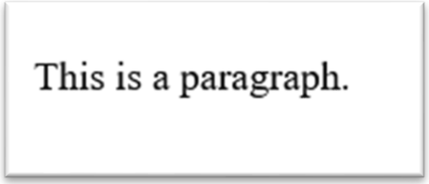
Tag	Description
<b>	Defines bold text
<em>	Defines emphasized text
<i>	Defines a part of text in an alternate voice or mood
<small>	Defines smaller text
<strong>	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

HTML comments

- HTML comments are not displayed in the browser, but they can help document your HTML source code

```
<!-- Write your comments here -->
```

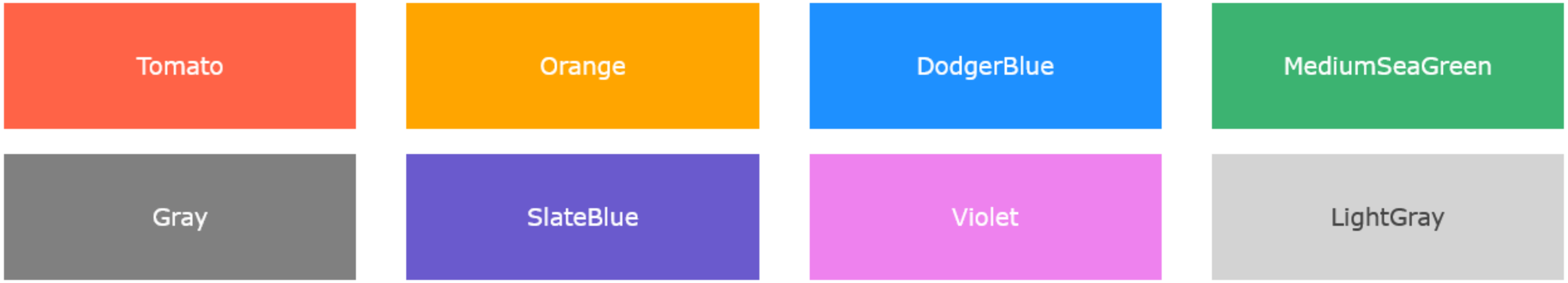
```
<!-- This is a comment -->

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

<!-- Remember to add more
information here -->
```



This is a paragraph.

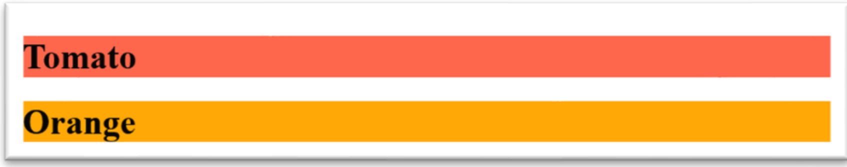
HTML Colors

- HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values



Tomato	Orange	DodgerBlue	MediumSeaGreen
Gray	SlateBlue	Violet	LightGray

```
<h1 style="background-
color:Tomato;">Tomato</h1>
<h1 style="background-
color:Orange;">Orange</h1>
```



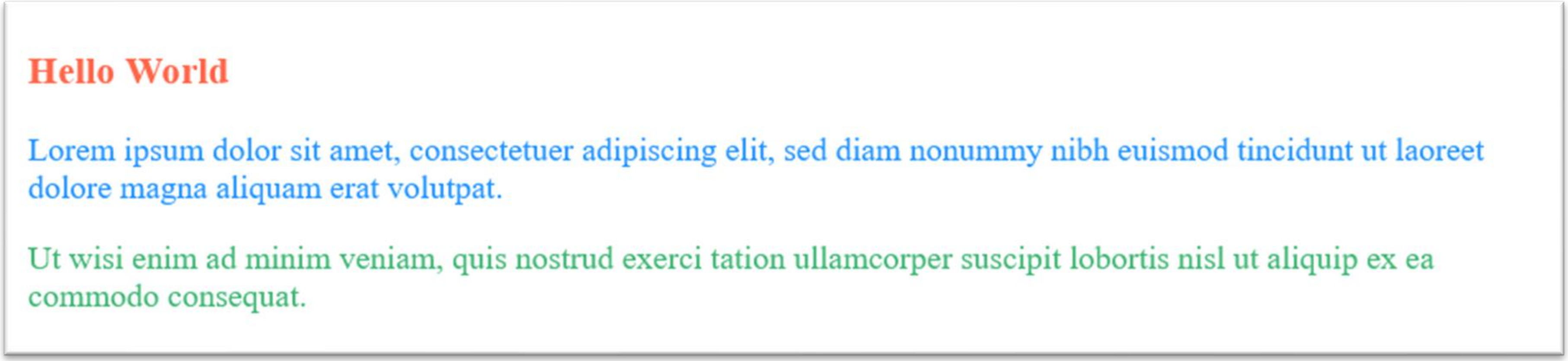
Tomato

Orange

HTML Colors

- Text color

```html
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```



Hello World

Lorem ipsum dolor sit amet, consectetuer 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 lobortis nisl ut aliquip ex ea commodo consequat.

HTML links

- Links are found in nearly all web pages. Links allow users to click their way from page to page

```
<a href="url">link text</a>
```

- By default, links will appear as follows in all browsers:
  - an unvisited link is underlined and blue
  - a visited link is underlined and purple
  - an active link is underlined and red

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

HTML images

- Images can improve the design and the appearance of a web page
- The <img> tag has two required attributes:
  - src - Specifies the path (URL) to the image
  - alt - Specifies an alternate text (description) for the image

```
<img src="pic_trulli.jpg" alt="Italian Trulli">
<img src="img_girl.jpg" alt="Girl in a jacket">
<img src="img_chania.jpg" alt="Flowers in Chania">
```

- You can use the style attribute to specify the width and height of an image

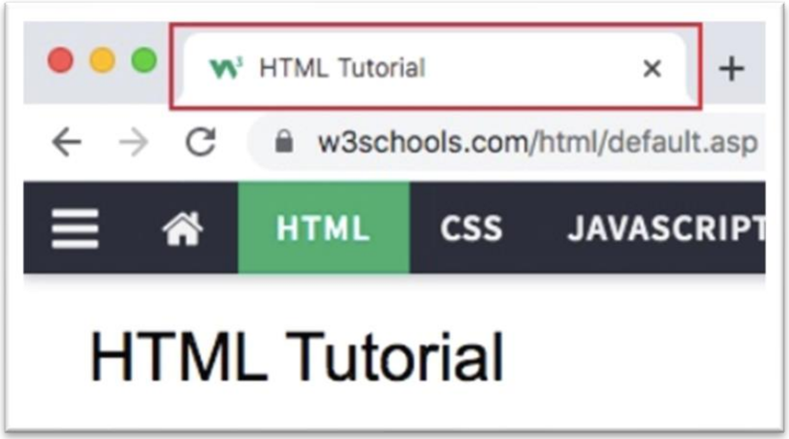
```
<img src="img_girl.jpg" alt="Girl in a jacket„
style="width:500px;height:600px;">
```

HTML page title

- Every web page should have a page title to describe the meaning of the page

- The <title> element adds a title to your page:

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Tutorial</title>
</head>
<body>

The content of the document......

</body>
</html>
```



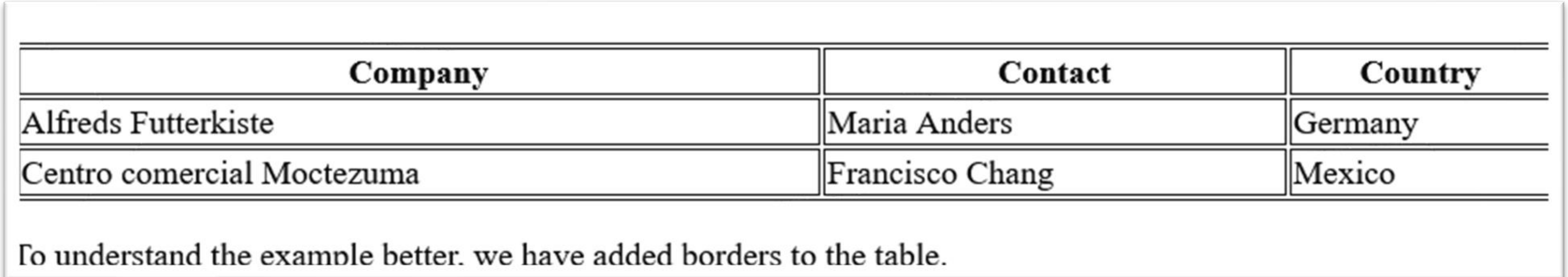
HTML Tutorial

w3schools.com/html/default.asp

HTML Tutorial

HTML tables

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

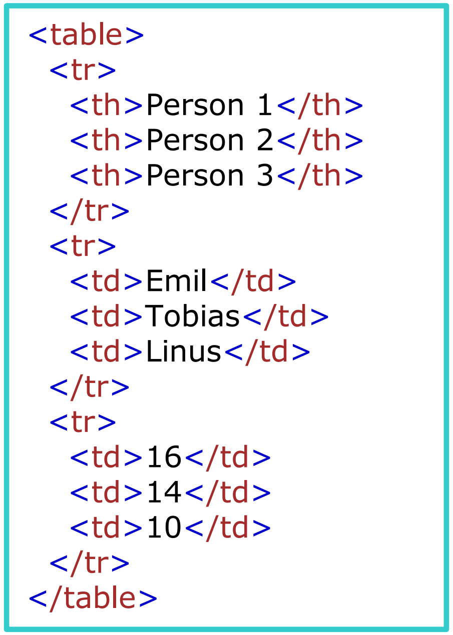


Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

To understand the example better, we have added borders to the table.

HTML tables

- Each table **cell** is defined by a `<td>` and a `</td>` tag
  - td stands for table data
- Each table **row** starts with a `<tr>` and ends with a `</tr>` tag
  - tr stands for table row
- **th** stands for table header `<th>Person 1</th>`



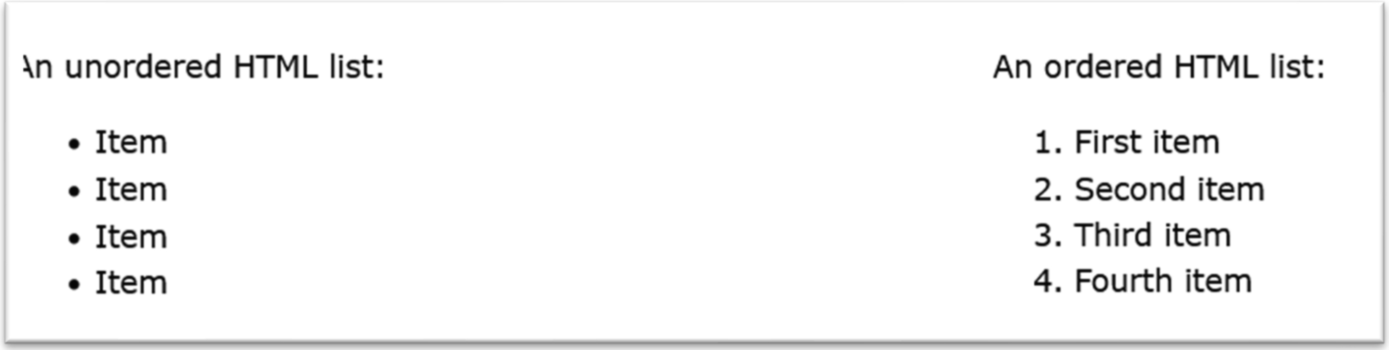
```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

HTML tables

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table
<caption>	Defines a table caption
<colgroup>	Specifies a group of one or more columns in a table for formatting
<col>	Specifies column properties for each column within a <colgroup> element
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table

HTML lists

- ○ HTML lists allow web developers to group a set of related items in lists



An unordered HTML list:

- Item
- Item
- Item
- Item

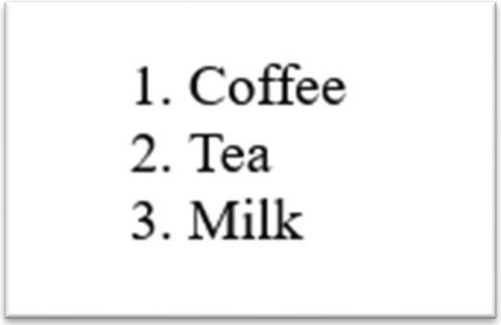
An ordered HTML list:

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

Tag	Description
<ul>	Defines an unordered list
<ol>	Defines an ordered list
<li>	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

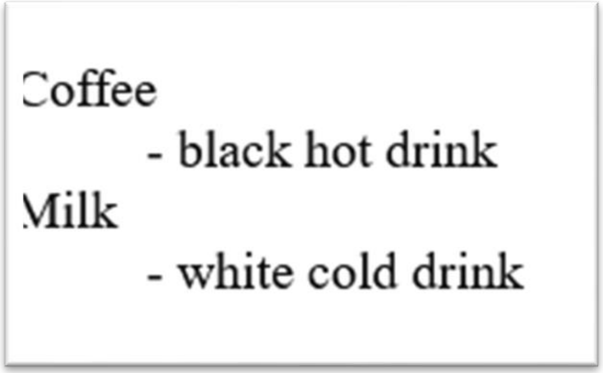
HTML lists

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



1. Coffee
2. Tea
3. Milk

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



Coffee

- black hot drink

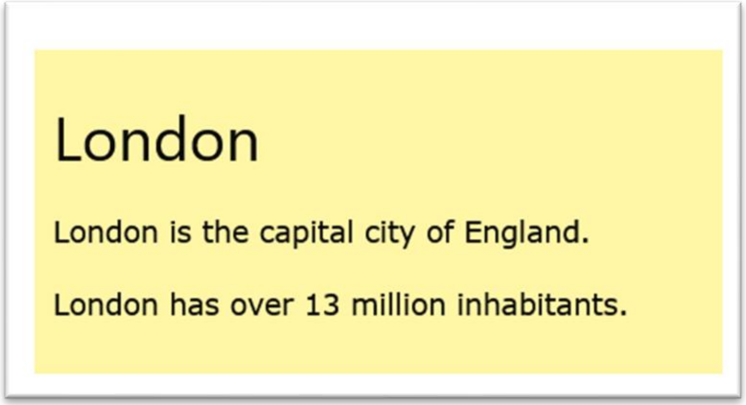
Milk

- white cold drink

HTML div

- The <div> element is used as a container for other HTML elements
- by default a block element, meaning that it takes all available width, and comes with line breaks before and after

```
<div>
  <h2>London</h2>
  <p>London is the capital city of
England.</p>
  <p>London has over 13 million
inhabitants.</p>
</div>
```



London

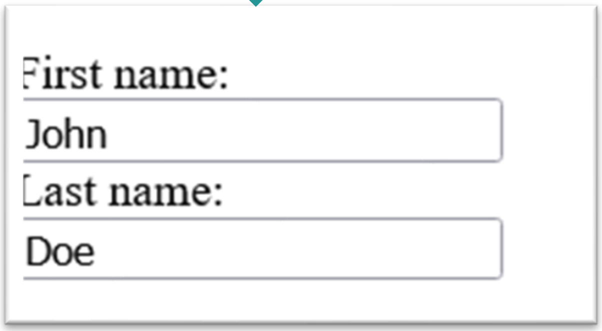
London is the capital city of England.

London has over 13 million inhabitants.

HTML forms

- An HTML form is used to collect user input. The user input is most often sent to a server for processing

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```



First name:

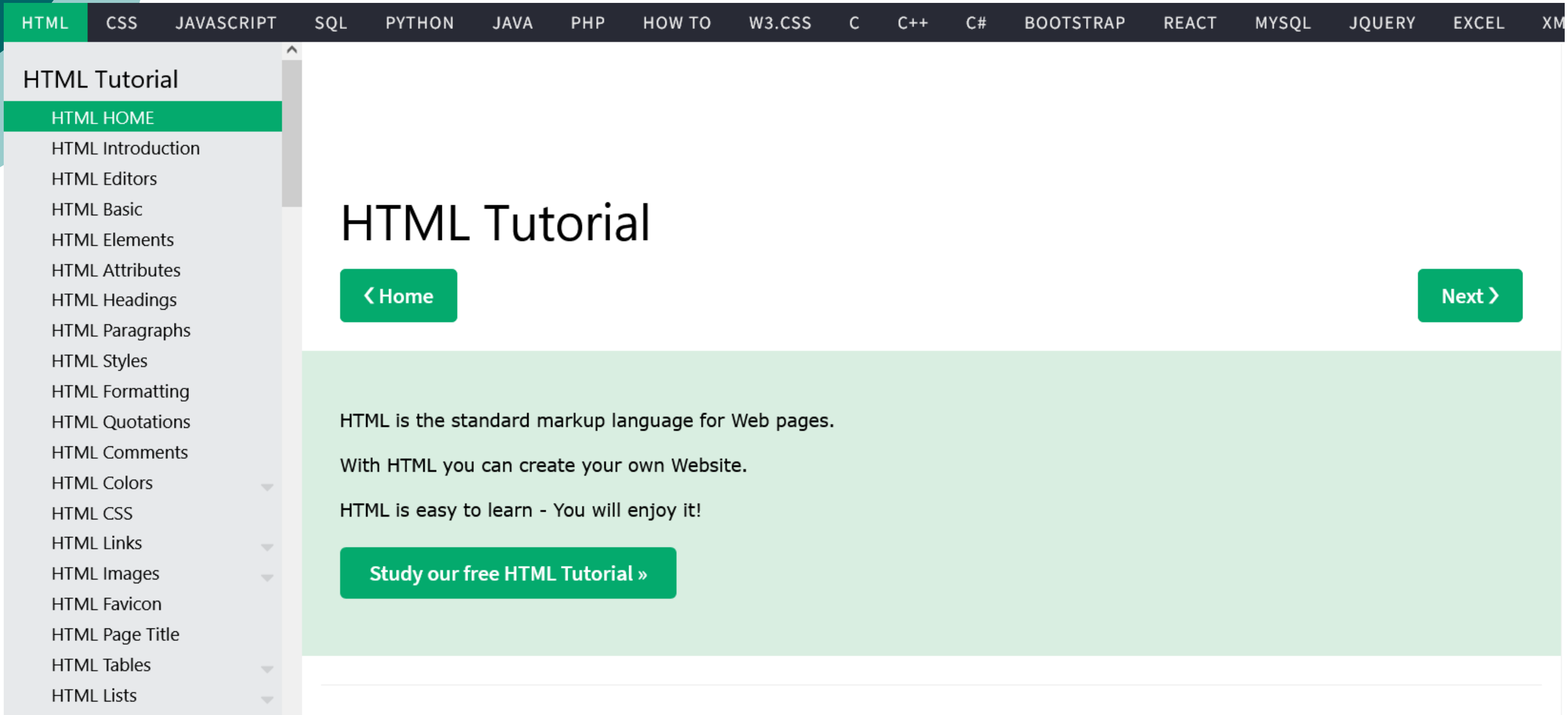
John

Last name:

Doe

HTML cont…

- More examples on the https://www.w3schools.com/html/default.asp web page...



HTML Tutorial

HTML is the standard markup language for Web pages.

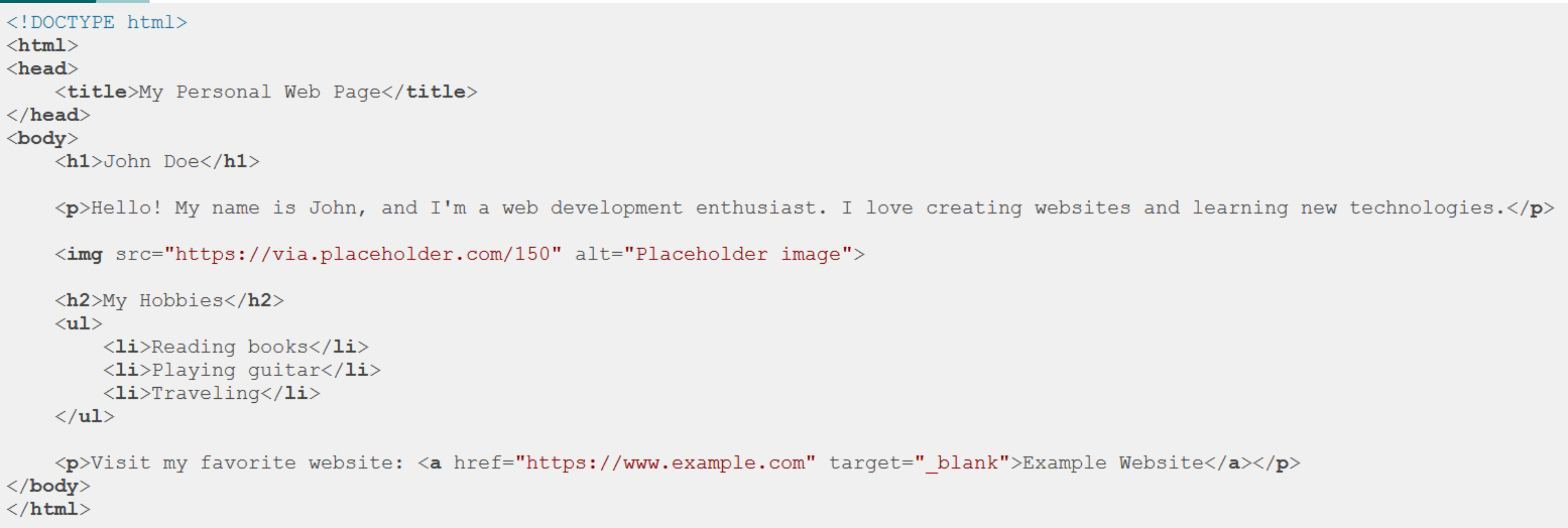
With HTML you can create your own Website.

HTML is easy to learn - You will enjoy it!

HTML Task: Create a Basic Web Page…

- Create a basic webpage using HTML that includes a header, a paragraph, an image, and a list of your favourite hobbies.
  - Create a new HTML file called index.html
  - In the HTML file, write the following content:
  - Add a title to your webpage
  - Add a header (<h1>) with your name
  - Add a paragraph (<p>) introducing yourself in a few sentences
  - Add an image of your choice from the internet using the <img> tag
  - Create an unordered list (<ul>) of at least 3 of your favourite hobbies
  - Add a link (<a>) to your favourite website

HTML Task: Create a Basic Web Page…

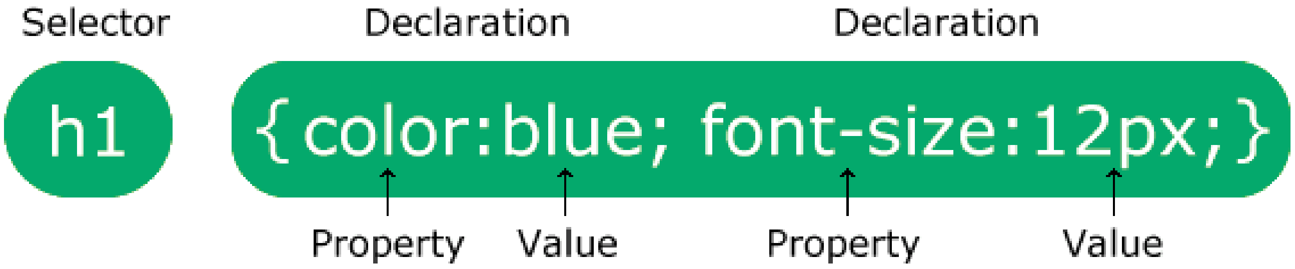


```html
<!DOCTYPE html>
<html>
<head>
    <title>My Personal Web Page</title>
</head>
<body>
    <h1>John Doe</h1>

    <p>Hello! My name is John, and I'm a web development enthusiast. I love creating websites and learning new technologies.</p>

    <img src="https://via.placeholder.com/150" alt="Placeholder image">

    <h2>My Hobbies</h2>
    <ul>
        <li>Reading books</li>
        <li>Playing guitar</li>
        <li>Traveling</li>
    </ul>

    <p>Visit my favorite website: <a href="https://www.example.com" target="_blank">Example Website</a></p>
</body>
</html>
```

What is CSS?

- CSS is the language we use to **style a Web page**
- 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
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes
- **HTML was NEVER intended to contain tags for formatting a web page!**
- **HTML** was created to **describe the content of a web page**, like
- **To solve this problem**, the World Wide Web Consortium (W3C) **created CSS**

CSS syntax



Selector

Declaration

Declaration

h1 {color:blue; font-size:12px;}

Property

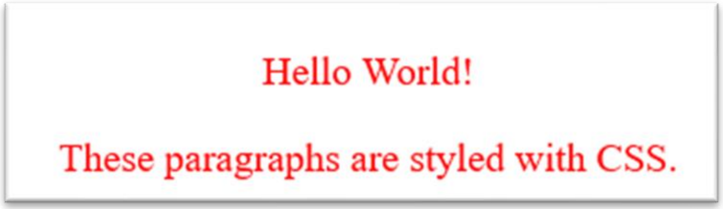
Value

Property

Value

- The selector points to the HTML element you want to style
- The declaration block contains one or more declarations separated by semicolons
- Each declaration includes a CSS property name and a value, separated by a colon
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces

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

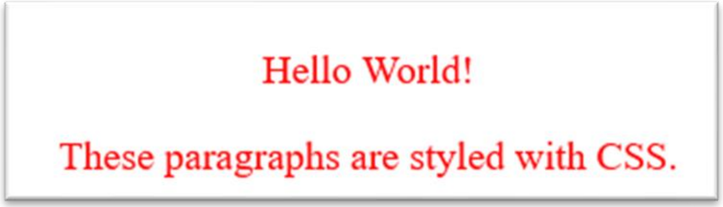


Hello World!

These paragraphs are styled with CSS.

CSS syntax example

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



Hello World!

These paragraphs are styled with CSS.

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

CSS insert

- There are three ways of inserting a style sheet:
  - **External CSS**
    - external css file
  - **Internal CSS**
    - in the HTML file, between `<style>` … `</style>`
  - **Inline CSS**
    - unique style for a single element

- Cascading Order
  - there is more than one style specified for an HTML element
    - 1. Inline style (inside an HTML element)
    - 2. External and internal style sheets (in the head section)
    - 3. Browser default

CSS insert - external

```html
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"
href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

"mystyle.css"

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

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

CSS insert - internal

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

CSS insert - inline

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a
heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

CSS selectors

- CSS selectors are used to "find" (or select) the HTML elements you want to style
- We can divide CSS selectors into five categories:
  - Simple selectors (select elements based on name, id, class)
  - Combinator selectors (select elements based on a specific relationship between them)
  - Pseudo-class selectors (select elements based on a certain state)
  - Pseudo-elements selectors (select and style a part of an element)
  - Attribute selectors (select elements based on an attribute or attribute value)

- All <p> elements on the page will be center-aligned, with a red text color:

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

CSS selectors

○ id Selector

- the id selector **uses the id** attribute of an HTML element to select a specific element
- the id of an element is **unique within a page**, so the id selector is used to select one unique element
- **to select an element** with a specific id, **write a hash (#)** character, followed by the id of the element

```
#para1 {
  text-align: center;
  color: red;
}

…
<p id="para1">Hello World!</p>
<p>This paragraph is not affected
by the style.</p>
```

CSS selectors

- **class Selector**
  - the class selector selects HTML elements with a **specific class attribute**
  - **to select elements** with a specific class, **write a period (.)** character, followed by the class name

```
.center {
  text-align: center;
  color: red;
}
…
<h1 class="center">Red
and center-aligned
heading</h1>
<p class="center">Red
and center-aligned
paragraph.</p>
```

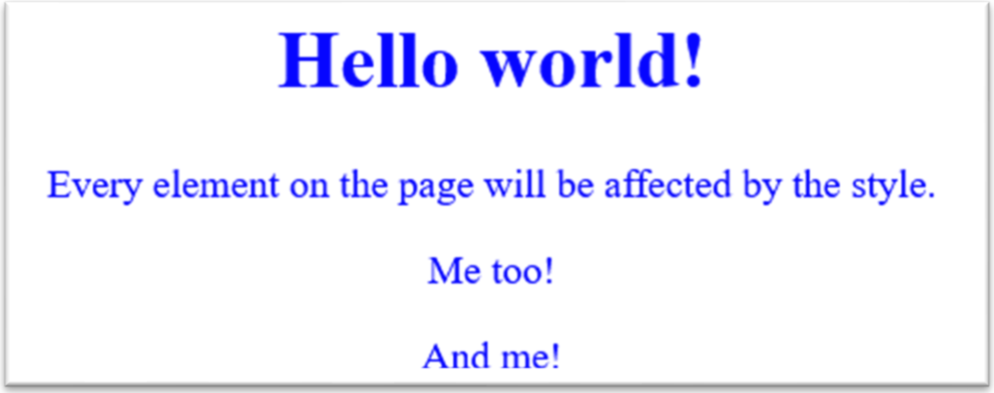
```
p.center {
  text-align: center;
  color: red;
}
…
<h1 class="center">This
heading will not be
affected</h1>
<p class="center">This
paragraph will be red and
center-aligned.</p>
```

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

p.large {
  font-size: 300%;
}
…
<p class="center
large">This paragraph refers
to two classes.</p>
```

CSS selectors

- ## **universal Selector**
  - the universal selector (*) **selects all HTML elements** on the page

```css
* {
  text-align: center;
  color: blue;
}
```



Hello world!

Every element on the page will be affected by the style.

Me too!

And me!

CSS selectors

- **grouping Selector**
  - the grouping selector selects **all the HTML elements with the same style definitions**
  - it will be better to group the selectors, to **minimize the code**
  - to group selectors, separate each selector with a comma

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

h2 {
  text-align: center;
  color: red;
}
```

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

→

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

CSS selectors summerize

Selector	Example	Example description
#id	#firstname	Selects the element with id="firstname"
.class	.intro	Selects all elements with class="intro"
element.class	p.intro	Selects only <p> elements with class="intro"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element,..	div, p	Selects all <div> elements and all <p> elements

CSS comments

- Comments are used to explain the code, and may help when you edit the source code at a later date
- Comments are ignored by browsers

```css
/* This is a single-line comment */
p {
  color: red;
}
```

```css
p {
  color: /*red*/blue;
}
```

```css
p {
  color: red;  /* Set text color to red */
}
```

```css
/* This is
a multi-line
comment */

p {
  color: red;
}
```

CSS colors

- Background color

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

- Text color

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

- Border color

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

CSS backgrounds

- The CSS background properties are used to add background effects for elements
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position
  - background (shorthand property)

CSS backgrounds

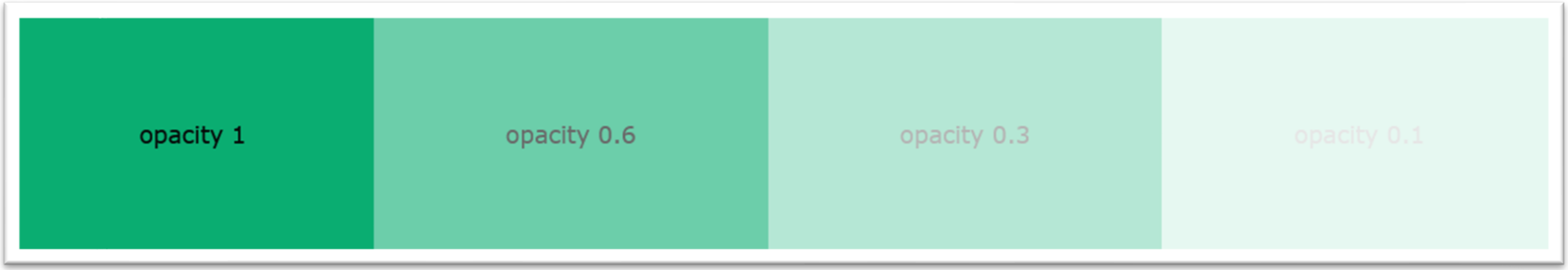
- The CSS background properties are used to add background effects for elements
  - background-color

```
h1 {
  background-color: green;
}

div {
  background-color: lightblue;
}

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

CSS backgrounds

- The CSS background properties are used to add background effects for elements
  - opacity

```
div {
  background-color: green;
  opacity: 0.3;
}
```



opacity 1

opacity 0.6

opacity 0.3

opacity 0.1

CSS backgrounds

- The CSS background properties are used to add background effects for elements
  - background-image

```
body {
  background-image: url("paper.gif");
}
```

```
p {
  background-image: url("paper.gif");
}
```

CSS backgrounds

- The CSS background properties are used to add background effects for elements
  - background-repeat: repeats an image both horizontally and vertically

```
body {
  background-image:
url("gradient_bg.png");
  background-repeat: repeat-x;
}
```

```
body {
  background-image:
url("img_tree.png");
  background-repeat: no-repeat;
}
```

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

CSS backgrounds

- The CSS background properties are used to add background effects for elements
  - background-attachment: specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page)

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
}
```

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: scroll;
}
```

CSS backgrounds

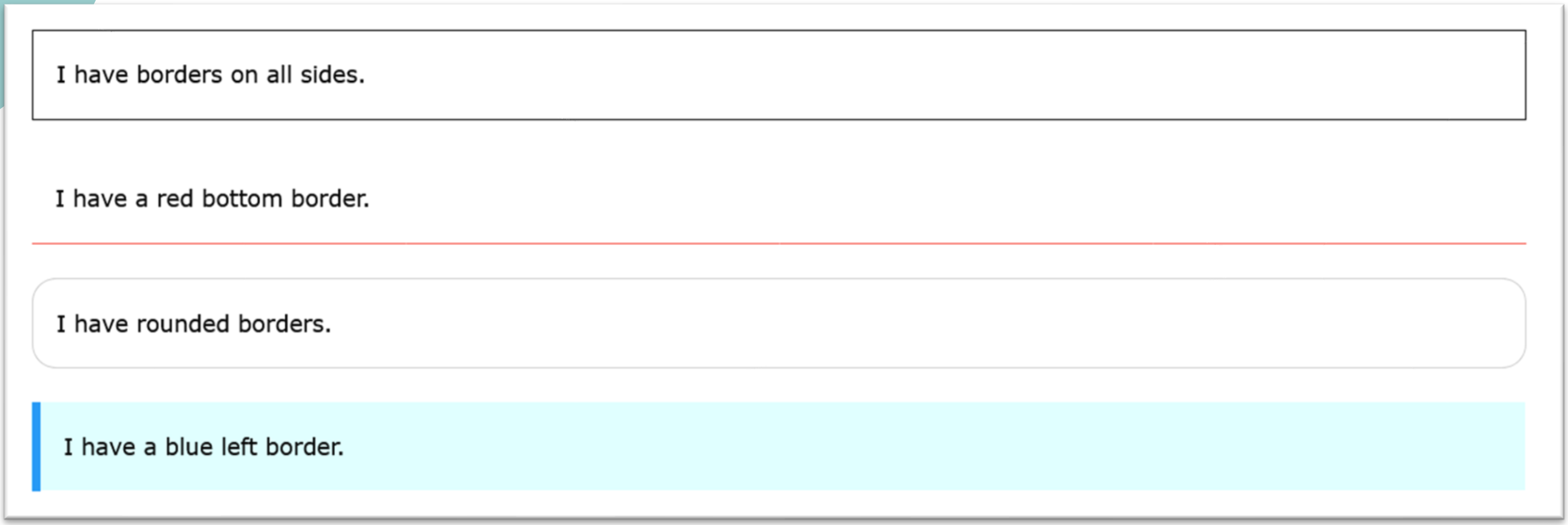
- The CSS background properties are used to add background effects for elements
  - background (shorthand property): to shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property

```
body {
  background-color: #ffffff;
  background-image:
url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
}
```

```
body {
  background: #ffffff url("img_tree.png")
no-repeat right top;
}
```

CSS borders

- The CSS border properties allow you to specify the style, width, and color of an element's border



I have borders on all sides.

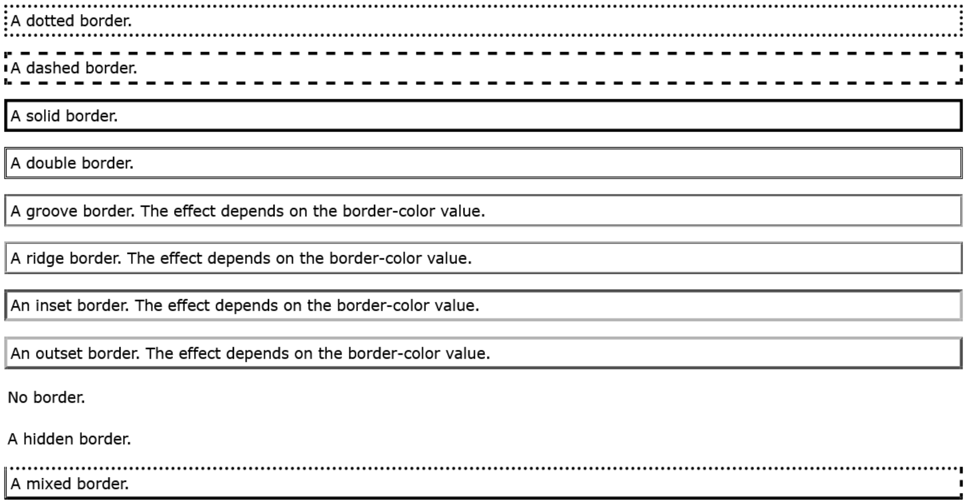
I have a red bottom border.

I have rounded borders.

I have a blue left border.

CSS borders

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```



A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

CSS margins

- Margins are used to create space around elements, outside of any defined borders

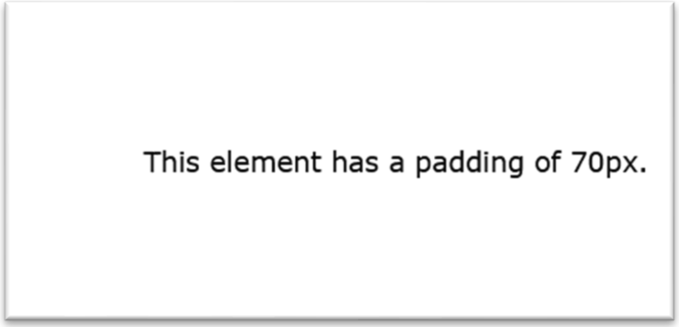
This element has a margin of 70px.

- CSS has properties for specifying the margin for each side of an element:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

CSS padding

- Padding is used to create space around an element's content, inside of any defined borders



This element has a padding of 70px.

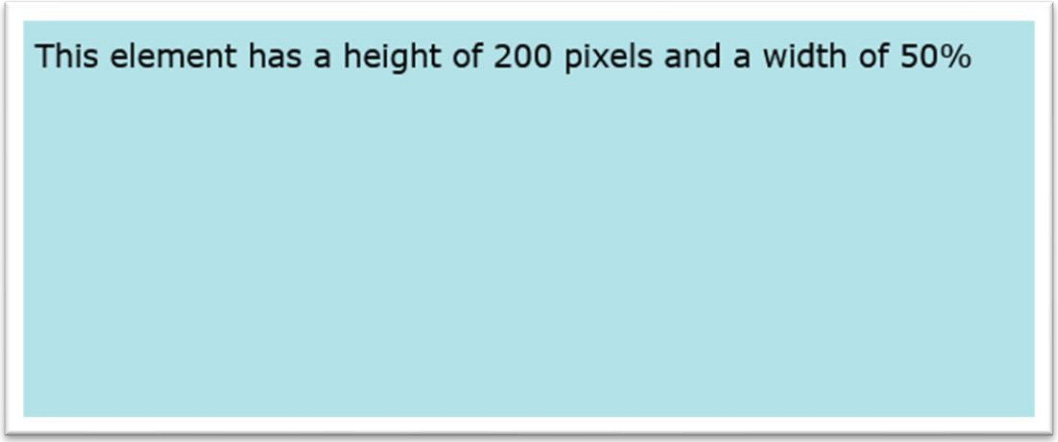
- CSS has properties for specifying the padding for each side of an element:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

CSS Setting height and width

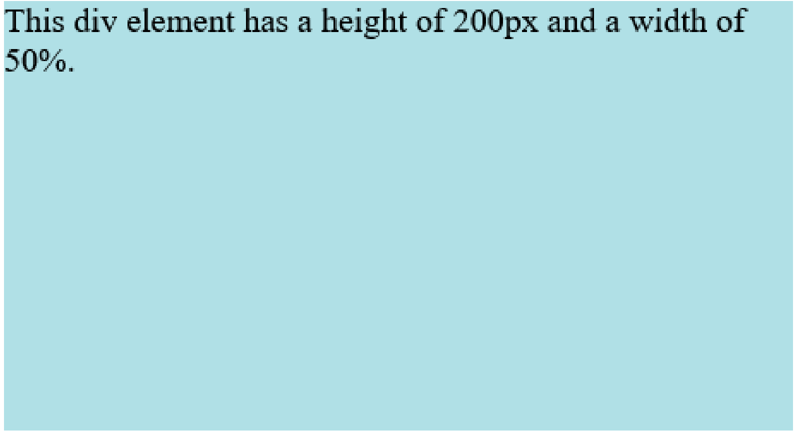
- The height and width properties are used to set the height and width of an element.
- The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element
- The height and width properties may have the following values:
  - auto - This is default. The browser calculates the height and width
  - length - Defines the height/width in px, cm, etc.
  - % - Defines the height/width in percent of the containing block
  - initial - Sets the height/width to its default value
  - inherit - The height/width will be inherited from its parent value

CSS Setting height and width



This element has a height of 200 pixels and a width of 50%

```
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
```



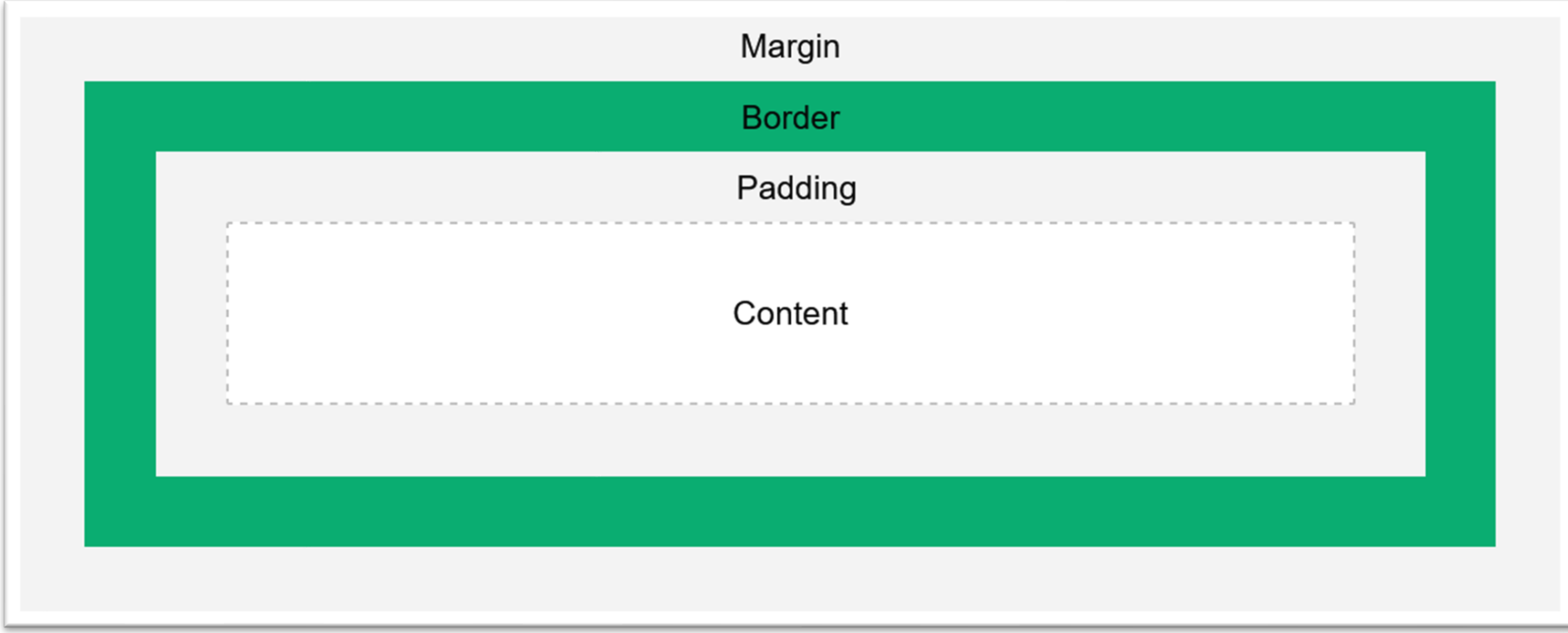
This div element has a height of 200px and a width of 50%.

All CSS Dimension Properties

Property	Description
height	Sets the height of an element
max-height	Sets the maximum height of an element
max-width	Sets the maximum width of an element
min-height	Sets the minimum height of an element
min-width	Sets the minimum width of an element
width	Sets the width of an element

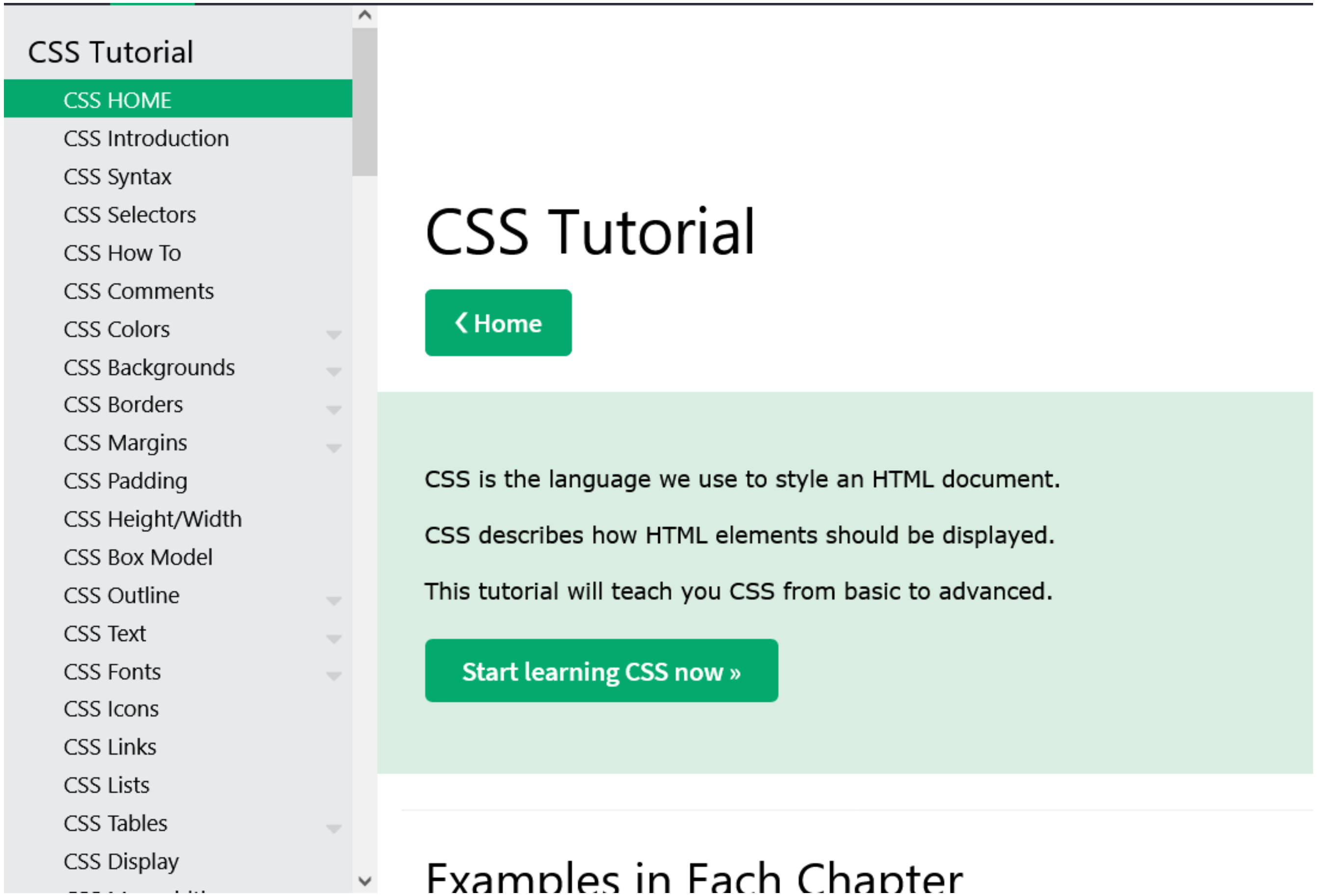
CSS box model

- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins. The image below illustrates the box model:



CSS cont…

- More examples on the https://www.w3schools.com/css/default.asp web page...



CSS Tutorial

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

This tutorial will teach you CSS from basic to advanced.

Examples in Each Chapter

CSS Task: Format your Web Page

- Create a new file called styles.css.
- In the styles.css file, write the following CSS rules to style your webpage:
  - Change the background color of the entire page.
  - Style the header (<h1>) to be centered and change its font color.
  - Add some padding and a border to the paragraph (<p>) element.
  - Add a border to the image and slightly reduce its size.
  - Style the unordered list (<ul>) so that the list items are displayed in a different font.
  - Change the color of the link (<a>) when you hover over it.

What is JavaScript?

- JavaScript is the programming language of the Web
  - developed by Brendan Eich in 1995
- JavaScript is a programming language initially designed to **interact with elements of web pages**
- Typically, you use JavaScript with HTML and CSS to enhance a web page's functionality, such as **validating forms, creating interactive maps, and displaying animated charts**
- When a web page is loaded, i.e., after HTML and CSS have been downloaded, the JavaScript engine in the web **browser executes the JavaScript code**
- Web browsers responsible for interpreting and executing JavaScript code
- **Client-side vs. Server-side JavaScript**
  - client-side: executed in web browsers
  - server side: executes on the server, e.g. Node.js

JavaScript HelloWorld

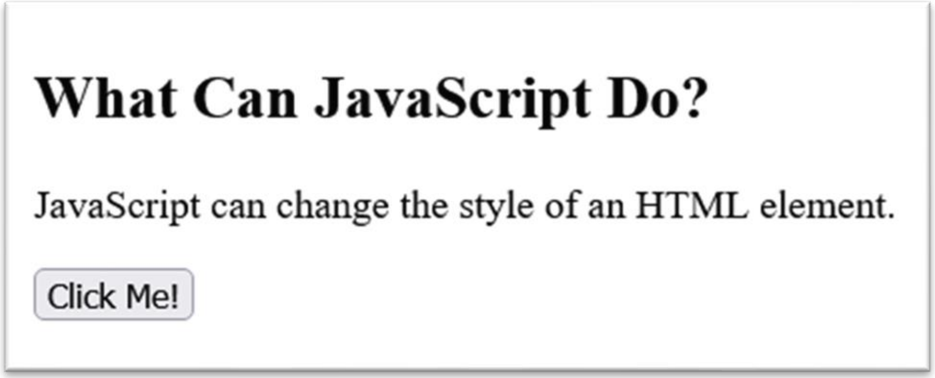
```
console.log("Hello, World!");
```

- `console.log`:
  - this is a method in JavaScript used to print messages to the web console
  - it is commonly used for debugging purposes to check the values of variables or the flow of execution in the code
- `"Hello, World!"`:
  - this is a string, a sequence of characters enclosed in double quotes
  - in this context, it is the message that will be printed to the console
- when the code console.log("Hello, World!"); is executed, it sends the string "Hello, World!" to the console

```
Hello, World!
```

JavaScript example

```
<!DOCTYPE html>
<html>
<body>
<h2>What Can JavaScript Do?</h2> <p id="demo">JavaScript can change the style of
an HTML element.</p>
 <button type="button„
onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>
</body>

</html>
```

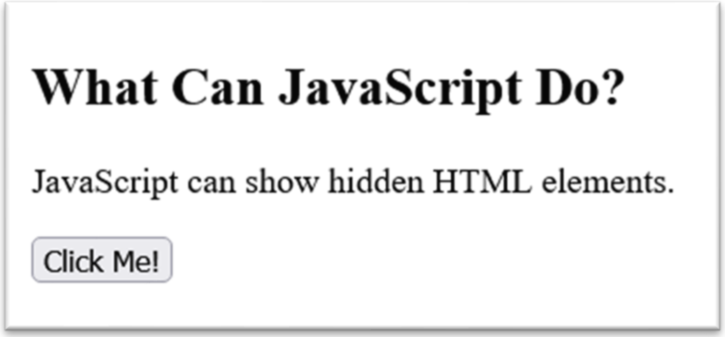


What Can JavaScript Do?

JavaScript can change the style of an HTML element.

JavaScript example

```
<!DOCTYPE html>
<html>
<body>
<h2>What Can JavaScript Do?</h2>
<p>JavaScript can show hidden HTML elements.</p>
<p id="demo" style="display:none">Hello JavaScript!</p>
<button
type="button„onclick="document.getElementById('demo').style.display='block'">Click
Me!</button>
</body>
 </html>
```



What Can JavaScript Do?

JavaScript can show hidden HTML elements.

JavaScript code where to

- In HTML, JavaScript code is inserted between <script> and </script> tags

```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

- External .js file (e.g. myScript.js)

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

  - Have to reference

```
<script src="myScript.js"></script>
```

```
<script src="https://www.w3schools.com/js/myScript.js"></script>
```

JavaScript Display Possibilities

- JavaScript can "display" data in different ways:
  - Writing into an HTML element, using innerHTML
  - Writing into the HTML output using document.write()
  - Writing into an alert box, using window.alert()
  - Writing into the browser console, using console.log()

JavaScript Display Possibilities

- JavaScript can "display" data in different ways:
  - Writing into an HTML element, using innerHTML

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

JavaScript Display Possibilities

- JavaScript can "display" data in different ways:
  - Writing into the HTML output using document.write()

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web
Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

JavaScript Display Possibilities

- JavaScript can "display" data in different ways:
  - Writing into an alert box, using window.alert()

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

JavaScript

JavaScript Display Possibilities

- JavaScript can "display" data in different ways:
  - Writing into the browser console, using console.log()

```
<!DOCTYPE html>
<html>
<body>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

JavaScript variables

- In a programming language, variables are used to store data values
- JavaScript uses the keywords var, let and const to declare variables
- All JavaScript identifiers are case sensitive
- In this example, x is defined as a variable. Then, x is assigned (given) the value 6:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Variables</h2>
<p>In this example, x is defined as a variable. Then, x is
assigned the value of 6:</p>
<p id="demo"></p>
<script>
let x;
x = 6;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

JavaScript variables

- **When to Use var, let, or const?**
  - Always **use const if the value should not be changed**
  - Only **use let if you can't use const**
  - Only **use var if you MUST support old browsers**
    - 1995-2015
  - Examples:

```
let x = 5;
const y = 6;

const price1 = 5;
const price2 = 6;
let total = price1 + price2;
```

JavaScript data types

- **8 data types:**
  - String
  - Number
  - Bigint
  - Boolean
  - Undefined
  - Null
  - Symbol
  - Object

```
// Numbers:
let length = 16;
let weight = 7.5;

// Strings:
let color = "Yellow";
let lastName = "Johnson";

// Booleans
let x = true;
let y = false;

// Object:
const person = {firstName:"John", lastName:"Doe"};

// Array object:
const cars = ["Saab", "Volvo", "BMW"];

// Date object:
const date = new Date("2022-03-25");
```

JavaScript comments

- Single line comments start with //
- Any text between // and the end of the line will be ignored by JavaScript (will not be executed)

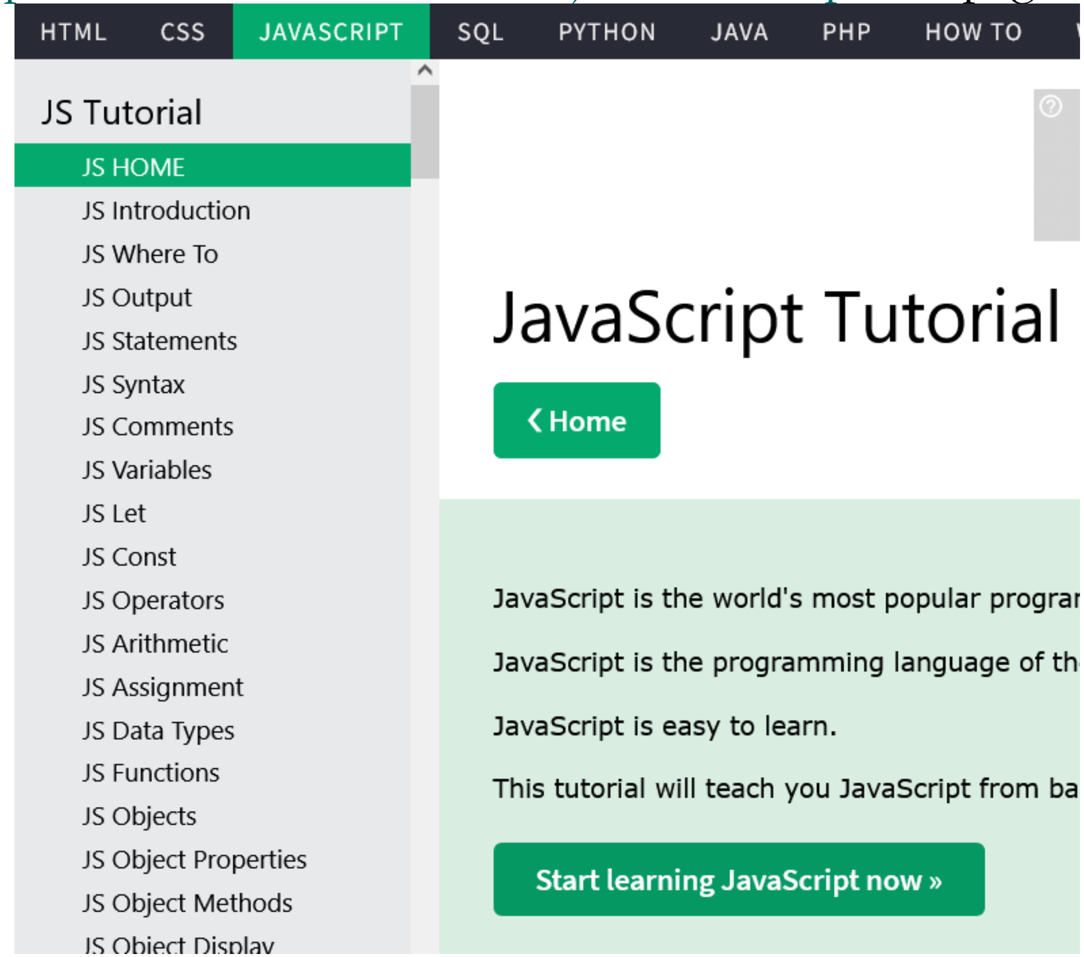
```
// Change heading:
document.getElementById("myH").innerHTML = "My First Page";

let x = 5;      // Declare x, give it the value of 5
```

- Multi-line Comments
  - multi-line comments start with /* and end with */

```
/*
The code below will change
the heading with id = "myH"
and the paragraph with id = "myP"
in my web page:
*/
document.getElementById("myH").innerHTML = "My First Page";
```

JavaScript cont…

- More examples on the https://www.w3schools.com/js/default.asp web page…



JavaScript Tutorial

JavaScript is the world's most popular prograr

JavaScript is the programming language of th

JavaScript is easy to learn.

This tutorial will teach you JavaScript from ba

JavaScript Task

- Add some basic interactivity to your webpage using JavaScript
  - External js file (script.js)
- You'll be creating a button that:
  - displays an alert message
  - changes the color of the header

```
function changeHeader() {
   alert("Hello, welcome to my webpage!");
   document.querySelector('h1').style.color = 'green’;
}
```

```
<button onclick="changeHeader()">Click Me!</button>
```

JavaScript Tasks

1. Write a function that receives two numbers as input and returns their sum
2. Write a function that decides if a number is even or odd
3. Write a function that calculates the sum of a list of numbers
4. Write a function that reverses a text (string)
5. Write a function that finds the maximum or minimum number in an array
6. Write a function that calculates the average of a list of numbers

JavaScript Tasks

1. Write a function that receives two numbers as input and returns their sum

```
function sum(a, b) {
    return a + b;
}

console.log(sum(3, 5)); // 8
console.log(sum(10, 15)); // 25
```

JavaScript Tasks

2. Write a function that decides if a number is even or odd

```
function isNumberEvenOrOdd (number) {
if (number % 2 === 0) {
    return number + " is even";
} else {
    return number  + " is odd";
}

console.log(isNumberEvenOrOdd(4)); // even
console.log(isNumberEvenOrOdd(7)); // odd
```

JavaScript Tasks

3. Write a function that calculates the sum of a list of numbers

```
function sumOfElementsInArray(array) {
    let sum= 0;

    for (let i = 0; i < array.length; i++) {
        sum+= array[i];
    }
return sum;



console.log(sumOfElementsInArray([1, 2, 3, 4])); // 10
console.log(sumOfElementsInArray([10, 20, 30])); // 60
```

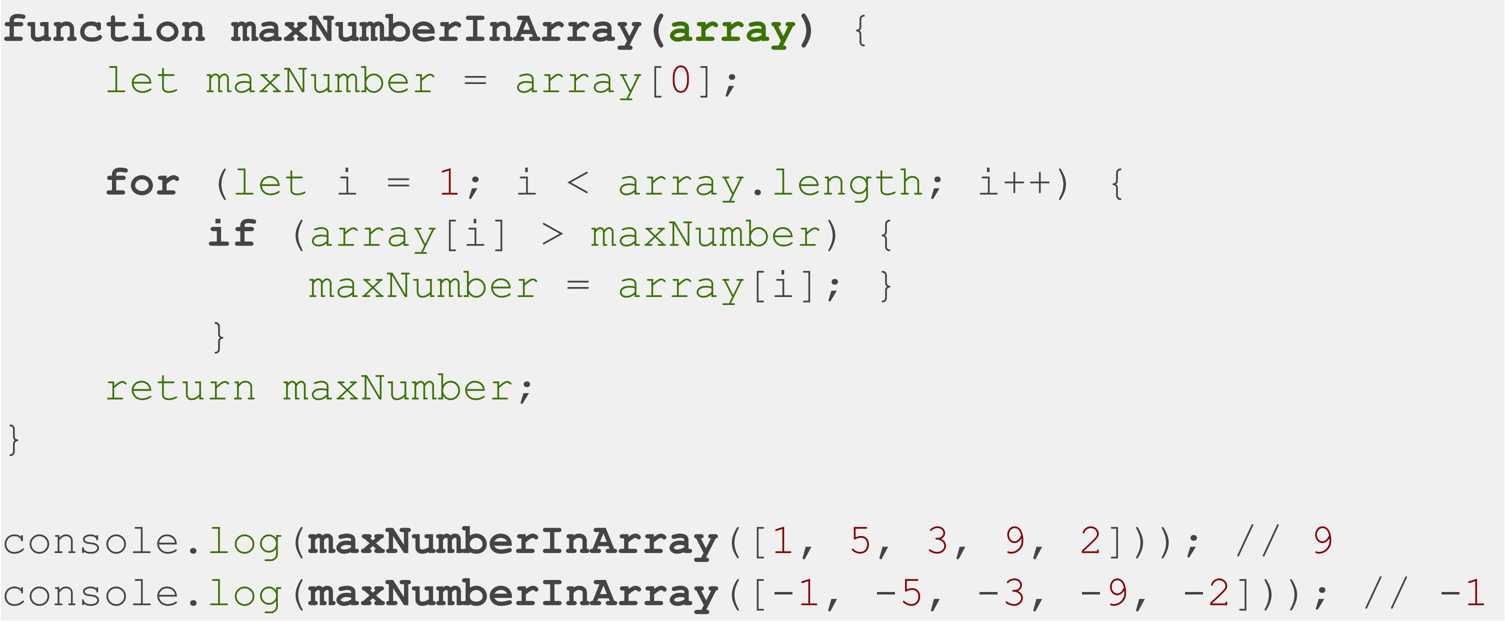
JavaScript Tasks

4. Write a function that reverses a text (string)

```
function textReversing(text) {
    return text.split("").reverse().join("");
}

console.log(textReversing("JavaScript")); // tpircSavaJ
console.log(textReversing("hello")); // olleh
```

JavaScript Tasks

5. Write a function that finds the maximum or minimum number in an array



```
function maxNumberInArray(array) {
    let maxNumber = array[0];

    for (let i = 1; i < array.length; i++) {
        if (array[i] > maxNumber) {
            maxNumber = array[i]; }
        }
    return maxNumber;
}

console.log(maxNumberInArray([1, 5, 3, 9, 2])); // 9
console.log(maxNumberInArray([-1, -5, -3, -9, -2])); // -1
```

JavaScript Tasks

6. Write a function that calculates the average of a list of numbers

```
function calculateAverage(array) {
    let sum = 0.00;
    let average = 0.00;

    for (let i = 0; i < array.length; i++) {
        sum += array[i];
    }

    average = sum / array.length;

    return average;
}
array1 = [1, 5, 3, 9, 2, 4.3];
console.log("The array is: " + array1);
console.log("The average of the elements is: " + calculateAverage(array1));
```

Thank you for your attention!

thank you ☺