CS50 Web Programming with Python & Javascript 2020

(Harvard University)

0

HTML & CSS

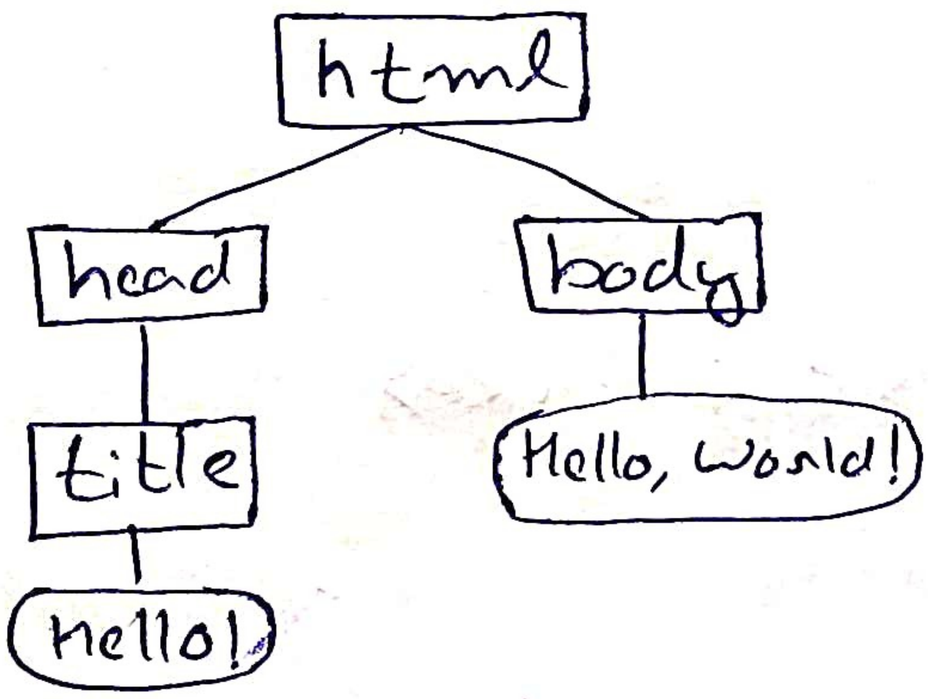
(HTML5 CSS3)
- CSS3 → Style of a web page
- HTML5 → Structure of a web page

★ HTML (Hyper text Markup Language)

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Hello!</title>
    </head>
    <body>
        Hello, world!
    </body>
</html>
```

- <!DOCTYPE html> → Doctype declaration
  - → Its a way of telling the web browser what version of html we are using
  - → This line indicate HTML5
- lang="en" → An attribute of tag html

★ Document Object Model (DOM)



★ Headings

<h1> ↑ Biggest heading
<h3>
⋮
<h6> ↓ Smallest heading

★ Lists

→ Ordered List

```
<ol>
    <li> First </li>
    <li> Second </li>
</ol>
```

⟶

1. First
2. Second

→ Unordered List

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

⟶

- First
- Second

★ Images

```
<img src="cat.jpg" alt="Cat">
```

↳ There is no closing tag.

⇒ Additional attribute:

width = "300"

★ link

```
<a href="https://google.com"> Click Here </a>
```

↓

Image.html

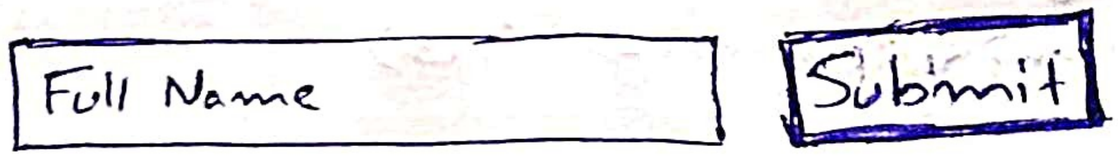
* Table

```
<table>
    <thead>
    <tr> <th> Ocean </th>
        <th> Average Depth </th>
        <th> Maximum Depth </th> </tr>
    </thead>
    <tbody>
        <tr>
            <td> Pacific Ocean </td>
            <td> 4,280 m </td>
            <td> 10,911 m </td>
        </tr>
        <tr>
            <td> Atlantic Ocean </td>
            <td> 3,646 m </td>
            <td> 8,486 m </td>
        </tr>
    </tbody>
</table>
```

Ocean	Average Depth	Maximum Depth
Pacific Ocean	4,280 m	10,911 m
Atlantic Ocean	3,646 m	8,486 m

★ Form

```
<form>
    <input type="text" placeholder="Full Name" name="name">
    <input type="Submit">
</form>
```



★ CSS (Cascading Style Sheets)

```
<h1 Style="Color: blue; text-align: center;">
  "Welcome to my web page!
</h1>
```

{Inline Styling}

⇒ Html element can inherit style information from there Parent.

or

```
<head>
    <style>
      h1 {
        Color: blue;
        text-align: center;
      }
    </style>
</head>
```

or

Styles.css

```
h1 {
    Color: blue;
    text-align: Center;
}
```

&

```
<head>
  <link rel="stylesheet"
        href="styles.css">
</head>
```

★ <u> tag</u>

{division of the page}

★ <u>Selector CSS</u>

```
td, th {
    border: 1px solid black;
}
```

* {Selecting multiple tags}

```
<h1 id="foo"> Heading 1 </h1>
```

```
#foo {
    Colour: blue;
}
```

* {Selecting via id}

⇒ Id should always be unique!

```
<h1 class="baz"> Heading 1 </h1>
```

```
.baz {
    Colour: blue;
}
```

* {Selecting via class}

⇒ Class may not be unique.

★ <u>Identifying Elements</u>

* div
* Span
* id
* class

★ Specificity

1. Inline
2. id
3. class
4. type

↑

★ Other CSS Selectors

a, b	——	Multiple Element Selector
a b	——	Descendent Selector
a > b	——	Child Selector
a + b	——	Adjacent Sibling Selector
c[a="b"]	——	Attribute Selector
a:b	——	Pseudoclass Selector
a::b	——	Pseudoelement Selector

★ Hover

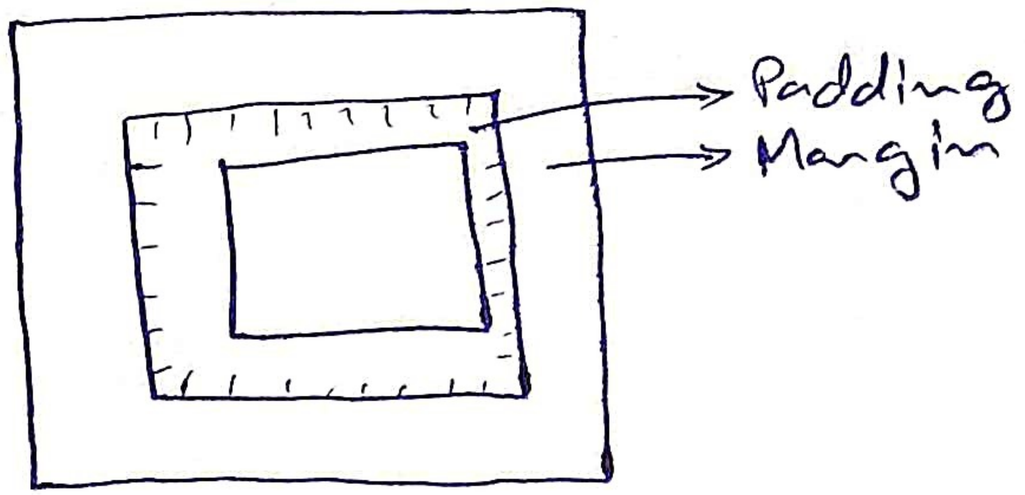
```
<button> Click Me </button>

button:hover{
    background-color: orange;
}
```

→ Pseudo class

CSS Styles

* Colors
* text-align
* background-Colors
* Width
* height
* Padding
* margin
* font-family
* font-Size
* font-weight {bold}
* borders
* borders-collapse



★ Responsive Design

- view port
- Media Queries
- Flex box
- Grids

→ Visual part of the screen that the user can actually see.

* View port

```
<meta name="Viewport" content="width=device-width,
      initial-scale=1.0">
```

* Media Queries

```
@media (min-width: 600 px){
    body {
        background-color: red;
    }
}
@media (max-width: 599 px){
    body{
        background-color: blue;
    }
}
```

* Flex box

```
<div id="Container">
    <div> First </div>
    <div> Second </div>
    <div> Third </div>
</div>
```

```
#Container {
    display: flex;
    flex-wrap: wrap;
}
```

* Grid

```
<div id="grid">
    <div> 1 </div>
    <div> 2 </div>
    <div> 3 </div>
</div>
```

```
#grid{
    display: grid;
    grid-column-gap : 20 px;
    grid-row-gap : 10px;
    grid-template-colums : 200px 200px auto;
}
```

⇒ There exists a lot of libraries, that do a lot of this for us.

→ Some people have already written CSS code to make our text look good, to make our buttons look good etc.

→ One of them is **Bootstrap**.

★ Bootstrap

⇒ To get started with bootstrap, all you have to do is to copy the bootstrap link to the top of your file.

★ Variables

Sass → Syntactically awesome style sheets

↳ It is a language that is essentially an extention to CSS.

Variable.scss

```
$ Colors: red;

ul {
    font-size: 14px
    colors: $colors
}
```

⇒ Our browser cannot understand *.scss file directly, we need to Compile our *.scss file to *.css file.

Sass Variables.scss : Variables.css

⇒ Sass --watch variables.scss : Variables.css

★ Nesting in Sass

```
div {
    font-size: 18px;
    P {
      Colour: blue;
    }
    ul {
      Colour: green;
    }
}
```

★ Inheritance in Sass

```
%message {
    ---
}

.Success {
  @extend %message
  Background-colour: green
}
```

Python

★ Decorators

⇒ Decorator is a function that takes a function as input and returns a modified version of that function as output.

⇒ This is known as a functional programming paradigm.

Example

```
def announce(f):          ← {decorator difinition}
    def wrapper:
        Print("About to run the function")
        f()
        Print("Done with the function")
    return wrapper

@announce                 ← {Adding decorator}
def hello():
    Print("Hello, world!")
```

★ lambda

```
lambda person: person["name"]
```

★ Exceptions

```
try:
    result = x/y
except ZeroDivisionError:
    Print("Error: Cannot divide by 0")
```

3
Django

⇒ It is a python web framework, which allow us to write Python code, that is able to dynamicdlly generate HTML & CSS.
  ↳ Ultimally allowing us to build dynamic web application.

HTTP → Hyper Text Transfer Protocol

GET / HTTP/1.1
Host: www.exaple.com
...

request

HTTP/1.1 200 OK
Content-Type: text/html
...

response

* django-admin startproject <Project Name>
  → { To Start & initialize a new Project }
* Python manage.py runserver
  → { For starting Server }
* Python manag.py startapp hello
  → { To start & initialize a new app }
* Python manage.py migrate

4
SQL, Models & Migrations

SQL {Structured Query Language}

↳ Language to interact with databases.

↳ Designed to managing data held in a related database management system. (RDBMS)

⇒ Different database management Systems:

→ MySQL

→ PostgreSQL

→ SQLite

★ SQLite Types

* Text
* Numeric
* Integer
* Real
* Blob {binary}

★ CREATE TABLE in SQLite

```
CREATE TABLE flights(
id INTEGER PRIMARY KEY AUTOINCREMENT,
origin TEXT NOT NULL,
destination TEXT NOT NULL,
duration INTEGER NOT NULL
)
```

★ Constraints

→ CHECK
→ DEFAULT
→ NOT NULL
→ PRIMARY KEY
→ UNIQUE

★ INSERT

```
INSERT INTO flights
    (origin, destination, duration)
    VALUES ("New York", "London", 415);
```

★ SELECT

```
SELECT * FROM flights;
SELECT origin, destination FROM flights;
SELECT * FROM flights WHERE id=3;
SELECT * FROM flights WHERE origin="New York";
SELECT * FROM flights WHERE origin IN ("New York"
                                      ,"Lima")
```

★ UPDATE

```
UPDATE flights
    SET duration = 430
    WHERE origin = "New York"
    AND destination = "London";
```

★ DELETE

```
DELETE FROM flights WHERE destination = "Tokyo";
```

★ Other Clauses

* LIMIT
* ORDER BY
* GROUP BY
* HAVING

★ JOIN

```
SELECT first, origin, destination FROM flights
JOIN passengers on passengers.flight_id = flight.id;
```

→ JOIN / INNER JOIN
→ LEFT OUTER JOIN
→ RIGHT OUTER JOIN
→ FULL OUTER JOIN

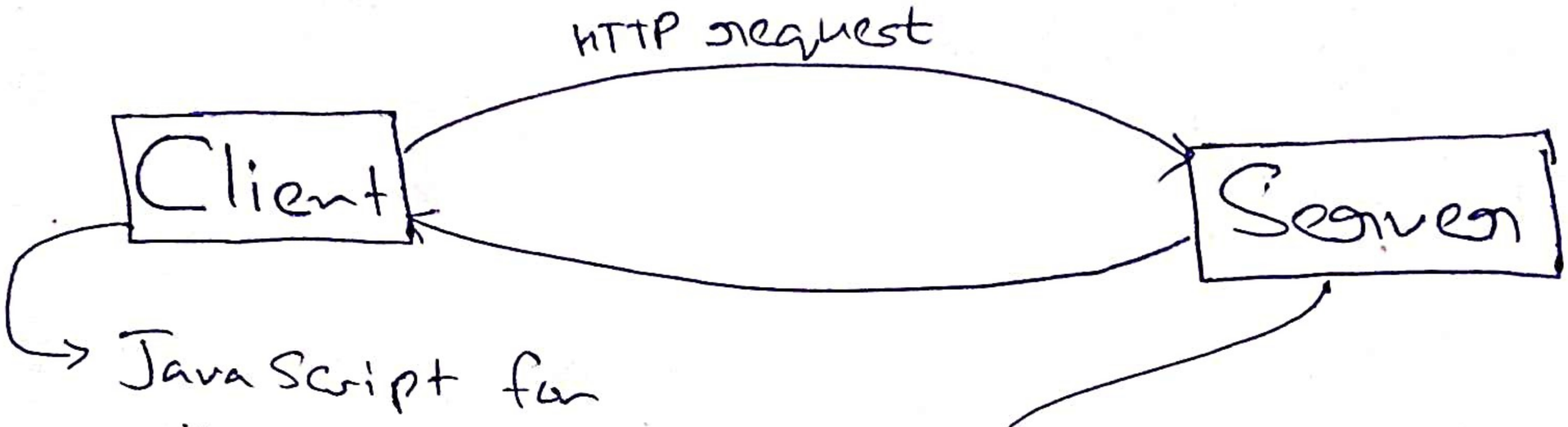
★ Django Models

```
from django.db import models

class Flight (models.Model):
    origin = models.CharField (max_length=64)
    destination = models.CharField (max_length=64)
    duration = models.IntegerField()
```

1:12:48

5
JavaScript



<Script>
---Java Script Code---
</Script>

alert ("Hello, world!");

⇒ One big area where JavaScript can be very powerful is with event driven Programming.

* function

```
function hell ( ){
  alert ('Hello, world!');
}
```

* Varible

```
let counter = 0;
Const Varible = 5;
```

alert ('Cont is now. ${counter}');