

# Udemy - Django 4 and Python Full-Stack Developer Masterclass

## Table of Contents

<b>Section 1-2: Overview .....</b>	<b>2</b>
<b>Section 3: HTML5 .....</b>	<b>2</b>
1. HTML – Basics .....	2
2. HTML – Lists.....	3
3. HTML – Divs and Spans .....	6
4. HTML Attributes – IMG and Anchor Tags .....	7
5. HTML – Tables .....	9
6. HTML Forms – Part One .....	9
7. Coding Exercise Check-in: HTML Forms - Part One.....	11
8. HTML Forms – Part Two .....	12
9. Coding Exercise Check-in: HTML Forms - Part Two.....	15
<b>Section 4: CSS .....</b>	<b>18</b>
10. CSS – Styling Basics and Color .....	18
11. CSS – Common Style Attributes .....	19
12. Coding Exercise Check-in: CSS - Common Style Attributes .....	21
13. CSS – Selectors – Classes and Ids.....	22
14. CSS – Fonts and Text .....	24
15. Coding Exercise Check-in: CSS - Fonts and Text.....	26
16. CSS – Box Models.....	27
17. Coding Exercise Check-in: CSS - Box Models.....	29
<b>Section 5: Bootstrap.....</b>	<b>30</b>
18. Introduction to Bootstrap .....	30
19. Bootstrap Buttons.....	31
20. Bootstrap Forms .....	33
21. Bootstrap Navbars .....	35
22. Bootstrap Grid System and Layout.....	37

## Section 1-2: Overview

### Overview:

User Browser → HTML/ CSS/ Bootstrap → Django → Database → Linode

### Front-End Section

- HTML
- CSS
- Bootstrap

### Back-End Section

- Python Basics
- Python Advanced
- Django Section
- Linode Deployment

### Student Types:

- **Totally New:** Start at HTML
- **Know Front-End:** Start at Python
- **Know only Python:** Start at HTML then possibly skip Python Basics
- **Know Front-End and Python:** Go to Django!

## Section 3: HTML5

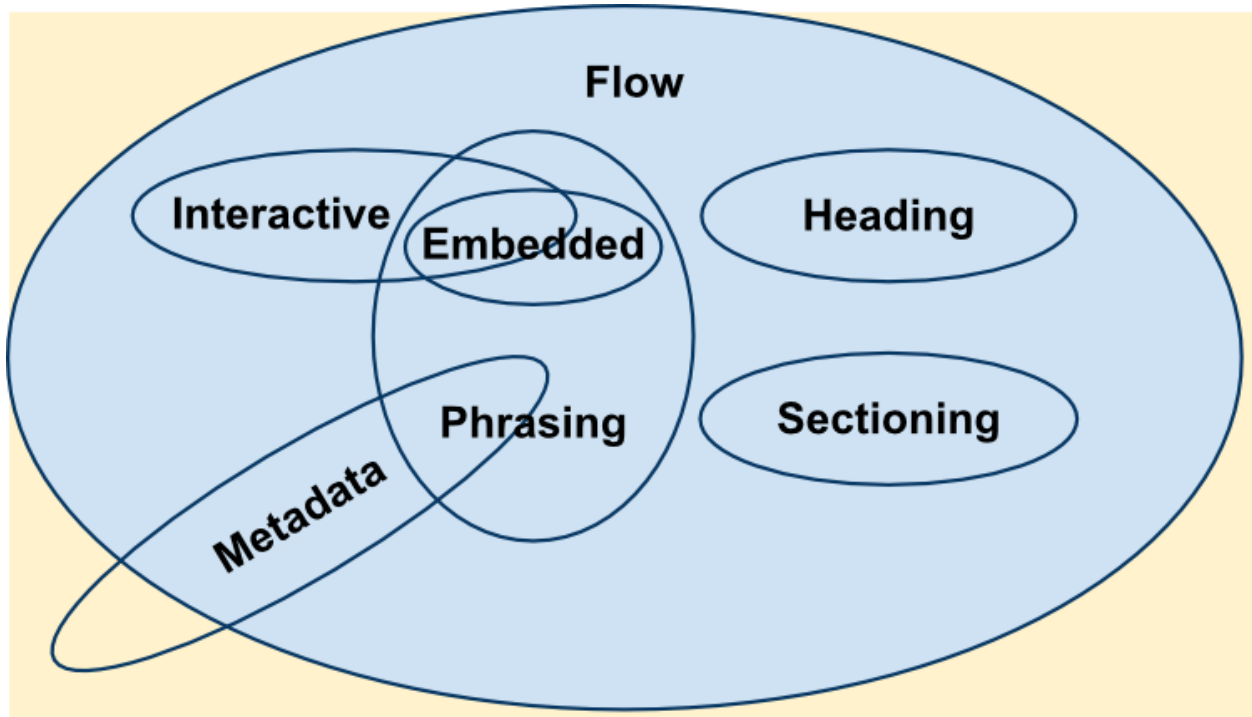
HTML = HyperText Markup Language

→ HTML just means it include metadata for annotating the document (webpage) which is visually distinguishable from how the user **sees** the document

### 1. HTML – Basics

Look up here:

[https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content\\_categories](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories)



On VSC:

Tipp: Copy path for adding a HTML file to the browser  
 Tipp: doc → stands for set up your HTML document

```
<html lang="en"> # English web page
<head>
  <meta charset="UTF-8"
  <meta....
  <title>Document</title>

  # More meta data about the page
  # Linking CSS file
  #

</head>
<body>
  <h1> I love HTML5</h1>
  <p> because it is cool!</p>
  # Actual Content

</body>
</html>
```

## 2. HTML – Lists

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  # unorded list
  <ul>
    <li> item one</li>
    <li> item two</li>
    <li> item three</li>
  </ul>

  # orded list
  <ol>
    <li> item one</li>
    <ul>
      <li>unorded list inside  orded list</li>
    </ul>
    <li> item two</li>
    <li> item three</li>
  </ol>
</body>
</html>
```

## Output:

# unorded list

- item one
- item two
- item three

# orded list

1. item one
  - unorded list inside orded list
2. item two

3. item three

### 3. HTML – Divs and Spans

**Division** in HTML can be created with `<div></div>` tags.

→ These will allow to separate out our HTML into different section, which will become useful later when we want to apply certain styles to certain section

**Spans** are similar but they are used to create inline containers.

**Spans** use the `<span></span>` tags

While it won't be entirely clear what the effect of these tags are beyond organization, once we know CSS, we can use these tags to apply styles to only certain elements on the webpage

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  # organize code into division section/container
  <div style="color:red">
    <p>sentence one</p>
    <p>sentence two</p>
  </div>
  <div>
    # span allow to make these little containers in line
    <p> sentence <span style="color:red"> three</span></p>
  </div>
</body>
</html>
```

**Output:**

# organize code into division section/container

sentence one

sentence two

# span allow to make these little containers in line

sentence three

## 4. HTML Attributes – IMG and Anchor Tags

Some HTML tags however need to have an attribute to function

- The Image Tag:  
<img src= "example.png"> (jpg, png)
- The Anchor Tag  
<a href=[www.google.com](http://www.google.com)>Text</a> (link to other webpages)

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  # Image tag<br>
  

  <h1>Example here</h1>
  # Anchor tag (provide reference to another file/website...)<br>
  <a href="http://www.google.com">CLICK HERE</a>

</body>
</html>
```

**Output:**

# Image tag  
**pic not found**

**Example here:**

# Anchor tag (provide reference to another file/website...)

[CLICK HERE](#)



## 5. HTML – Tables

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <table border="1">
    <thead>
      <th>Col 1</th>
      <th>Col 2</th>
      <th>Col 3</th>
    </thead>
    <tr>
      <td>entry 1</td>
      <td>data 2</td>
      <td>data 3</td>
    </tr>
  </table>
</body>
</html>
```

**Output:**

Col 1	Col 2	Col 3
entry 1	data 2	data 3

## 6. HTML Forms – Part One

### Hypertext Transfer Protocol (HTTP)

It is responsible for the communication (HTML Transfer) between a client and server

HTTP just means that the HTTP is secured through encryption

There are many methods available using HTTP, such as:

- GET method
  - o Request data from a source
- POST method
  - o Sends data to a server

A typical user interacting with webpage won't be aware of GET and POST but still use them often filling out a **form on a page**

For example, a google search is filling out a **form** with a single text field and a submit button

**Form** are one of the most important HTML elements we'll learn about since it's the main way we will receive information from the client using the website

Inside a HTML Form there are **input** tags and **lable** tags, where the attributes play a large role in determining how they work

The **label tag** dictates what the user can see on the Form page

The **input tag** dictates what the user can actually provide as information

The **id** attribute specifies a unique id for an HTML element. The value of the id attribute must be unique within the HTML document. The **id** attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific **id**.

Let's explore these concepts inside our code editor!

```
<form action="other.html">

<label for="fname">First Name:</label>
<input type="text" id="fname" name="fname"><br>

<label for="lname">Last Name</label>
<input type="text" id="lname" name="lname">
<br><br>
<input type="submit">

</form>
```

### Output:

First Name: [.....]  
Last Name: [.....]

[Submit]

```
<form action="other.html">

<label for="email">Email:</label>
<input type="email" id="email" name="email" placeholder="email goes here"><br>

<label for="pw">Password:</label>
<input type="password" id="pw" name="pw"><br>
<br><br>
<input type="submit" value="SUBMIT HERE">

</form>
```

### Output:

Email: [...email goes here...] (accept emails only [..@.de/com](mailto:..@.de/com) or any  
Password: [.....] (is encrypted)

[SUBMIT HERE]

## 7. Coding Exercise Check-in: HTML Forms - Part One

TASK: In this exercise you will build the order form for an online shop.  
Please implement the following forms entry fields:

1. Name
2. Surname
3. Address
4. E-Mail
5. Coupon
6. Submit input field

**Extra Task:** Please color the label word coupon in **red** using a span. Don't forget the submit button. As usual, please also create the HTML header

```
<html lang="en">
  <head>

</head>
<body>
  <form action="other.html">
```

```
<label for="name">1. Name:</label>
<input type="text" id="name" name="name"><br>

<label for="sname">2. Surname:</label>
<input type="text" id="sname" name="sname"><br>

<label for="address">3. Address:</label>
<input type="text" id="address" name="address"><br>

<label for="email">4. E-mail:</label>
<input type="email" id="email" name="email"><br>

<label for="coupon"><span style="color:red">5. Coupon:</span></label>
<input type="text" id="coupon" name="coupon"><br>

<input type="submit">

</form>
</body>
</html>

<!-- TASK: Create the form.
# MAKE SURE TO READ THE FULL INSTRUCTIONS ABOVE CAREFULLY, AS THE EVALUATION SCRIPT IS
VERY STRICT.
# Link to Solution: Solution: https://gist.github.com/b4shy/3a5862862e5f639d46a64957f5f03023
-->
```

## 8. HTML Forms – Part Two

There are many different types of input selection we can setup for the user.

→ Check out the MDN web docs: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>

**<radio> element**

```
<h1>Hotel Feedback</h1>

<form action="other.html">
  <p>Are you from Inside or Outside USA?</p>
  <label for="in">Inside: </label>
  <input type="radio" id="in" name="loc" value="in"><br>

  <label for="out">Outside: </label>
  <input type="radio" id="out" name="loc" value="out"><br>

  <input type="submit">

</form>
```

Output:

# Hotel Feedback

Are you from Inside or Outside USA?

Inside:

Outside:

Submit

**<select>**: The HTML Select element:

The **<select>** HTML element represents a control that provides a menu of options.

```
<h1>Hotel Feedback</h1>
<form action="other.html">
  <select name="stars" id="">
    <option value="Great">3</option>
    <option value="OK">2</option>
    <option value="Bad">1</option>
```

```
</select>
<br>
<input type="submit" value="SUBMIT">
</form>
```

Output:

# Hotel Feedback

Advanced select with multiple features:

```
<label>Please choose one or more pets:
<select name="pets" multiple size="4">
  <optgroup label="4-legged pets">
    <option value="dog">Dog</option>
    <option value="cat">Cat</option>
    <option value="hamster" disabled>Hamster</option>
  </optgroup>
  <optgroup label="Flying pets">
    <option value="parrot">Parrot</option>
    <option value="macaw">Macaw</option>
    <option value="albatross">Albatross</option>
  </optgroup>
</select>
</label>
```

Output:

Please choose one or more pets:

Cat  
Hamster  
**Flying pets**  
Parrot

## <textarea>: The Textarea element

The **<textarea>** HTML element represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.

```
<h1>Hotel Feedback</h1>
<form action="other.html">
  <p>Any other comments?</p>
  <textarea name="" id="" cols="30" rows="10"></textarea>

  <br>
  <input type="submit" value="SUBMIT">
</form>
```

Output:

# Hotel Feedback

Any other comments?

Not for today

SUBMIT

## 9. Coding Exercise Check-in: HTML Forms - Part Two

TASK: In this exercise you will extend the order form for the previously defined online shop.

Please implement the following forms:

1. Name
2. Surname
3. Address
4. E-Mail
5. Coupon
6. Create Account (Yes or No). Please use radioboxes for this. Make sure that you cannot select both at the same time!
7. Order Experience (1-5). Please use a select box for this
8. Comments.

Don't forget the submit button. As usual, please also create the HTML header.

```
<html lang="en">
  <head>
  </head>
  <body>
    <form action="other.html">
      <label for="name">1. Name:</label>
      <input type="text" id="name" name="name"><br>

      <label for="sname">2. Surname:</label>
      <input type="text" id="sname" name="sname"><br>

      <label for="address">3. Address:</label>
      <input type="text" id="adress" name="adress"><br>

      <label for="email">4. E-mail:</label>
      <input type="email" id="email" name="email"><br>

      <label for="coupon">5. Coupon:</label>
      <input type="text" id="coupon" name="coupon"><br>

      <p>Create Account</p>
```



```
<label for="yes">Yes</label>
<input type="radio" id="yes" name="account" values="yes"><br>
<label for="no">No</label>
<input type="radio" id="no" name="account" values="no">

<p>Order Experience (1-5)</p>
<select name="order">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
</select>

<p>Any Comments?</p>
<textarea name="" id="" cols="30" rows="10">Write something here</textarea>

<br>
<input type="submit">

</form>
</body>
</html>
```

**Output:**

1. Name:
2. Surname:
3. Address:
4. E-mail:
5. Coupon:

Create Account

- Yes
- No

Order Experience (1-5)

1

Any Comments?

Write something here

Submit

## Section 4: CSS

Cascading Style Sheets (CSS)

CSS is designed to separate out the styling of the HTML elements into a separate document.

- HTML elements have a style attribute, for example:
- `<p style="color:green">Text</p>`

Style = Attribute

Color:green = Style Property with Value

It's a lot more efficient to separate out these style property assignments to another CSS file and then link that CSS file to the HTML

This allows us to assign styles to multiple HTML elements at once using selectors

## 10. CSS – Styling Basics and Color

HTML:

```
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="master.css"> #relation to stylesheet, location reference href → master.css
  <title>Document</title>
</head>
<body>
  <h1 style="color:blue">Heading Here</h1>
  <h1>Heading One</h1>
  <h2>Heading Two</h2>
</body>
</html>
```

## CSS:

```
h1 {
  color: blue;
}

h2 {
  color:red;
  background-color: chartreuse;
}
```

## Output:

**Heading Here**  
**Heading One**

**Heading Two**

Google search: CSS Style properties

<https://www.w3schools.com/cssref/>

<https://developer.mozilla.org/en-US/docs/Web/CSS>

## 11. CSS – Common Style Attributes

## CSS Specificity

[https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity#how\\_is\\_specificity\\_calculated](https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity#how_is_specificity_calculated)

Keep at in mind, there are going to be certain calls that are going to overwrite other calls and typically the smaller scale is going to overwrite the larger scale

Span should overwrite div because we are focused on very specific portion, and that's actually where the word comes from specificity

### HTML:

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="master.css">
  <title>Document</title>
</head>
<body>
  <p>Outside a div</p>
  <div>
    <p>Inside the div</p>
    <p>Inside the div an this is a <span>SPAN</span> </p>
  </div>
</body>
</html>
```

### CSS:

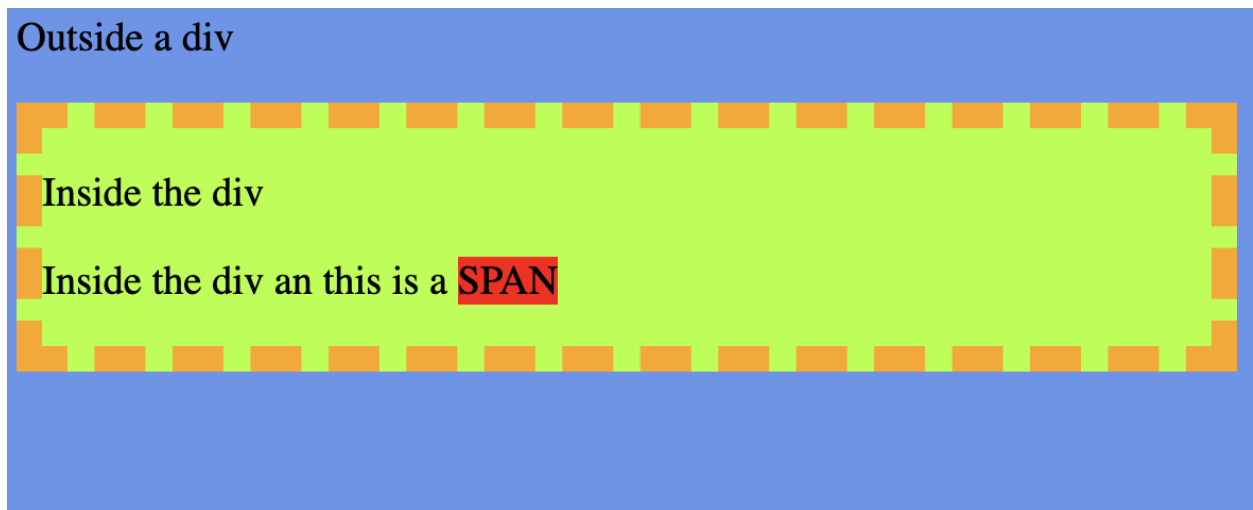
```
body {
  background-color: cornflowerblue;
}

div {
```

```
background-color: greenyellow;
border-color: orange;
border-width: 10px;
border-style: dashed;
}

span {
background-color: red;
}
```

### Output:



Div = large container  
Span = little container inside a div

## 12. Coding Exercise Check-in: CSS - Common Style Attributes

TASK: Fill the css file "styling.css" so that all div elements have:

1. Orange background
2. Blue text color
3. A dashed border with a width of 20

Second to that, make sure that all spans have red text color.

Then fill the HTML file "index.html" so that it contains at least 2 divs and one span element. You can choose any text you like!

As usual, please also create the HTML header. Don't forget to add the CSS file.

```
<!--
div{
  background-color: orange;
  color: blue;
  border-width: 20px;
  border-style: dashed;
}
span{
  color: red;
}
-->
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="styling.css">
  </head>
  <body>
    <div> One div element</div>
    <div> Another div element</div>

    <p> Some text with a <span>span</span> in it!</p>

  </body>
</html>
```

### 13. CSS – Selectors – Classes and Ids

We've seen that we can use **divs** and **spans** to apply styling to only certain elements, even if they share the same tag.

However, we can use selectors like **class** and **id** to have a lot more control and flexibility over this.

→ Imagine following HTML code:

```
<p>Outside Div</p>
<div>
  <p>Inside Div</p>
```

</div>

<p>Also Outside Div</p>

We can assign elements a **class** attribute with a specific name for CSS purposes. Classes are typically used within **divs** to group the styling of everything inside the div container together.

Classes can be repeatedly used across a webpage across elements.

IDs are an attribute also used to connect for styling, but should be completely unique and used for only a single element on the webpage.

Let's revisit our example...

→ Imagine following HTML code:

→ with unique IDs for <p> and class for <divs>

<p id="one"> Outside Div</p>

<div class="myclass">

    <p>Inside Div</p>

</div>

<p id="two">Also Outside Div</p>

In CSS id with #ID{...} and class myclass{...}

```
<!--
CSS file:
.myclass{
  color:red;
}

#one{
  color:blue;
}

#two{
  color:green;
}
-->

<html lang="en">
  <head>
    <meta charset="UTF-8">
```

```
<link rel="stylesheet" href="master.css">
</head>
<body>
  <p id="one">Above the div</p>
  <div class="myclass">
    <p>Inside the div</p>
    <p>also inside div</p>
  </div>
  <p id="two">Below the div</p>
  <di class="myclass">
    <p>Second div</p>
    <p>also in second div</p>
  </div>
</body>
</html>
```

### Output:

Above the div

Inside the div

also inside div

Below the div

Second div

also in second div

## 14. CSS – Fonts and Text

Google fonts → free for use

<https://fonts.google.com/specimen/Inspiration>

[https://developers.google.com/fonts/docs/getting\\_started](https://developers.google.com/fonts/docs/getting_started)

font-size:

Pixel (px)



Setting the font size in pixel values (**px**) is a good choice when you need pixel accuracy. A px value is static. This is an OS-independent and cross-browser way of literally telling the browsers to render the letters at exactly the number of pixels in height that you specified. The results may vary slightly across browsers, as they may use different algorithms to achieve a similar effect.

## Ems (em)

Using an em value creates a dynamic or computed font size (historically the em unit was derived from the width of a capital "M" in a given typeface.). The numeric value acts as a multiplier of the font-size property of the element on which it is used. Consider this example:

You should now see a drop shadow under the text:

*Making the Web Beautiful!*

And that's only the beginning of what you can do with the Fonts API and CSS.

## Overview

You can start using the Google Fonts API in just two steps:

1. Add a stylesheet link to request the desired web font(s):

```
<link rel="stylesheet"
      href="https://fonts.googleapis.com/css?family=Font+Name">
```

2. Style an element with the requested web font, either in a stylesheet:

```
.css-selector {
  font-family: 'Font Name', serif;
}
```

or with an inline style on the element itself:

```
<div style="font-family: 'Font Name', serif;">Your text</div>
```

## HTML:

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet"
      href="https://fonts.googleapis.com/css?family=Inspiration">
    <link rel="stylesheet" href="master.css">
  </head>
```

```
<body>

  <div class="stuff">
    <p>Text</p>
    <p>New text</p>

  </div>
</body>
</html>
```

## CSS:

```
body{
  font-family: Inspiration;
  font-size: 90%;
}

p{
  font-size: 3em;
}
```

## 15. Coding Exercise Check-in: CSS - Fonts and Text

### TASK:

Fill "styling.css" such that all h1 headings use **Arial** ,all paragraphs have **font size 1**.

Additionally create an id called **small** to reduce the font size to 0.5.

Fill "index.html", such that it contains at least one h1 heading, one paragraph and some text to which you apply **small** (please use a div container).

As usual, please also create the HTML header. Don't forget to add the CSS file.

```
<!-- h1{
  font-family: Arial;
}
```

```
p{
  font-size:1em;
}
#small{
  font-size:0.5em;
}
-->

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="styling.css">
  </head>
  <body>

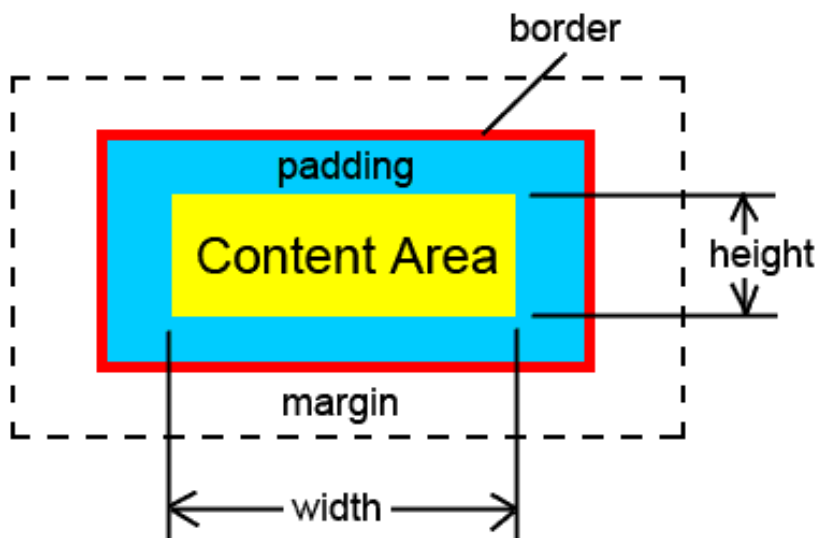
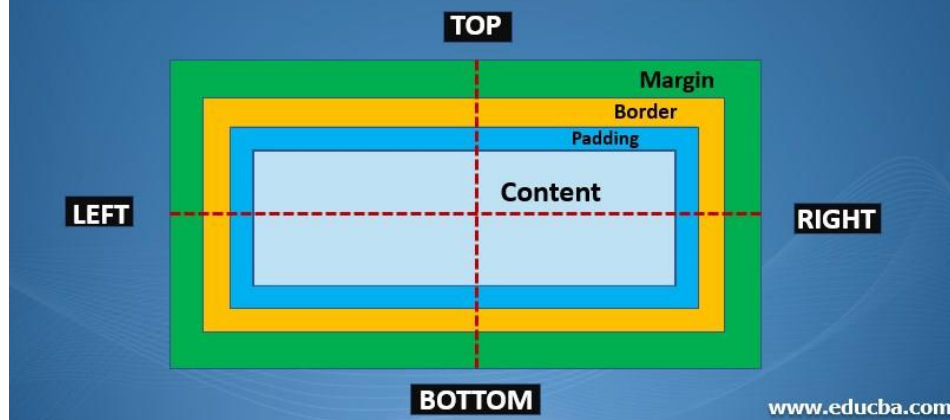
    <h1>Heading in Arial</h1>
    <p>Some paragraph</p>
    <div id="small">Some small text</div>

  </body>
</html>
```

## 16. CSS – Box Models

We've seen that using CSS we can specify a border around an HTML element  
→ This border is actually part of the **CSS Box Model** which is a group of edges surrounding element

# CSS Padding



Margin = transparent

## HTML:

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="master.css">
  </head>
  <body>
    <div id="up">UP</div>
    <div id="down">Down</div>
  </body>
</html>
```

```
</body>
</html>
```

## CSS:

```
#up{
  text-align: center;
  border: 10px solid blue;
  margin-left: 100px;
}

#down{
  text-align: center;
  border: 10px solid red;
  margin: 10px 3px 30px 100px; /* top: 10px margin */
                               /* right: 3px margin */
                               /* bottom: 30px margin */
                               /* left: 100px margin */
  padding-top: 100px;
}
```

## 17. Coding Exercise Check-in: CSS - Box Models

### TASK:

Use the box model to create a box which has a 5px wide solid green border and contains the centered word "CENTRAL".

The box should have a distance of 50 px to all page-borders (or to be more precise, to the div element).

As usual, please also create the HTML header. Don't forget to add the CSS file.

```
<!--
#central{
  text-align: center;
  border: 5px solid green;
```

```
margin: 50px;
}
-->

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="styling.css">
  </head>
  <body>
    <div id="central">CENTRAL</div>

  </body>
</html>

<!-- TASK: Fill the html and css files.
# MAKE SURE TO READ THE FULL INSTRUCTIONS ABOVE CAREFULLY, AS THE EVALUATION SCRIPT IS
VERY STRICT.
# Link to Solution: Solution: https://gist.github.com/b4shy/b22c0ca10eea7538d4cca3441e2d680e
-->
```

## Section 5: Bootstrap

### 18. Introduction to Bootstrap

**Bootstrap Framework** for quickly creating common front-end components that look good for our website

You can think of Bootstrap as a combination of a CSS file already created for your and ready to use arranged HTML components.

It also uses a grid system to easily resize your webpage for mobile, tablet, or desktop screens.

The documentation is designed to be used as a reference

<https://getbootstrap.com/>

## Example:

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <!-- CSS only -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeUOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  </head>
  <body>
    <form>
      <div class="mb-3">
        <label for="exampleInputEmail1" class="form-label">Email address</label>
        <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
        <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
      </div>
      <div class="mb-3">
        <label for="exampleInputPassword1" class="form-label">Password</label>
        <input type="password" class="form-control" id="exampleInputPassword1">
      </div>
      <div class="mb-3 form-check">
        <input type="checkbox" class="form-check-input" id="exampleCheck1">
        <label class="form-check-label" for="exampleCheck1">Check me out</label>
      </div>
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
  </body>
</html>
```

## 19. Bootstrap Buttons

Link together: CSS + JS + OWN CSS

```
<!-- CSS only -->
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>
<!-- Own CSS -->
<link rel="stylesheet" href="master.css">

```

**The best way to use bootstrap documentation is just to have fun with it**

## OVERWRITING CLASSES

BS Documentation → Copy HTML → Overwrite CSS classes by calling the classes on own CSS (It is possible because it was added after BS CSS link)

Example:

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- CSS only -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <!-- JavaScript Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
  <link rel="stylesheet" href="master.css">
  <title>Document</title>
</head>
<body>
  <button type="button" class="btn btn-primary">Primary</button>
  <button type="button" class="btn btn-secondary">Secondary</button>
  <button type="button" class="btn btn-success">Success</button>
  <button type="button" class="btn btn-danger">Danger</button>
  <button type="button" class="btn btn-warning">Warning</button>
  <button type="button" class="btn btn-info">Info</button>

```



```
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

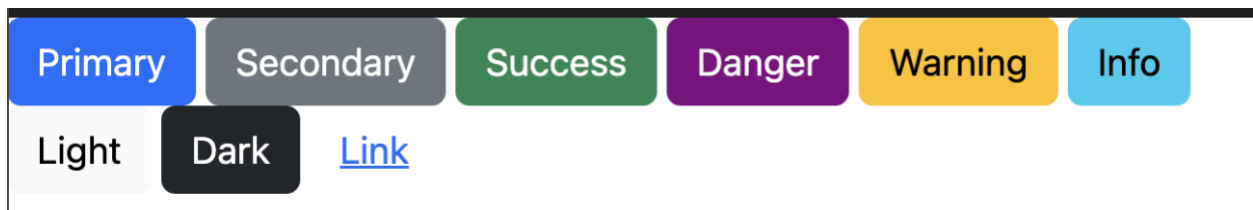
<button type="button" class="btn btn-link">Link</button>
</body>
</html>
```

## OWN CSS:

```
.btn-danger{
  background-color: purple;
  border-color: purple;
}

.btn-danger:hover{
  background-color: purple;
  color:red
}
```

## Output:



## 20. Bootstrap Forms

### Div's are utilized in forms:

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- CSS only -->
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <!-- JavaScript Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
  <link rel="stylesheet" href="master.css">
  <title>Document</title>
</head>
<body>
  <form>
    <div class="mb-3">
      <label for="exampleInputEmail1" class="form-label">Email address</label>
      <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
      <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
    </div>
    <div class="mb-3">
      <label for="exampleInputPassword1" class="form-label">Password</label>
      <input type="password" class="form-control" id="exampleInputPassword1">
    </div>
    <div class="mb-3 form-check">
      <input type="checkbox" class="form-check-input" id="exampleCheck1">
      <label class="form-check-label" for="exampleCheck1">Check me out</label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
</body>
</html>

```

**Output:**

Email address

We'll never share your email with anyone else.

Password

Check me out

## 21. Bootstrap Navbars

### Navigation Bar

Collapsing → change the size by zooming and so on

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- CSS only -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <!-- JavaScript Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
  <link rel="stylesheet" href="master.css">
  <title>Document</title>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
```

```
</button>
<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link active" aria-current="page" href="#">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Features</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Pricing</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled">Disabled</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="other.html">Other HTML</a>
    </li>
  </ul>
</div>
</div>
</nav>
</body>
</html>
```

**Output:**

## Navbar

Home

Features

Pricing

Disabled

Other HTML



## 22. Bootstrap Grid System and Layout

Bootstrap uses a Layout convention based on containers and a grid system

**These allow you to organise your HTML components based on rows and columns as well as easily auto-resize your web page for smaller or larger screens**

Typically, everything goes a container class (class="container") and from there is further separated out into rows and columns

Using the grid system you can specifically dictate how to resize for screens or let Bootstrap auto-resize for you

- What happens on different screen sizes?



- We can use any combination of numbers that will eventually add up to 12 columns



The grid system call will make use of the class="row"

Inside of a row class, we then have the following format:

- col-ScreenSize-NumberOfColumns
- Example: col-md-6

### Bootstrap: Containers and Grid are importing

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- CSS only -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTR"
crossorigin="anonymous">
  <!-- JavaScript Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
  <link rel="stylesheet" href="master.css">
  <title>Document</title>
</head>
<body>
  <div class="container text-center">
```

```
<div class="row">
  <div class="col-sm-2 extra">
    2/12
  </div>
  <div class="col-sm-10 extra">
    10/12
  </div>
</div>
<div class="row">
  <div class="col extra">
    1 of 3
  </div>
  <div class="col extra">
    2 of 3
  </div>
  <div class="col extra">
    3 of 3
  </div>
</div>
</div>
</body>
</html>
```

## Output:

2/12	10/12	
1 of 3	2 of 3	3 of 3