

EiABC

Computer Programming

Part 2

Students may access the exercise files from the exercise file link from their class content site: [link](#)
Students are advised to use exercise files as per their instructor's directions. Examples of the exercise files are launched at these two sites: (1) <https://iupac22.github.io/education/>
(2) <https://iupac22.github.io/education1/>

Fundamentals of HTML and CSS

Contents

Fundamentals of HTML and CSS	1
Contents.....	1
List of tables:	2
List of figures:	2
1. Fundamentals of HTML	2
1.1. The Different Elements of an HTML	4
a) Tag Elements	4
b) Attributes:.....	5
c) Meta elements	5
d) Scape characters.....	5
e) Bold, strong, italic and em elements	5
f) Iframes:.....	6
g) Sups and subs	6
h) Inserting Audio and video files to your browser through html	6
Class Assignment 1: Example one: create a website with html by making use of only notepad:.....	7
Home assignment 1:.....	8
2. Fundamentals of CSS.....	8
2.1. Inline CSS:	8
2.2. Internal CSS:.....	8
2.3. External CSS:	9
3. FLEXBOX:.....	11

Home assignment 2:.....	12
-------------------------	----

List of tables:

Table 1: CSS and HTML Color Code.....	11
---------------------------------------	----

List of figures:

Figure 1, Basic HTML structure	4
Figure 2: Box Model in CSS, margin and padding	10
Figure 3: CSS selectors.....	10
Figure 4: Flexbox container	12

Fundamentals of HTML and CSS

The following lecture note and exercises are prepared by the readings of the instructor and references to Traversy Media.

This particular topic provides an opportunity for students to be creative on the processes and outcomes of their class and home assignments related to programing website with remote repository/ server.

1. Fundamentals of HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. In computer text processing, a markup language is a system for annotating a document in a way that it is syntactically distinguishable from the text (Wikipedia).

HTML can be described as the web version of a word document even if it has further applications, when integrated with other programing languages such as python. Therefore if you expertly know how to prepare a document by making use of Microsoft word, you have a better chance of understanding how HTML works. In fact since you will need to use Microsoft office word documentation during your study and career, you may use this opportunity to revise your skill on application of Microsoft word.

Even if you will be working on a project that has something to do with the worldwide web with other languages, it is important that you know how to work with/ understand HTML.

- HTML is a hypertext markup Language (not a Programming Language)
- HTML is a document that contains text
- Web browsers read and display documents that read HTML

When you are working on HTML, it is mostly like you are interpreting what happens on your regular word document. You need to work on HTML because you are planning to display your content on the web. Therefore you need to write your syntax with the basic requirements of a normal text structures such as: headers, lines, paragraphs, body, bold, italic, etc. In your syntax on HTML the following are important: All of the above components of texts are tags and will have < > (angle brackets). The html will have a structure where elements are placed as Root Elements and Nested Elements (chilled and body elements), for example: the body element will have paragraphs and the paragraphs will have lists, etc.

To work with HTML and CSS you can use browsers: (google chrome; Mozilla Firefox; safari; edge, etc.); Text editors: (sublime text; Ayto.io; Visual studio Code; Brackets; Notepad++, notepad, etc.)

Body of your text will be written as : <BODY> your message</BODY> (This is basically what a markup element is. The first tag in this syntax is an opening tag while the last one is closing tag with forward slash). We also have tags that do not encompass elements. These tags are called empty element like for example a break
. While almost other languages need heavy software for text editing and programming, HTML can be executed with only **notepad**. You can write your syntax in notepad; save it in .html extension and open it with any browser. Other text editors include: **notepad++**, **text wrangler**, **sublime text**, **visual studio** and **brackets (easy and better for beginners)**. You may download bracket from this link: <http://brackets.io/>; sublime from this link: <https://www.sublimetext.com/3> and Visual studio Code from this link: www.code.visualstudio.com. But if it is with in your interest, you may integrate your work process with Dreamweaver and other software as well. You can save the syntax you prepared in note pad with html extension and open the file in any web browser and reopen the html file in notepad for editing. See the following figure for your reference and discussion basic HTML structure:

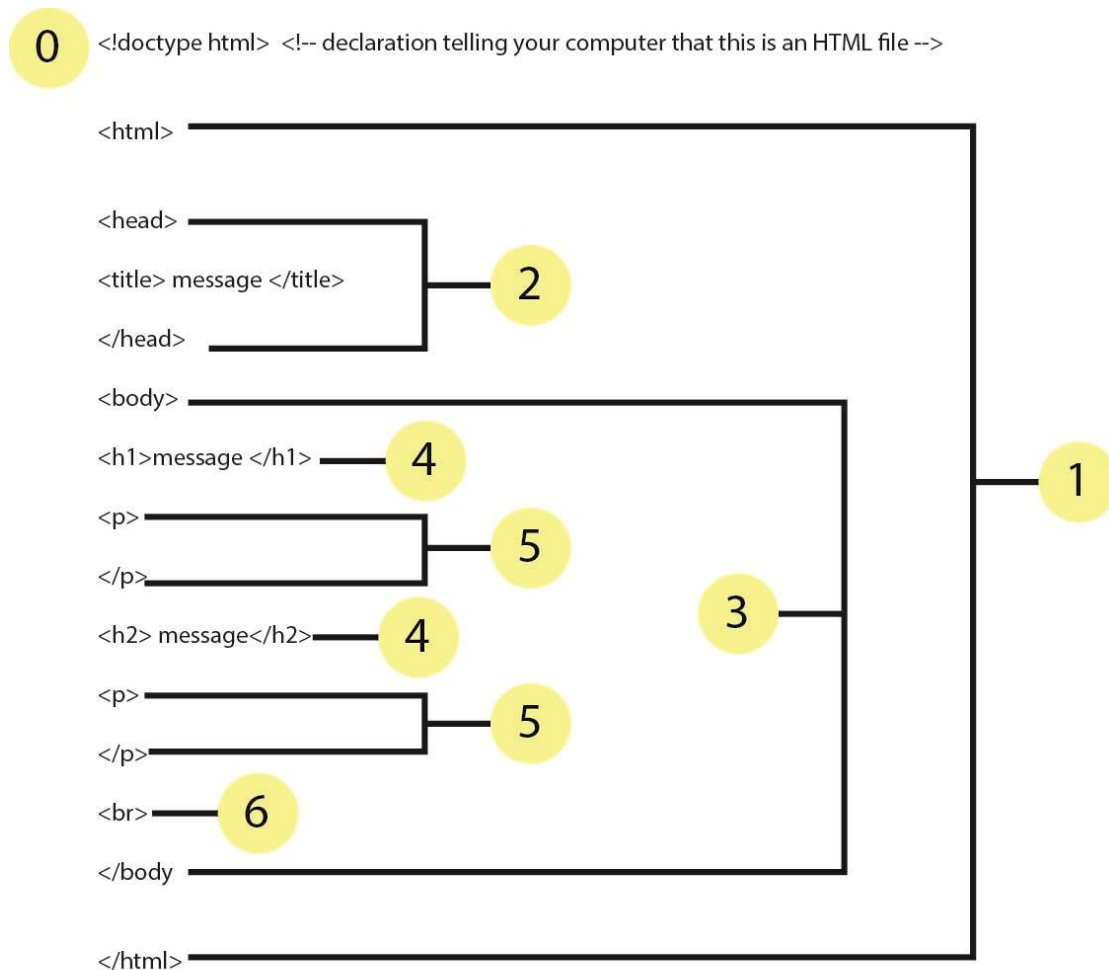


Figure 1, Basic HTML structure
Source: own

1.1. The Different Elements of an HTML

The different components of HTML are (1) Tag/ Tag Elements ;(2) Attributes (3) Meta Elements (4) scape characters; etc:

Main examples of elements of an html document are the following:

a) Tag Elements

`< >` `</ >` are opening and closing angle bracket consecutively. When they are tagged with paragraphs or headers they are named as tagged elements as follows:

```

<p> </p>
<h1> </h1>

```

<body> <body>

b) Attributes:

Some attributes are specific to elements like the label while some are valid for most elements like the class attribute. Attributes are highly important when we work with CSS, java script and ASP. ALL attributes have three unique syntaxes: example, class="test". All of the three components are placed in the opening tag only. See the following examples:

```
<p class=" test" > text</p>
<label for=" name" > name</label>
```

Other examples can be about forms:

```
<form action="mailto:professionalcad3@gmail.com"
method="post">
<p>Upload here any file you might think is supportive to our course. </p>
<input type="file" name="file">
<input type="submit" value="upload">
</form>
```

Here the attributes can be IDs or Classes the basic difference between the two are an **ID** can be used to identify one element, whereas a **class** can be used to identify more than one.

c) Meta elements

: are elements we put in html syntax for web search engines. They are not visible to the user. Meta elements do not need closing tags. See the following examples:

```
<meta name="description" content="This is a webpage for teaching Architecture students on basics of
setting up a website in HTML."/>
<meta name="keywords" content="architecture, html, design, computer aided design"/>
<meta http-equiv="author" content="cad chair"/>
<meta http-equiv="content-language" content="en-us"/>
```

d) Scape characters

: do not have opening or closing tag. See the following examples:

Space: ; TM: ™ ; &: & ; ©: © ; <: < ; >: > ; ? : ?

e) Bold, strong, italic and em elements

: These elements are simply placed in your html syntax simply with opening and closing tags. See the following examples: ; ; <i></i> and

f) Iframes:

:are used to bring other webpages in to your index page. The pages brought in to your index page can be your own secondary pages or other pages from other sites that are not yours such as advertisements. We can have more than one sites displayed in one frame by switching between them. See the following examples:

```
<iframe frameborder="1" width="number1" height="number2"
  src="name1.html" name="ouriframe" id="ouriframe"></iframe><br/>
<a href="name2.html" target="ouriframe">switch to new document</a>
```

g) Sups and subs

: are super scripts and sub scripts. The following are the syntaxes for both:

```
<sup></sup>; <sub></sub>
```

h) Inserting Audio and video files to your browser through html

:Supported formats may be: .mp3, .wav, .ogg, .mp4 and webm. See the following examples on how to add audio controls and video controls respectively. The syntaxes in red color are optional.

Example audio:

```
<audio controls autoplay loop ="audiocontrols">
<source src="mandela.mp3"/>
</audio>
```

Example Video:

```
<video width="640" height="360" controls>
  <source src="ethioplugin.mp4"/>
</video>
```

Further lists of more tags:

Declaration: <!doctype html> ; **Html:** <html>; **Main Heading:** <head>; **Title:** <title>; **Body tags comprise most of the components of the text:** <body>; **Headers:** <h1>; **Paragraphs:** <p>; **Lists:** , , ; **Tables:** <table>, <tr>, <td>, <th>; **Comments in html:** <!--text -->; **space between words:** , **space between lines:**
 ; **Links to External URL:** <p>text: text </p>; **Links to Internal Sections:** text ; text: </p>; **Inserting Image:** (here keep in mind that you do not have close tag for inserting images); **Image resizing:** (you may also reduce (change) image size on image editor software such as paint and Photoshop); **Dropdown list:** <select name="name" size="number" multiple="multiple"><option value="name1">name1</option> <option value="name2">name2</option><option value="name3">name2</option> <option

value="name4">name4</option> </select> (the syntax in blue is optional); **Forms:** <form action="" method=""> <input type="text" name=""> </form>; **Multiline text:** <textarea rows="number" cols="number"></textarea></br>; **Text submit:** <input type="Submit" value="Submit">;

The following syntax shows the basic structure of an html. It is placed here so you will use it for you next class assignment 1:

```
<!doctype html> <!-- declaration telling your computer that this is an HTML file -->
<html>
<head>
<title> message </title>
</head>
<body>
<h1>message </h1>
<p>
</p>
<h2> message</h2>
<p>
</p>
<br>
</body>
</html>
```

Web browsers can display different kinds of files. Not just HTMLs but also SVG files, XML files, etc.

Class Assignment 1: Example one: create a website with html by making use of only notepad:

Put the following texts in to the above indicated structure on notepad and save the notepad file with a format extension of: example1.htm and open the saved file with any browser you might have on your computer. And show your output to your instructor.

Main header: Ethiopian Institute of Architecture, Building Construction and City Development

Title: Professional Computer Aided design II

Paragraph One: In professional Computer Aided Design II Course students learn about three major topics: (1) implementation of Building Information Management that include documentation and analysis of an entire projects life cycle; (2) data visualization, graphics design and generative design through programing. (3) Students also learn about parametric and computation al design of buildings and components through visual programing.

Paragraph Two: In order to complete the tasks of the course students need to build an understanding of computation through proprietary software such as Revit architecture, rhinoceroses, etc. and programing with different languages such as HTML, JAVA, PHP, RUBY, etc.

Paragraph Three: The course is presented to students with different methods that include lectures, discussions, lab practices, workshops, assignments and exams. The following table shows the detail lectures and exams students take during the course:

Header 2: This is an exercise for students to work on their first website.

Paragraph Four: Students are expected to practice hard and show their skill on their other projects at school and elsewhere.

Comment: This tag is for placing comments in your text. It does not show up on the main display of the html on your browser.

Home assignment 1:

Please use the files given to you by your instructor or the files located at this link: ([link](#)) to study how to develop further the above example, **example1.htm. (example1, 2,3,4,5 and 6)**

All the files will get you through learning the following topics: (1) Basic structure (2) Tables and lists (3) Internal and external links (4) Images, forms and labels (single and multi-line text box), radio button, check box, drop down list (attributes) (5) Meta Elements, scape elements, Bold, strong, italic and em elements (6) iframes, super scripts, subscripts, audio and video controls.

2. Fundamentals of CSS

Cascading Style Sheets (CSS) is a style sheet not a programming but a styling language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript (Wikipedia).

CSS is more flexible. CSS is used more for layout design than just tagging text elements like in HTML. CSS used alongside HTML, XML, XHTML, SVG. Please refer further in to flexbox and animation in CSS. You can extend the functionalities of CSS by using technologies like Sass and Less to work on variables and conditionals. In your HTML code if you type anything between `<style></style>` tags it means the information is in CSS just like you would put an information between `<svg></svg>` would be in XML. We can code CSS in three ways (a) Inline CSS: Directly in the HTML element (NO!), (b) Internal CSS: using `<style>` tags within a single document and (c) Linking an external .css file. (Preferred method).

2.1. Inline CSS:

This option is executed in the following manner: This option is not recommended of use if the coder/designer has a knowledge of the other methods.

The following syntax should be placed with in the body of the html structure and inside any of the tags (this option as it mixes the HTML and the styling work, it is not advisable)

```
<h1 style="color:brown">message</h1>
```

2.2. Internal CSS:

This option is executed in the following manner:

The following syntax should be placed above body of the html structure and should be designed to stylize the body (layout) of the html structure.

```
<style type="text/css">
h1{color:brown;}
</style>
```

2.3. External CSS:

This option is executed by separating the CSS file and saving it in a similar directory as the main html file or elsewhere. To be referenced for styling of your layout work in your main html code. Out of the three, this one is most recommended since it separates the text work in html from the styling in CSS. Here there will be a separate file for styling that will be called in to the main html code. The list recommended option is '[2.1](#)'.

To work with option '[2.3](#)', we have to put the following syntax above the <body> of the main html code(structure) in order to link the html main code and the external CSS style file:

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

The following are examples of what we can put in our separate CSS file both for ID's and classes in CSS selectors. Once you work out all the task putting the following text in to your note pad or other text editor you have to save it as .css file.

```
body{background-color:#f4f4f4;color:#555555; font: normal 16px Arial, Helvetica, sans-serif; line-height: 1.6em; }
```

```
.container{
width: 80%;
margin: auto;}
.box-1{
background-color: #333;
margin: 0;
color: #fff;}
.box-2{
background-color: gray;
margin: 0;
color: #fff;}
```

Reference to styles in the main html code can look like the following examples:

```
<div class="container"></div> or <div class="box-1"></div> or ...
```

ID should be unique to a task in html/CSS while classes can be reused for further purposes or tasks. Example of ID can be: `<div id="side-bar"></div>` while example of a class can be `<div class="box-1"></div>`.

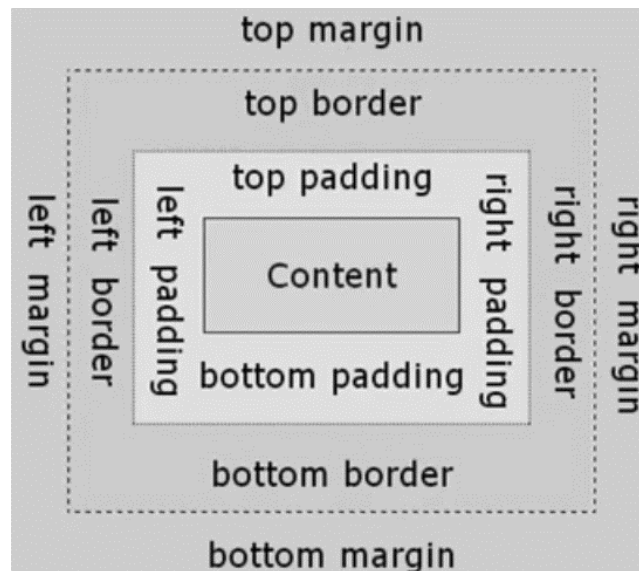


Figure 2: Box Model in CSS, margin and padding

Source: Traversy Media

There is a long way and short way of adding margins and paddings to tag of our html layout: (1) `p{margin-top: numberpx; margin-bottom: numberpx; margin-right: numberpx; margin-left: numberpx; }` and (2) `p{margin: numberpx numberpx numberpx numberpx; }` (3) `P{ margin: numberpx numberpx; }`. If you are working with padding or border, just replace the word margin with padding or border in the above syntaxes.

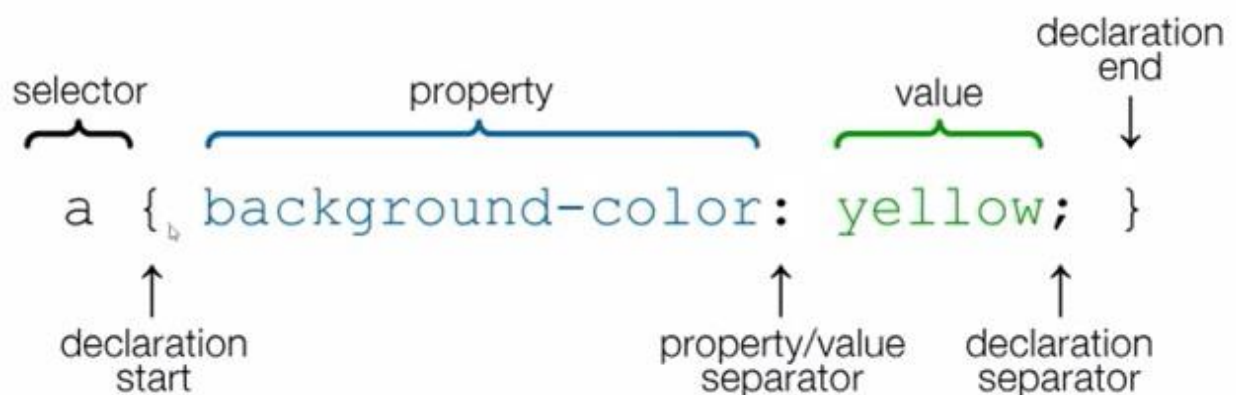


Figure 3: CSS selectors

Source: Traversy Media

To add colors to text and background, you can use the following options: (a) `body {color: red; background: yellow;}`, (b) `h1 {color:#00ff00;}`, (c) `p{color:rgb(0,0,255);}`. See the following link for reference to see different color values in Hexadecimal, RGB, HSV, HSL and CMYK: [Color picker](#). And [HTML and CSS color Cod](#). In CSS we use color references by color names; HTML5 color names; Hexadecimal and RGB.

Table 1: CSS and HTML Color Code

Source: <https://www.rapidtables.com/web/color/html-color-codes.html>

HTML / CSS Color Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
coral	#FF7F50	<code>rgb(255,127,80)</code>
tomato	#FF6347	<code>rgb(255,99,71)</code>
orangered	#FF4500	<code>rgb(255,69,0)</code>
gold	#FFD700	<code>rgb(255,215,0)</code>

In CSS, we can use **web-saved fonts** that we can find directly from within HTML file. If we want to use other special fonts, we have to download fonts like google font and save them as CSS file in to our directory and call them into our main html file. The following table **contains list of font examples**:

You can add comment that you do not need to be displayed in your browser by making use of the following syntaxes for CSS and HTML respectively:

In you CSS: `/* any message or ids or tags or .. */` and in your HTML coding: `<!-- Any text -->` .

See 'example7.html' file in the give exercise folder to study further: styling forms, floating blocks, boxes, anchors, buttons, etc. with further styling of margins, borders, color, paddings, list styling, etc.

Positioning in CSS: Static (is a default property. components get rendered in order of the document flow therefore statically positioned), Relative (here the element or component is placed relative to its normal position with properties such as top, right, left corner and we can also push it to any location), Absolute (allows us to target any position we want inside of a relative element), Fixed (an element will be places fixed to a location inside a browser window and scrolling or minimizing will not change the location of the element), Initial (sets the property of an element to its default value), Inherit (an element will inherit whatever the parent element has as a property).

3. FLEXBOX:

ELEXBOX: is short for flexible box model. It is a method to prepare a lay out of items with in a container in clear and flexible way. FLEXBOX is responsive and mobile friendly; positioning child is easier; Flex container's margins do not collapse with the margins of its content; order of elements can easily be changed without editing the source HTML; best fit element in the available space in the container by altering width and height of an element; flexbox is direction agnostic; built for small-scale layouts while "grid" specification if for more large scale layouts.

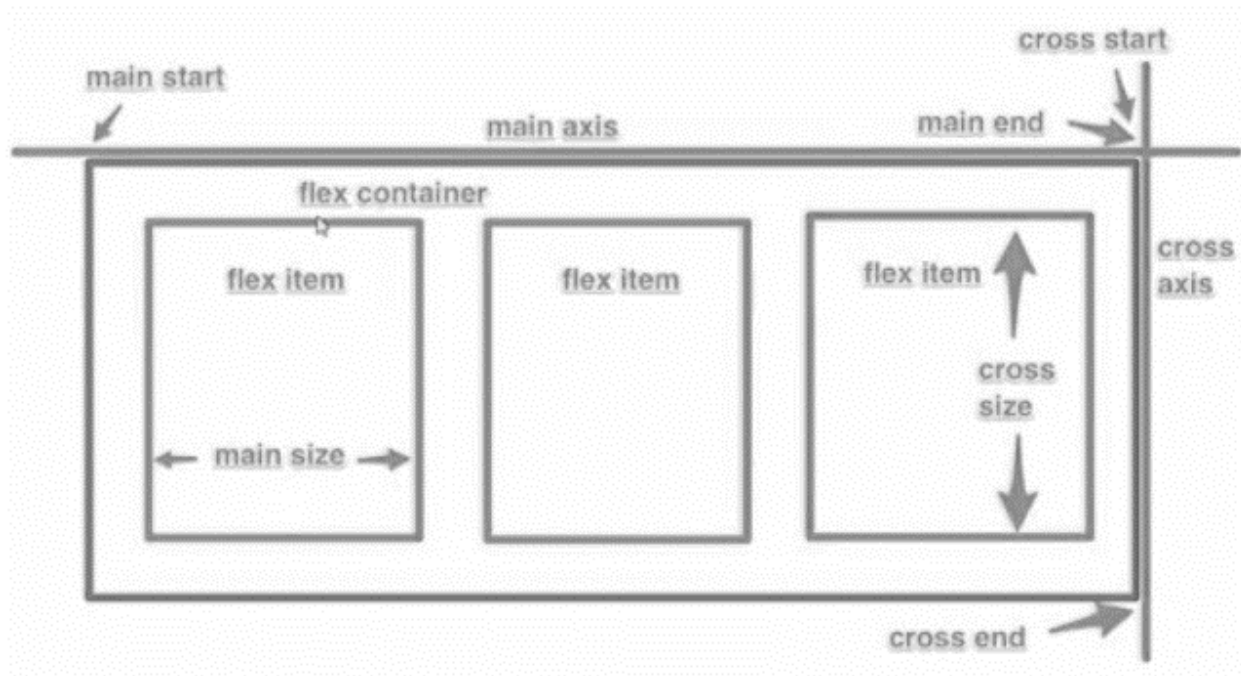


Figure 4: Flexbox container

Source: Traversy Media

The following are the main flex properties:

display: flex | inline-flex;
 flex-direction: row | column
 flex-wrap: wrap | nowrap | wrapreverse
 flex-basis:<length>
 justify-content: flex-start | flex-end | center
 align-self: flex-start | flex-end | center
 align-items: flex-start | flex-end | center
 align-content: flex-start | flex-end | center
 flex-grow:<number>;
 flex-shrink:<number>;
 flex:<integer>;
 order:<integer>;

See flexbox_exercise.html for practice on this topic. Study the three options presented on the exercise file and discuss the differences.

Home assignment 2:

Make use of the exercise file given by the name of example7.html as a base (starting) point and create your own site with HTML designed with your own custom CSS.

End of lecture note!