

# CYDEO

## HTML CLASS

---

# What will we cover today?

- Introduction to Front End
- Introduction to HTML
- Intermediate HTML
- Introduction to CSS
- Create a simple website
- Give some simple style to our website

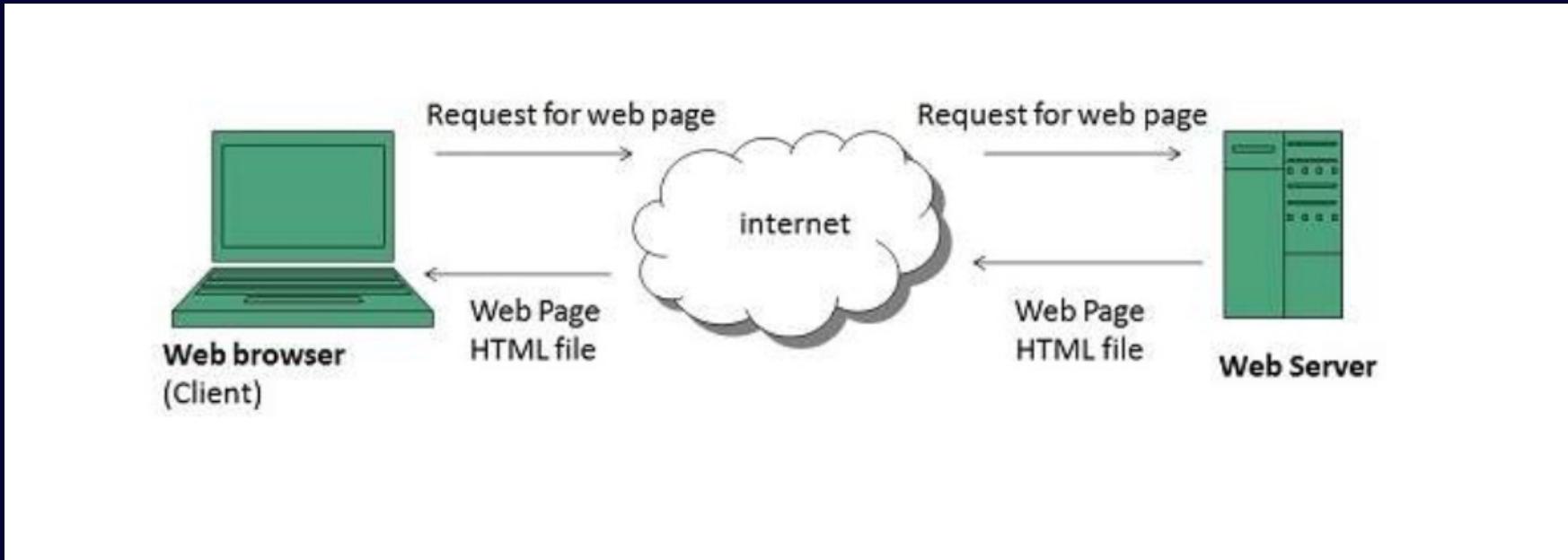
# What is the goal of the HTML class?

- To understand why HTML, CSS, JavaScript is used.
- To be able to read and understand high level HTML structure.

## **WHY do we learn HTML as testers?**

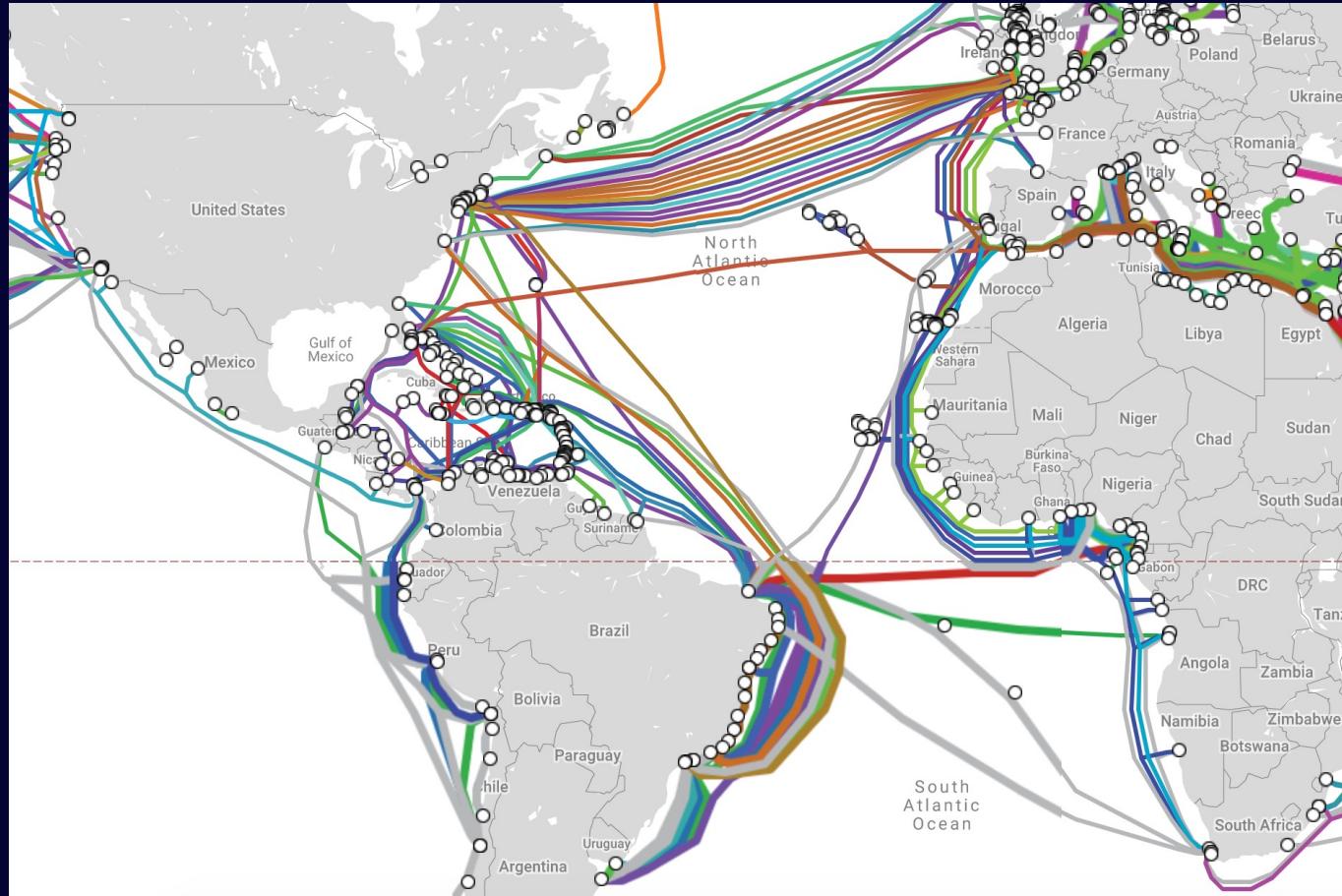
- To understand why HTML, CSS, JavaScript is used.
- To be able to read and understand high level HTML structure.

# How the internet works?



- A Packet's Tale: What is the internet, and how it works.

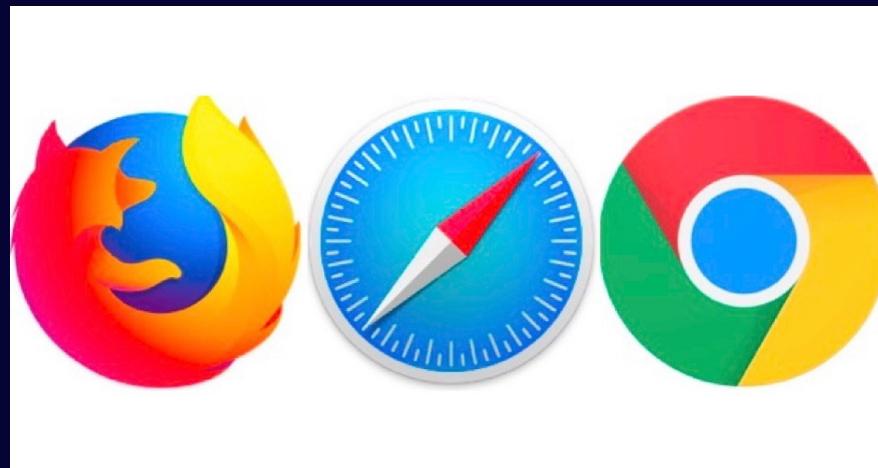
# Computers connected to each other with physical wires



# Most important take-aways

- We request something from the server.
- And server responding, and giving our data back to us.
- REQUEST >>>>> Request server figures out exactly what we are asking
- The server builds us the right content, often pulling information from database
- RESPONSE <<<<< The server responds with any combination of HTML, CSS, and JavaScript

# What is the purpose of the web browsers?



- The only purpose of the web browsers is to interpret/translate your files that should be displaying your web site.
- So basically it translates and displays the HTML+CSS+JS in a way that people can understand and use.

# HTML + CSS + JavaScript

```
<html itemscope="itemtype="http://schema.org/WebPage" lang="en"><head><meta content="Search the world's information, including webpages, images, videos and more."><meta content="Google has many special features to help you find exactly what you're looking for." name="description"><meta content="https://www.google.com/search?hl=en&q=*&tbo=q&tbs=s+&sa=X&source=hp" href="https://www.google.com/search?hl=en&q=*&tbo=q&tbs=s+&sa=X&source=hp" data-bbox="107 168 904 880"/>
```



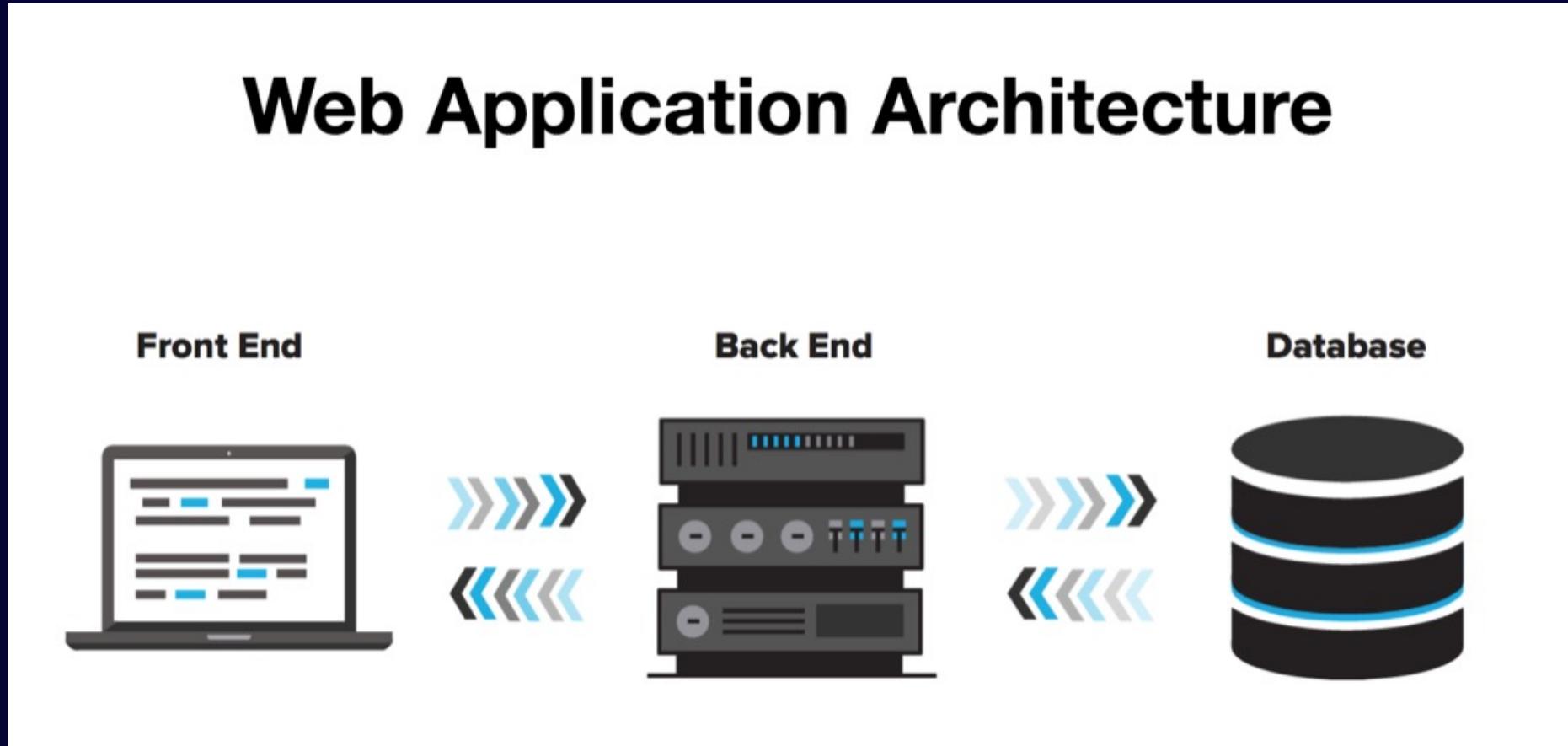
# Web Browser



# This same process happens for every single web application you visit

- The code that is translated by the web browser is called the page source.
- There is a way for us to see the code behind every web app.
- Let's view page source!

Difference between “front-end”, “backend” and “database”?



# What is front-end on a web app?

- The Front End is the stuff you see and interact with.
- Front-end is usually created using HTML, CSS and JavaScript



<http://skillcrush.com/2012/04/17/frontend-vs-backend-3/>

# What is back-end?

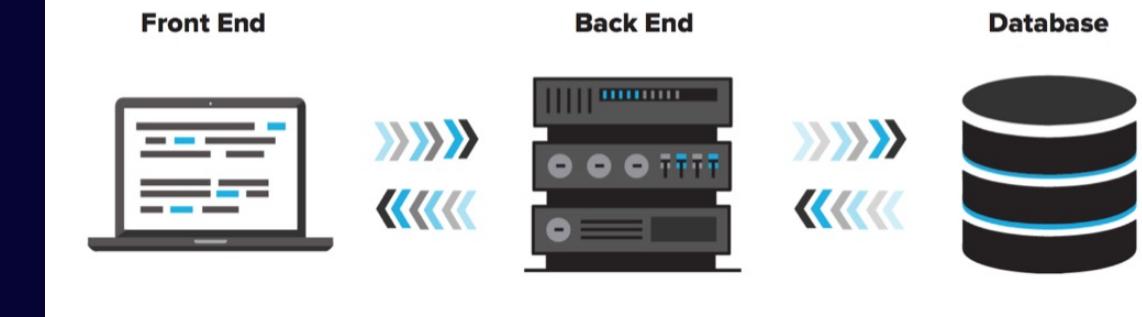
- Backend is everything else...
- The backend is what constructs the HTML+CSS+JS that is sent back to user.
- The backend of a web application is what enables you to have a good frontend user experience.
- It doesn't matter how beautifully designed a web application is, if the backend is not structured well, it will not work properly.
- The backend of the application is responsible for things like;
  - Calculations
  - Business logic
  - Database interactions
  - Performance etc...



<http://skillcrush.com/2012/04/17/frontend-vs-backend-3/>

## Web Application Architecture

# What is a database?



- A database is an organized collection of data
- Generally stored and accessed electronically from a computer system.
- Think of it as just a storage that stores information in an organized manner, so that it is easy to reach when needed.

# Restaurant Example:



FRONT-END



BACK-END

**Frontend**



**Backend**



# Front end specific technologies

- HTML
- CSS
- JavaScript
- What they are?
- How do they fit together?
- What roles they play?
- How they are similar/different?

# What is HTML

HTML is a language for describing Web pages.

- HTML stands for HyperText Markup Language.
- HTML is not a programming language, it is a **markup language**.
- Programming languages have programming logic such as; variables, conditions, loops etc.
- A markup language is a computer language that uses **tags** to define **elements** within a document.

# What is a “Markup Language”

- A mark up language is used to control the presentation of the data.
- HTML is a mark- up language, and it uses markup tags to describe and structurize Web pages

The Guardian | Monday 11 September 2011

**Read more on  
MediaGuardian.co.uk**

**Digital economy at last**  
Part 3, in which the team turn up the volume with inside track on The X Factor – and get a glimpse of the future.

**Coming up this week**  
Monday: Shortlist for Student Media Awards announced. Wednesday to Friday: Coverage of the RTS Cambridge Conference.

**Interview Rio Caraeff**

## Vevo revolutionary

Universal's former mobile chief is leading the music industry's fight to shake up online video. He reveals his frustration with MTV, and says why no one need own music if his site succeeds. Interview by **Mark Sweeney**

**Rio Caraeff** succeeds, perhaps, in the same way that Google did. His on-demand video site, part-owned by the two largest music companies, will not have the same impact as MTV and YouTube, but it will achieve those goals in spite of that: making the industry less dependent on the past, and giving the Guardianship of music a new lease of life.

Caraeff is the result of a confluence of forces – he was in late talks with the backing of three of the four major groups, Sony, Universal and EMI, before he left MTV, who is taking the venture international with a rollout starting in the UK and one million users. The other two major categories are the only entertainment categories on the planet that are growing, and they are the only ones that have had a significant influence on the scale of billions of people, he says. "It's in the business of helping people to connect with each other steadily and easily."

He has been a pioneer, ploughing his own furrow, and it paid off. Caraeff's vision is clear. "We wouldn't have created Vevo if we didn't believe in it," he says. "It was necessary if MTV's existing agent job-purchasing system, if MTV's business model, if MTV's culture, if MTV's management, if MTV's business plan, we have invested time and effort to be responsible for that. We've invested in the company to say 'I hope Apple or someone figures that out.'

Vevo's relationship with Google, the owner of the world's largest video library, is also clear. "Google has a massive catalog of music videos, and MTV has a huge collection of original TV content," he says. "We're creating a 'partner' and he [Mark] is well described as an 'ally'." The two companies are now looking to the music industry has hopefully fully had its chance to prove itself. "Giant profits to characterize Vevo's dealings with YouTube as 'syndication' are probably not appropriate," he says. "It's a symbiotic relationship that would be most appropriate."

"We said 'let's figure out what we want to do with YouTube'. There are no duplicate copies [of music videos] loaded on YouTube, so there's no revenue. The official versions are only available from us. They don't determine, because it's a platform, what people can watch on YouTube in the world, we're not trying to compete with YouTube. YouTube is the search, and YouTube is the destination, and YouTube is the side of the YouTube soft pages when a user is viewing chips."

**FREE ACCESS**

Vevo's business model is all about providing free video that fans can access first, funded by advertising, so you get it for free, and it's not necessarily something to be worried about. "I believe the future is access, not ownership, not if you are it rather than me. It's not about who owns the people, it's about who owns the music. We are about providing access, it is the only sensible model for the music industry to have. It's how we can make that and make money."

Vevo is a success, but Caraeff doesn't want to give away his commerciality, but says it's not "free to download millions of dollars" as revenue, although these are hosting costs to pay. More than half of the music videos on YouTube are – the label, artist or licensee – with the remainder being kept by the video provider. "That's the reason that Vevo is 'significantly ahead' of its original business plan, and it's the reason that we are on track to achieve profitability." In the very early part of next year, Vevo will be launching its own channel, and Caraeff says that the channel is dependent on advertising, and he is optimistic about the low rates that come with it. "Our contention is that advertising needs to be in the prime content because it's the 'water' the prime content lives in. It's the prime content that then and there is able to provide it as a premium product, and the free content, premium product of \$4.99 is coming. Premium product of \$4.99 is coming."

"The problem is that there's music to eat and breathing; it should be treated as

**Curriculum vitae**

Age 35

Education Didn't go to college. I started my first business at 16.

Career 2000: Was awarded Pictures 2000 prize for best mobile media division, and new technology book, *Web*.

If Rio Caraeff succeeds, perhaps only digital firms will need to listen to music. His on-demand video is part-owned by the two largest music companies, after hopes to keep the same impact as MTV and YouTube. Check those guys in with that of making the music industry less dependent on the purchase of recordings, and for Caraeff there is clearly plenty to do.

Caraeff is the plentiful chief executive of Vevo – he started in late 2009.

Page 23 / 514 — Q +

- HTML is often named as the structure, or the **skeleton** of a WebApp.
- There used to be no way to share a structured data.
- For example: I have a text that has:
  - **Bolded**
  - *Italic* with some bullet points etc.
- You couldn't have shared this text as it is.
- This is why the HTML was created.

# In summary:

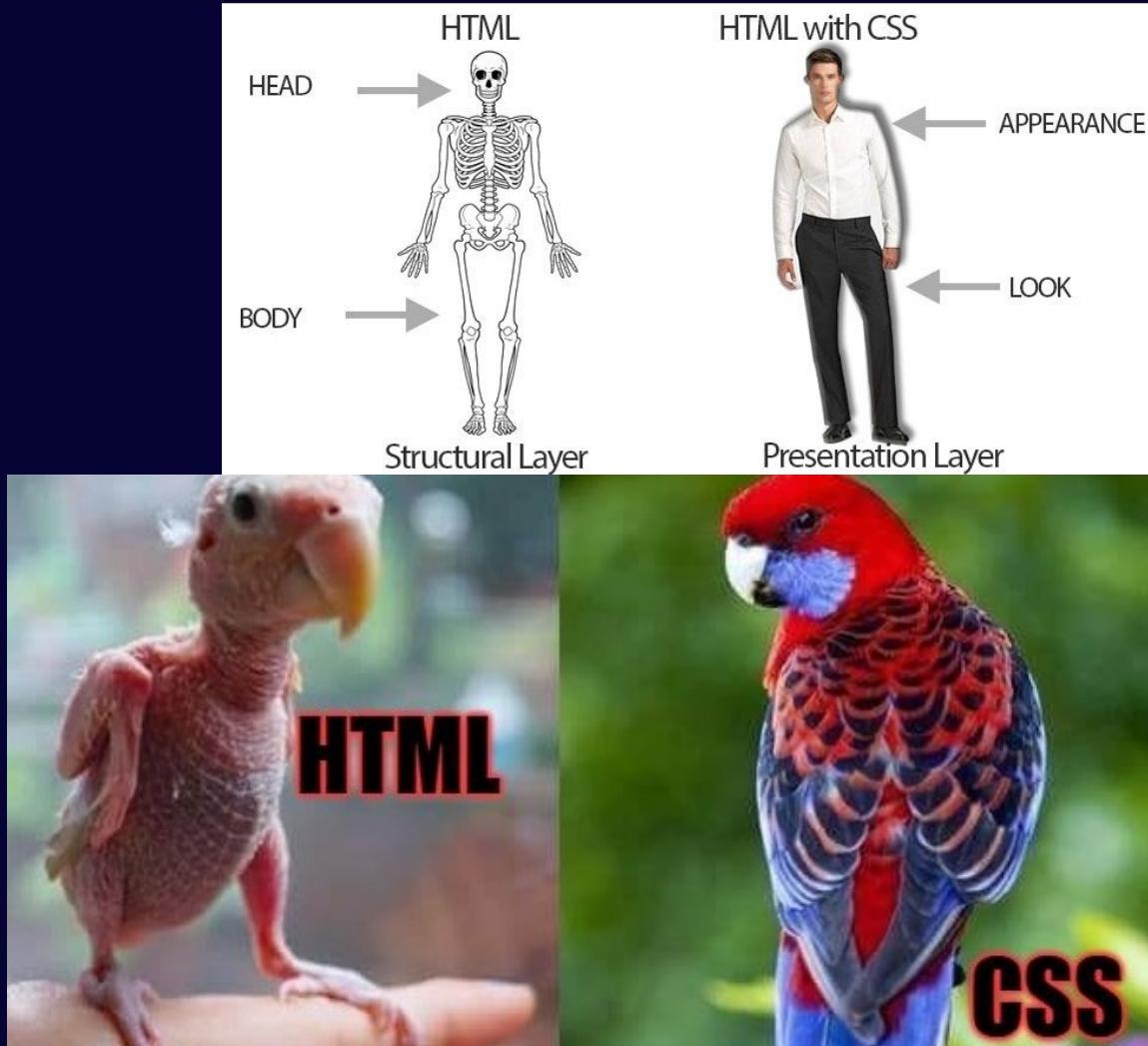
- HTML is a mark up language.
- Mark up languages are based on TAGS.
- Mark up languages are **NOT** programming languages, therefore they do not have any programming logic like loops or conditions.
- Mark up languages are used to create structured data.

# You can see and change the HTML structure of a Web App locally

- Lets change some news!
- <https://www.cnn.com/2019/06/12/politics/9-11-john-feal-brian-mcguire-jon-stewart-cnntv/index.html>

# CSS

- Cascading Style Sheets
- Defines the style of existing HTML
- CSS Cannot be used without HTML
  - You cannot create an element with CSS. You can only give style to an existing element.
- You give style to HTML using CSS
  - Make it purple with red borders
  - Give the first table a gray border, and yellow background
- Commonly addressed as THE SKIN of HTML skeleton.
- CSS allows you to specify how you want your website to look like.



# CSS ZenGarden

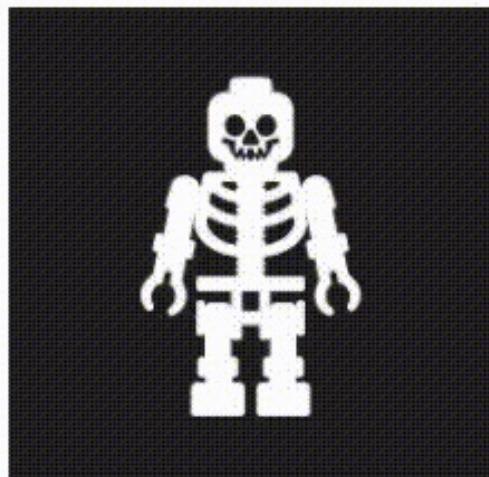
- The whole purpose of this website is to show us what you can accomplish with different types of CSS Styles.
  - Every single one of these different versions of the webpage have the EXACT same html back-bones (literally exactly same content).
  - Only different CSS styles has been applied to change the presentation.
- 
- <http://www.csszengarden.com/>

- CSS does not add any more buttons or texts to the page.
- It ONLY styles the existing structure.

# JavaScript

- JavaScript is a scripting language.
- JavaScript is used to add logic, interactivity, dynamism to a web page
- Allows your website to actually do things, and have behavior.
  - “Do some calculations”
  - “Change color when user clicks to some specific button”
  - “Load new data from a page”

**HTML**  
structure



**CSS**  
presentation/appearance



**JavaScript**  
dynamism/action



# Creating a simple web app

- We will only use two tools:

## 1. Text Editor

- You could be using any text editor from free ones to \$500 paid ones.
- At the end for what we will be doing, SUBLIME TEXT will be good enough.

## 2. Web Browser

- We will be using Chrome as our main web browser, simply because it has some “Developer tools” that makes our job easy.
- Firefox also has some “Dev tools” of its own, but not as intuitive as Chrome.



# Sublime text

- Basic note pad tool.
- Allows us to apply some easy syntax options, such as;
  - HTML
  - CSS
  - Java
  - JavaScript
  - Etc...

# Web browsers

- They let you view and cruise internet content
- Google chrome is used by a lot of developers
- It has some “DevTools” that makes developing easy
  - Which is something we will be taking advantage of as well

# Intro to HTML

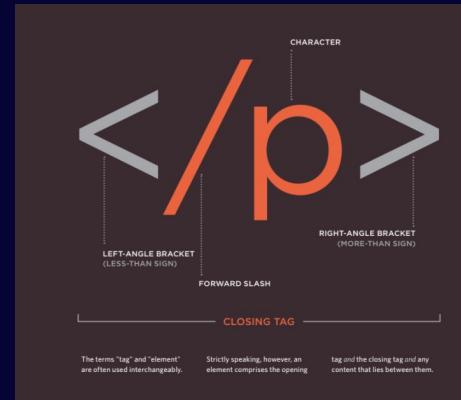
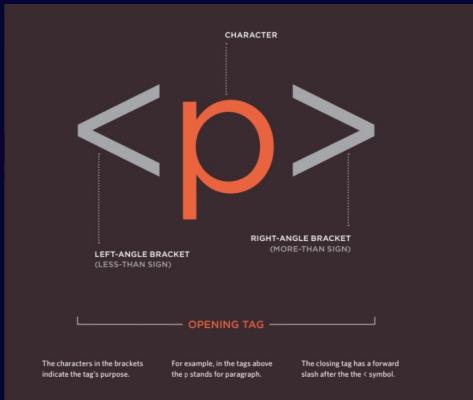
- Created in 1989/1990
- HTML allowed scientists to publish and exchange their technical documents
- Also allowed scientists to hyperlink to other documents
- [Another Document](#)
- [Codepen example](#)
- [Web Archive](#)
- [MIT Professor's personal page](#)

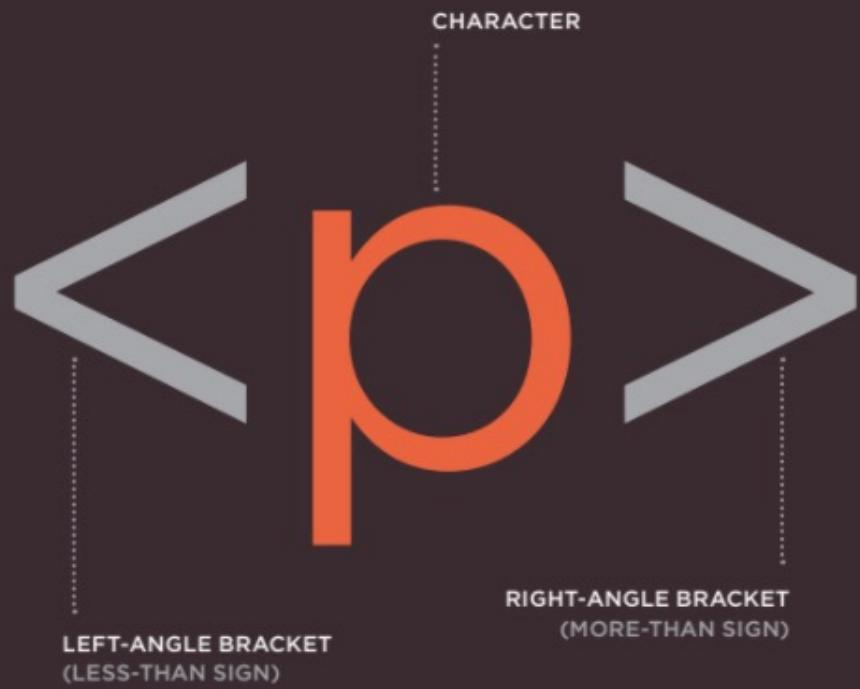
# What is a web element?

- Everything you see on a web page is a web element.
- Basically; every single box, button, text, image, link etc you see on a web app is a web element.
- <https://www.google.com>

# What are tags?

- Tags act like containers. They tell you something about the information that lies between their openings and closings.
- Tags are how HTML defines different areas and sections of the document of a web browser would call and interpret.
- Tags start with angular brackets and end with angular brackets → <tag>
- In between brackets you specify what kind of tag you want to use.



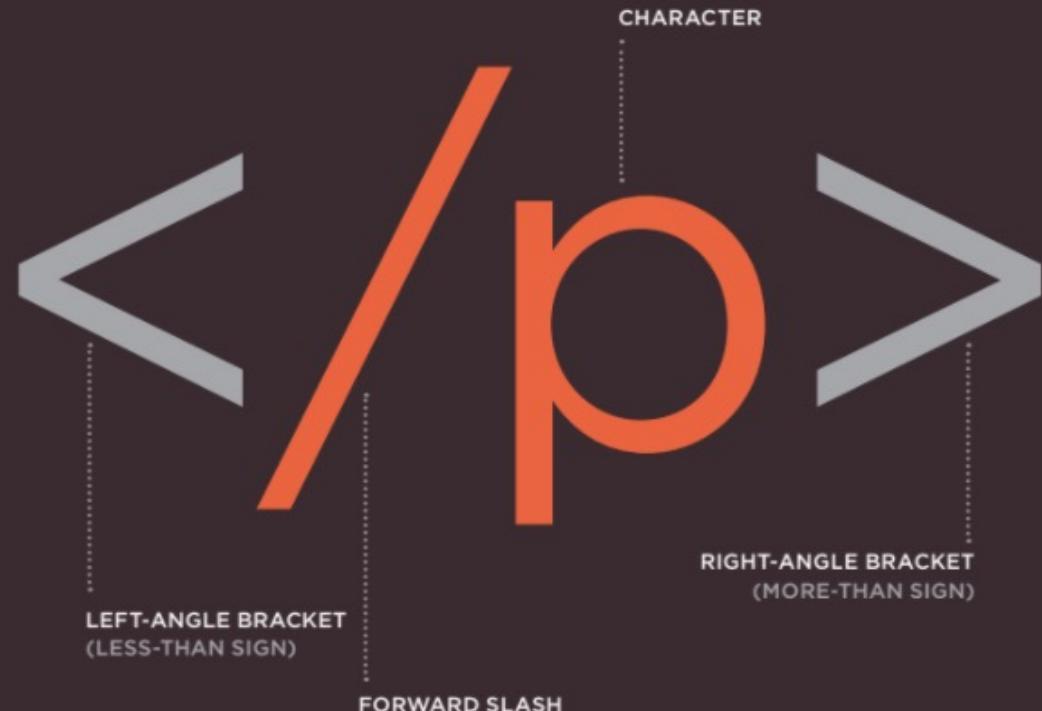


### OPENING TAG

The characters in the brackets indicate the tag's purpose.

For example, in the tags above the p stands for paragraph.

The closing tag has a forward slash after the < symbol.



### CLOSING TAG

The terms "tag" and "element" are often used interchangeably.

Strictly speaking, however, an element comprises the opening

tag and the closing tag and any content that lies between them.

# There are two types of tags:

## 1. Paired tags:

- Consists of two parts. Opening tag and closing tag.
- Ex: <div> </div>

## 2. Unpaired tags: also known as Self-closing tags

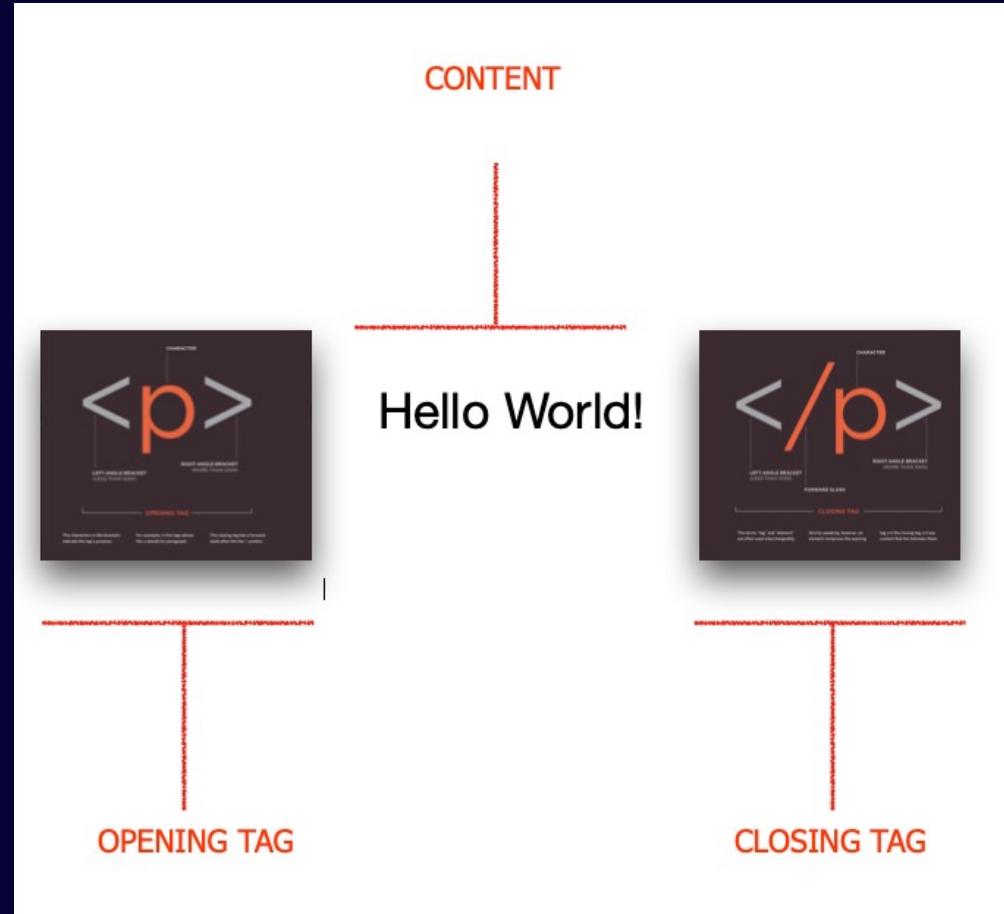
- Consists of only one part. No closing tag required.
- <br>
- <img>

# The general rule of HTML tags

```
<tagName> Some Content </tagName>
```

- We create two TAGS.
- First is called opening tag, and second is called closing the tag.
- Basically, this allows us to apply some specific condition to the content that is in between this specific tag.
- What is important to remember is that (if it is paired tag):
  - You open the tag
  - And then you close the tag
  - Then you can apply the specific condition that you want to apply
  - There are some tags that don't require you to close, that are called self closing tags.

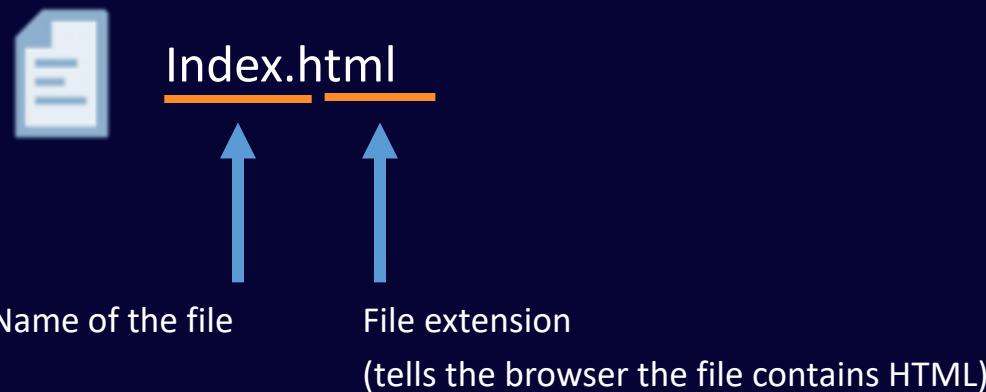
# The anatomy of an HTML tag



# HOW TO WRITE HTML FILES?

Just like most programming languages, we type a bunch of HTML into a file so we can send it around.

“.html” extension tells your browser that this is an html file, treat it as such.



# HOW TO DISPLAY HTML FILES?

Web Browsers are basically “HTML Readers.”



# Structure of an html page

```
<!DOCTYPE html>
```

```
<HTML>
```

```
  <HEAD>
```

```
    </HEAD>
```

```
<BODY>
```

CONTENT

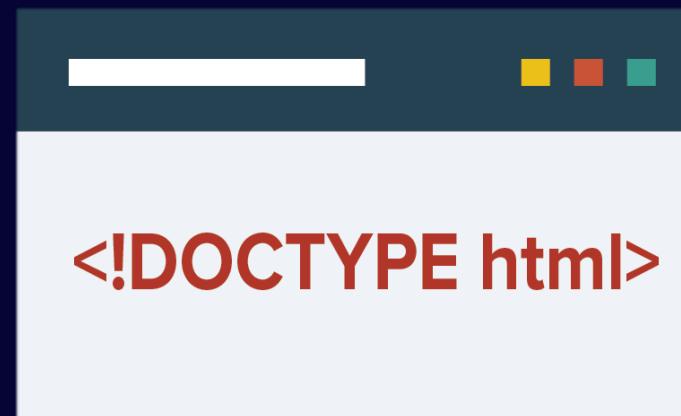
```
  </BODY>
```

```
</HTML>
```

# DOCTYPE DECLARATION

It is an instruction to the web browser about what version of HTML the page is written in.

- Doctype in HTML must be at the very top of the document, before all elements.
- Doctype declaration is not case-sensitive.

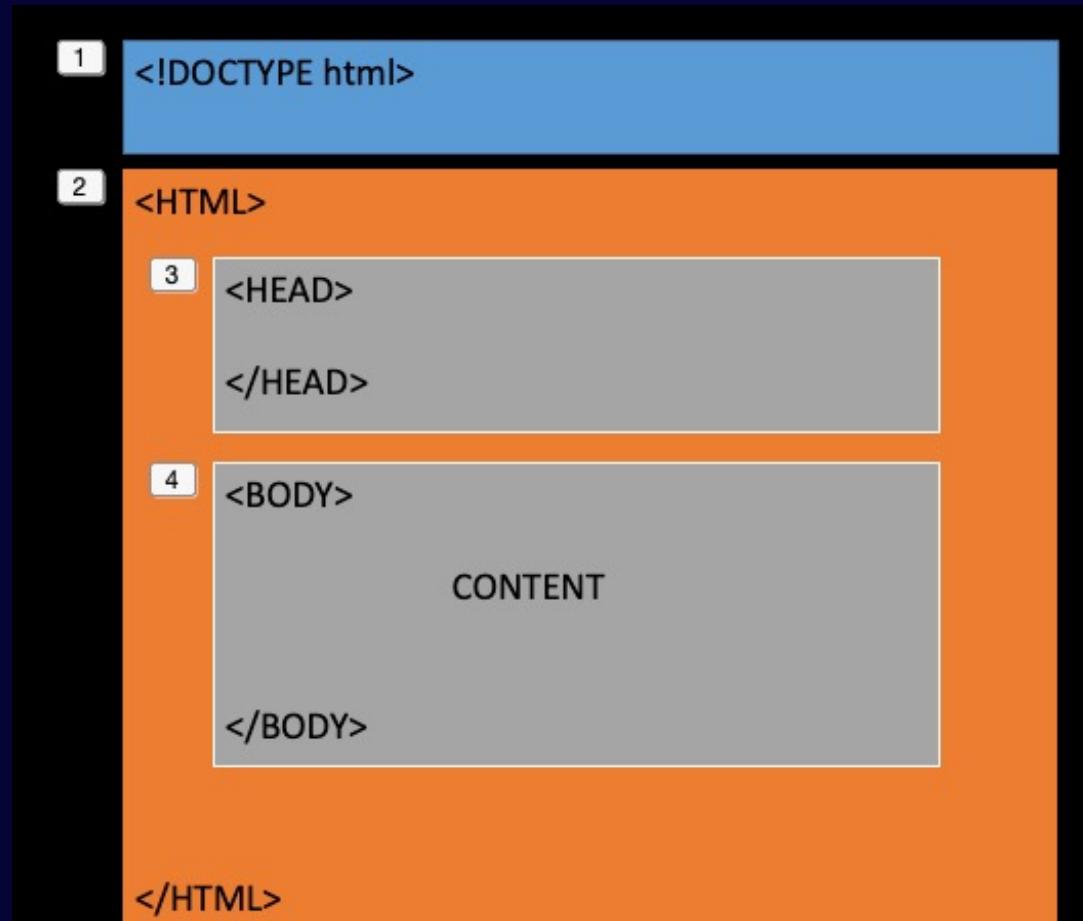


`<html> </html>`

- `<html>` tag is the **root** element of **every** web application
- Every other web element must be **descendent/child** of the `html` element.
- Which means, **everything** must go inside of the `html` element.
- It is followed by two other elements.
- `<head></head>` and `<body></body>` tags

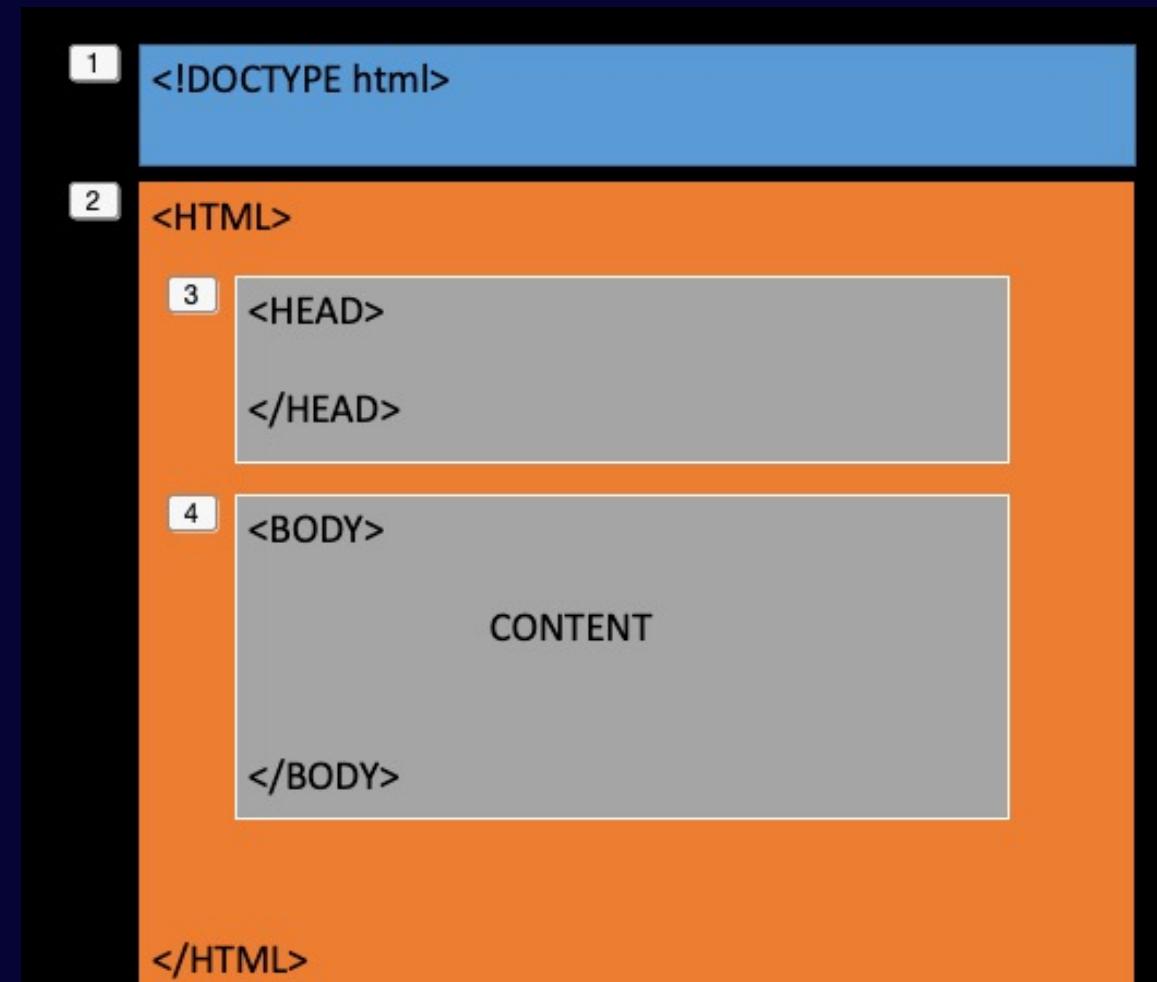
# Head

- Head element provides general information about the document, including titles, links and definition
- Basically **everything we don't see on the page as a user**; font files, JavaScript files, titles etc is put in the <head >



# Body

- Body represents the content of an HTML document
- Basically where we put all of our content and create the structure of the web page.



# Let's create our first HTML file

1. Create a new folder on your **Desktop** called: HTML Class
2. Open SublimeText
  - You should have it downloaded. Open sublime text.
3. Create a new file inside of the “HTML Class” folder
  - File > New File
4. Give it a name → myFirstPage.html
  - Save as > Select Desktop > Select HTML Class folder > give name
5. Save it and open it with Google Chrome
6. <h1>this is my first h1 tag</h1>
7. Save the file again.
8. Open in browser.

# Ways for you to view your HTML file

1. You can save and then double click to the file itself.
2. You can right click on the HTML file and open with Chrome.
3. You can drag and drop it into chrome.
4. You can right click inside sublime, and select “Open in Browser”

# Was it the correct HTML structure that we just created?

- No.
- We need to put everything inside of an **html** tag.
- Then we have the **head** tag.
- And all the content needs to go inside of **body** tag.
- Let's type the correct structure of the html page.

```
<!DOCTYPE html>
```

```
<HTML>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BODY>
```

CONTENT

```
</BODY>
```

```
</HTML>
```

# The correct structure of every web app

```
<!DOCTYPE html>
<html>

    <head>
        <title></title>
    </head>

    <body>
    </body>

</html>
```

# Title

- Title is what you see inside of the tab of the web browser.
- You don't see the title inside of the web page.
- But you do see it on the top when you create a new tab.
- It is also important for search engines, because the title is what comes up when you search something on google.
- And we will use it a lot when we do automation.
- Check from Google, Etsy etc.

# HTML HEADINGS

- HTML headings are defined with the `<h1>` to `<h6>` tags.
- `<h1>` is used for main headings
- `<h2>` is used for subheadings
- IF there are further sections under the subheadings then the `<h3>` element is used, and so on
- Headings are block elements: which means they take the whole line.

# Practice #1

- Write the code to achieve this outcome.

**This is a Main heading**

**This is a Level 2 heading**

**This is a Level 3 heading**

**This is a Level 4 heading**

**This is a Level 5 heading**

**This is a Level 6 heading**

# HTML HEADINGS

```
<!DOCTYPE html>

<html>
  <head>
    <title>This is my title</title>
  </head>

  <body>
    <h1>This is a Main heading</h1>
    <h2>This is a Level 2 heading</h2>
    <h3>This is a Level 3 heading</h3>
    <h4>This is a Level 4 heading</h4>
    <h5>This is a Level 5 heading</h5>
    <h6>This is a Level 6 heading</h6>
  </body>

</html>
```

**This is a Main heading**

**This is a Level 2 heading**

**This is a Level 3 heading**

**This is a Level 4 heading**

**This is a Level 5 heading**

**This is a Level 6 heading**

# HTML PARAGRAPHS

- To create a paragraph, you surround the words that make up the paragraph with an opening `<p>` tag and closing `</p>` tag.
- `<p>` is also a block element. It takes its own line by itself by default.

A paragraph consists of one or more sentences that form a self-contained unit of discourse. The start of a paragraph is indicated by a new line.

Text is easier to understand when it is split up into units of text. For example, a book may have chapters. Chapters can have subheadings. Under each heading there will be one or more paragraphs.

# Practice#2

- Create three paragraphs.
- Get the paragraphs from Lorem Ipsum. (What is lorem ipsum?)

## **My first paragraph**

  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Mauris rhoncus aenean vel elit scelerisque mauris pellentesque. Sit amet luctus venenatis lectus magna fringilla urna. Dui accumsan sit amet nulla facilisi. Phasellus egestas tellus rutrum tellus pellentesque. Mi proin sed libero enim sed faucibus turpis in. In metus vulputate eu scelerisque felis imperdiet proin fermentum leo. Vitae et leo duis ut diam quam. Aliquam vestibulum morbi blandit cursus risus at ultrices mi. Ut sem viverra aliquet eget sit amet tellus cras. Et molestie ac feugiat sed lectus vestibulum mattis ullamcorper. Tempus urna et pharetra pharetra massa massa ultricies. Maecenas volutpat blandit aliquam etiam erat. Consectetur lorem donec massa sapien faucibus et molestie ac. Amet nisl suscipit adipiscing bibendum est ultricies. Sed ullamcorper morbi tincidunt ornare massa eget egestas purus. A condimentum vitae sapien pellentesque habitant. Non odio euismod lacinia at quis. Eu volutpat odio facilisis mauris sit amet massa.

## **My second paragraph**

  Ultrices mi tempus imperdiet nulla malesuada. Arcu risus quis varius quam. Donec adipiscing tristique risus nec. In cursus turpis massa tincidunt dui ut ornare. Quis vel eros donec ac odio tempor orci. Egestas congue quisque egestas diam. Adipiscing elit duis tristique sollicitudin nibh sit amet. Mauris pharetra et ultrices neque ornare aenean euismod. Pharetra massa massa ultricies mi quis hendrerit dolor magna eget. Senectus et netus et malesuada fames ac turpis egestas. Fusce ut placerat orci nulla pellentesque dignissim enim sit amet. Ut morbi tincidunt augue interdum.

## **My third paragraph**

  Purus in massa tempor nec feugiat nisl pretium fusce id. Tempor id eu nisl nunc mi ipsum. Bibendum neque egestas congue quisque egestas diam in arcu. Sit amet nisl suscipit adipiscing bibendum. Semper eget duis at tellus at urna condimentum mattis. Est sit amet facilisis magna. Pellentesque habitant morbi tristique senectus et netus et malesuada. Integer quis auctor elit sed vulputate mi sit amet. Sit amet commodo nulla facilisi nullam vehicula ipsum a arcu. Sit amet nulla facilisi morbi tempus iaculis urna id volutpat. Etiam tempor orci eu lobortis elementum nibh tellus. Pellentesque diam volutpat commodo sed egestas egestas fringilla phasellus. Ante metus dictum at tempor commodo. Ac auctor augue mauris augue neque.

# Solution

(lorem +tab)

```
<h1>My first paragraph</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Mauris rhoncus aenean vel elit
scelerisque mauris pellentesque. Sit amet luctus venenatis lectus magna fringilla
urna. Dui accumsan sit amet nulla facilisi. .</p>

<h1>My second paragraph</h1>
<p>Ultrices mi tempus imperdiet nulla malesuada. Arcu risus quis varius quam.
Donec adipiscing tristique risus nec. In cursus turpis massa tincidunt dui ut
ornare. Quis vel eros donec ac odio tempor orci. Egestas congue quisque egestas
diam. Adipiscing elit dui tristique sollicitudin nibh sit amet. Mauris pharetra
et ultrices neque ornare aenean euismod.

</p>

<h1>My third paragraph</h1>
<p>Purus in massa tempor nec feugiat nisl pretium fusce id. Tempor id eu nisl nunc
mi ipsum. Bibendum neque egestas congue quisque egestas diam in arcu. Sit amet
nisl suscipit adipiscing bibendum. Semper eget dui at tellus at urna condimentum
mattis. Est sit amet facilisis magna. Pellentesque habitant morbi tristique
senectus et netus et malesuada. Integer quis auctor elit sed vulputate mi sit
amet. Sit amet commodo nulla facilisi nullam vehicula ipsum a arcu. Sit amet nulla
facilisi morbi tempus iaculis urna id volutpat.
</p>
```

# LINE BREAKS & HORIZONTAL RULES

- To add a line break inside the middle of a paragraph, use the line break tag `<br>`
- To create a break between themes, add a horizontal rule between sections using the `<hr>` tag.

`hr`: stands for horizontal ruler.

```
<p>Hello <br> This is new line. <br> This is another new line</p>
<hr>
```

Hello  
This is new line.  
This is another new line

---

# Practice #3

My name is Mike

My age is 25

My job is Automation Tester

---

MY name is John

My age is 30

My job is Developer

# Solution #3

```
<body>
  <p>My name is Mike <br /> My age is 25 <br /> My job is
  Automation Tester</p>
  <hr />
  <p>MY name is John <br /> My age is 30 <br /> My job is Developer
  </p>
</body>
```

# BOLD & ITALIC

- By enclosing words in the tags `<strong></strong>` we can make characters appear bold.
- By enclosing words in the tags `<em>` and `</em>` we can make characters appear italic.
- By enclosing words in the tags `<u>` and `</u>` we can underline the characters

My name is **Mike**

My age is 25

My job is Automation Tester

---

MY name is **John**

My age is 30

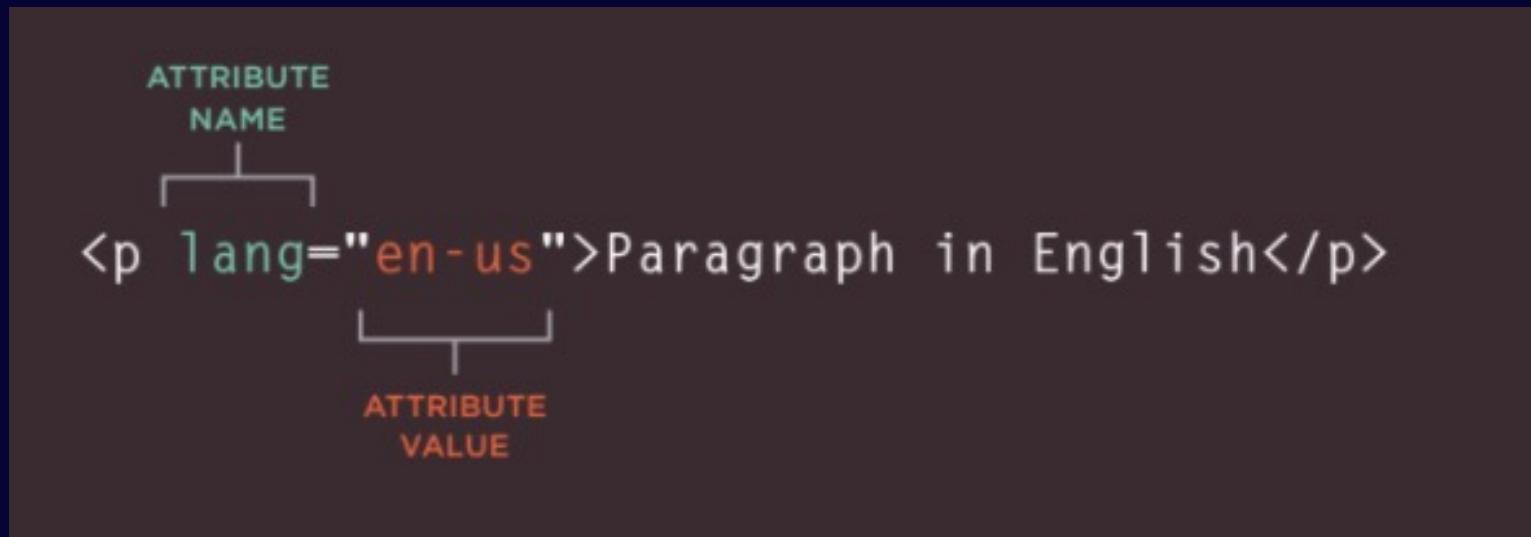
My job is Developer

# Practice #4

```
<body>
  <p>My name is <strong>Mike</strong> <br> My age is 25 <br> My job is <u><em>Automation Tester</em></u></p>
  <hr>
  <p>My name is <strong>John</strong> <br> My age is 30 <br> My job is <u><em>Developer</em></u></p>
</body>
```

# HTML ATTRIBUTES

- Attributes provide additional information for that specific web element only.
- They appear in the **opening tag** of the element and are made up of two parts: a **name** and a **value**, separated by an equal sign.



# Practice #6

- Create a horizontal ruler using `<hr>` tag.
- Change the “size” attribute value to 5px.
- It should look like this.

My name is **Mike**

My age is 25

My job is Automation Tester

---

My name is **John**

My age is 30

My job is Developer

# HTML LISTS

- There are **two** types of HTML lists.
  - 1- **Ordered lists** are lists where each item in the list is numbered.
  - For example, the list might be a set of steps for a recipe that must be performed in order.
  - 2- **Unordered lists** are lists that begin with a bullet point, no numbers.

# ORDERED LISTS

- The ordered list is created with the `<ol>` element/tag.
- Each item in the list is placed between an opening `<li>` tag and a closing `</li>` tag.
- Let's create an ordered list!

```
<p>How to make an amazing tomato sauce for pasta:</p>
<ol>
    <li>Put some tomato sauce</li>
    <li>Put some butter</li>
    <li>Put some seasoning</li>
    <li>Stir and cook for 5 minutes</li>
</ol>
```

How to make an amazing tomato sauce for pasta:

1. Put some tomato sauce
2. Put some butter
3. Put some seasoning
4. Stir and cook for 5 minutes

# UNORDERED LISTS

The unordered list is created with the `<ul>` element.

Each item in the list is placed between an opening `<li>` tag and a closing `</li>` tag.

Let's create an unordered list!

```
<p>List of items you need to make pasta</p>
<ul>
    <li>Pasta</li>
    <li>Water</li>
    <li>Salt</li>
</ul>
```

List of items you need to make pasta

- Pasta
- Water
- Salt

# NESTED LISTS

Second list can be placed inside an `<li>` element to create a sub-list or nested list.

```
<p>List of colors</p>
<ul>
    <li>White</li>
    <li>Green</li>
        <ul>
            <li>Dark Green</li>
            <li>Light Green</li>
        </ul>
    <li>Yellow</li>
</ul>
```

List of colors

- White
- Green
  - Dark Green
  - Light Green
- Yellow

Let's create a nested list!

# Practice #5

- Create the following list of fruits using:
- Ordered lists, and unordered lists.
- Make main apples bold.
- Make the other inner list items emphasized (italic)

## My Fruits List

1. **Apple**
  - *Red Apple*
  - *Green Apple*
2. **Banana**
  - *Frozen Banana*
  - *Banana milk shake*
3. **Cherries**
4. **Strawberries**

## Solution #5

```
<p><strong>My Fruits List</strong></p>
<ol>
    <li><strong>Apple</strong></li>
        <ul>
            <li><em>Red Apple</em></li>
            <li><em>Green Apple</em></li>
        </ul>
    <li><strong>Banana</strong></li>
        <ul>
            <li><em>Frozen Banana</em></li>
            <li><em>Banana Milk Shake</em></li>
        </ul>
    <li><strong>Cherries</strong></li>
    <li><strong>Strawberries</strong></li>
</ol>
```

# HTML COMMENTING

HTML comments are visible to anyone that views the page source code, but are not rendered when the HTML document is rendered by a browser.

<!-- Comment -->

Also, when you are in a line, you can just press “cmd+/"

```
<body>
    <p>My name is Mike</p>
    <!--This is a comment line-->
    <p>My age is 35</p>
</body>
```

My name is Mike

My age is 35

# Html images

To add an image into the page you need to use an **<img>** element.

This is a self closing element.

It needs to carry the following attribute:

**src** : This tells the browser where it can find the image file.

```
<body>
  <p>This is an image</p>
  
</body>
```



# Practice #7

- Find and image from google.
- Add an image to your html document.



# Solution

- Make sure the image and the .html file is in the same folder directory.

```

```

- This will add the image to the HTML page and change the width of the image to 150px.

# CREATING LINKS

- Links are created using the `<a>` tag. (anchor)
- Users can click on anything between the opening `<a>` tag and the closing `</a>` tag.
- You specify which page you want to link to using the `href` attribute.
- What does `href` stands for?
- Hypertext Reference



# LINK (anchor tag) use cases

- User can be re-directed to different types of pages using links, such as;
- Links from one website to another
- Links from one part of a web page to another part of the same page
- Links that open in a new browser window
- So on and so forth...

# LINKING TO OTHER SITES

- Users can click on anything that appears between the opening `<a>` tag and the closing `</a>` tag and will be taken to the page specified in the href attribute.



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

# Practice #8

- Create a new html file, name & title:
  - Car Reviews
- When you click to the links it should take you to
  - KBB.com
  - EDMUNDs.com
  - CARMAX.com

Car Reviews:

- KBB
- EDMUNS
- CARMAX

# Solution

```
<body>
<font size="20">
    <p>Car Reviews:
        <ul>
            <li><a href="https://www.kbb.com/">KBB</a></li>
            <li><a href="https://www.edmunds.com/">EDMUNDS</a></li>
            <li><a href="https://www.carmax.com/">CARMAX</a></li>
        </ul>
    </font>
</body>
```

# LINKING TO OTHER PAGES ON THE SAME SITE

- When you are linking to other pages within the same site, you do not need to specify the domain name in the URL. You can use a shorthand known as a **relative** URL
- If all the pages of the app are in the same folder, then the value of the href attribute is just the name of the file.

```
<a href="index.html">Home</a>
```

# Practice #9

- Create a new page called CarReviews2
- Add these specific links that are taking the user to these sample pages
- In the new pages just have one paragraph, content: anything.

## Car Reviews:

- [Home](#)
- [About](#)
- [Contact](#)

# Solution:

```
<body>
<font size="20">
    <p>Car Reviews:
        <ul>
            <li><a href="index.html">Home</a></li>
            <li><a href="about-us.html">About</a></li>
            <li><a href="contact.html">Contact</a></li>
        </ul>
    </font>
</body>
```

# Bigger Practice &/ Homework

- Create a new folder inside of HTML Class called OlafVlog
- Create a new HTML file called olaf.html
- Create the page using whatever we have learned so far
- Make it look like the example

**Hi! I'm Olaf!**

Also known as "O-Laugh the happy dog!"

**Welcome to my blog! Woof!**



**Some of my favorite activities:**

- Chasing squirrels
- Chasing birds
- Playing with my owner
- Running into mud

Please share my page and subscribe to my Youtube channel here: [My Youtube Channel](#)

# Solution – and image

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>OLAF's Blog</title>
5 </head>
6 <body>
7
8     <h1>Hi! I'm Olaf! </h1>
9     <h3>Also known as "O-Laugh the happy dog!"</h3>
10    <h2>Welcome to my blog! Woof!</h2>
11
12    
13
14    <p><strong>Some of my favorite activities: </strong></p>
15    <ul>
16        <li>Chasing squirrels</li>
17        <li>Chasing birds</li>
18        <li>Playing with my owner</li>
19        <li>Running into mud</li>
20    </ul>
21
22    <p>Please share my page and subscribe to my Youtube channel here: <a href="https://www.youtube.com">My
23        Youtube Channel</a></p>
24
25 </body>
</html>
```



# HTML TABLES

- The `<table>` element is used to add tables to a web page
- A table is drawn out row by row. Each row is created with the `<tr>` element.
- Inside each row there are a number of cells represented by the `<td>` element(or `<th>` if it is a header).
- You can make cells of a table span more than one row or column using the `rowspan` and `colspan` attributes.

# BASIC TABLE STRUCTURE

The `<table>` element is used to create a table. The contents of the table are written out row by row.

Start of each row is indicated using the opening `<tr>` tag.(The tr stands for table row).

Each cell of a table is represented using a `<td>` element.(The td stands for table data)

```
<body>
  <table>
    <tr>
      <td>15</td>
      <td>10</td>
      <td>30</td>
    </tr>
    <tr>
      <td>45</td>
      <td>60</td>
      <td>45</td>
    </tr>
    <tr>
      <td>60</td>
      <td>90</td>
      <td>100</td>
    </tr>
  </body>
```

15	10	30
45	60	45
60	90	100

# TABLE HEADINGS

The **<th>** element is used just like the **<td>** element but its purpose is to represent the heading for either a column or a row.(The th stands for table heading).

```
<body>
  <table>
    <tr>
      <th></th>
      <td>Saturday</td>
      <td>Sunday</td>
    </tr>
    <tr>
      <th>Tickets sold:</th>
      <td>120</td>
      <td>135</td>
    </tr>
    <tr>
      <th>Total sales:</th>
      <td>$600</td>
      <td>$1000</td>
    </tr>
  </body>
```

	Saturday	Sunday
<b>Tickets sold:</b>	120	135
<b>Total sales:</b>	\$600	\$1000

# Practice #11

- Create the following table

Name	Age
MuddyBuddy	2y
Fur-dinand	3y
Subwoofer	6m

# Solution

```
<table border="1px">

    <thead>
        <tr>
            <th>Name</th>
            <th>Age</th>
        </tr>
    </thead>

    <tbody>
        <tr>
            <td>MuddyBuddy</td>
            <td>2y</td>
        </tr>
        <tr>
            <td>Fur-dinand</td>
            <td>3y</td>
        </tr>
        <tr>
            <td>Subwoofer</td>
            <td>6m</td>
        </tr>
    </tbody>

</table>
```

# Practice #12 (Homework)

- Create the table for champions league results.
- You can get the images from google.

FINAL 4 OF CHAMPION'S LEAGUE

Logo	Order	Team Name	Captain
	1	Liverpool	Jordan Henderson
	2	Tottenham	Hugo Lloris
	3	Barcelona	Messi
	4	Ajax	Joel Veltman

# There are more than 100 HTML tags

- BUT! Not all of them are commonly used.
- The common ones we will be seeing as testers when inspecting are:
- Div
- Select (Dropdowns)
- Radio buttons & Check boxes
- Buttons
- Text input
- Password input

# GROUPING TEXT & ELEMENTS IN A BLOCK

The `<div>` element allows you to group a set of elements together in one block-level box.

`<div>` is also a block element. It takes the whole line.

`<span></span>` is an inline grouping tag. Doing similar things as `div`.

```
<body>
  <font size="10">
    <div>
      <h3>This is first group</h3>
      <ul><li>Java</li>
          <li>JavaScript</li>
          <li>Ruby</li></ul>
    </div>
    <div>
      <h3>This is second group</h3>
      <ul><li>Java</li>
          <li>JavaScript</li>
          <li>Ruby</li></ul>
    </div>
  </font>
</body>
```

**This is first group**

- Java
- JavaScript
- Ruby

**This is second group**

- Java
- JavaScript
- Ruby

# GROUPING TEXT & ELEMENTS IN A BLOCK

The `<div>` element allows you to group a set of elements together in one block-level box.

```
<body>
  <font size="10">
    <div>
      <h3>This is first group</h3>
      <ul><li>Java</li>
          <li>JavaScript</li>
          <li>Ruby</li></ul>
    </div>
    <div>
      <h3>This is second group</h3>
      <ul><li>Java</li>
          <li>JavaScript</li>
          <li>Ruby</li></ul>
    </div>
  </font>
</body>
```

## This is first group

- Java
- JavaScript
- Ruby

## This is second group

- Java
- JavaScript
- Ruby

# GROUPING TEXT & ELEMENTS INLINE

The `<span>` element acts like an inline equivalent of the `<div>` element.  
It is used to either:

1. Contain a section of text where there is no other suitable element to differentiate it from its surrounding text.
2. Contain a number of inline elements.

```
<body>
  <h1>This is <span style="color:red">how</span> <span style="color:green">span
  works</span></h1>
</body>
```

This is **how** span works

# HTML FORMS

- Forms are basically a container for all of these different types of inputs.
- Forms are created using <form></form> tag.
- Checkboxes, buttons, dropdown menus, and color pickers go in form tag.
- Forms are sending data to server or database or wherever we set it up to sent.
- The form we create could be just creating a get request.
  - Example: Google search, or logins
- Or it could be creating a post request;
  - Example: Signing up to a website

**Registration Form**

First name:

Last name:

Age:

E-mail:

Password:

SEX::  Male  Female

Checkbox:  I am a student  I am a business man

Birthday:

Day:  Month:  Year:

Submit Button:

Reset Button:

New to Twitter? Sign up

Full name

Email

Password

**Sign up for Twitter**



Email or Phone

Password

Keep me logged in [Forgot your password?](#)

**Log In**

## Sign Up

It's free and always will be.

First name

Last name

Email or mobile number

Re-enter email or mobile number

New password

Birthday

Month  Day  Year  [Why do I need to provide my birthday?](#)

Female  Male

By clicking Sign Up, you agree to our [Terms](#) and that you have read our [Data Policy](#), including our [Cookie Use](#).

**Sign Up**

# TEXT INPUT

## <input>

The <input> element is used to create several different form controls. The value of the type attribute determines what kind of input they will be creating

**type="text"**

When the type attribute has a value of text, it creates a single line text input.

**name**

When users enter information into a form, the server needs to know which form control each piece of data was entered into.

```
<form>
  <p>Username:
    <input type="text" name="username">
  </p>
</form>
```

Username:

# PASSWORD INPUT

type="password"

When the type attribute has a value of password it creates a text box that acts just like a single-line text input, except the characters are blocked out.

name

The name attribute indicates the name of the password input, which is sent to the server with the password the user enters.

```
<form>
  <p>Username:<br/>
    <input type="text" name="username">
  </p>
  <p>Password:<br/>
    <input type="password" name="password">
  </p>
</form>
```

Username: CYBERTEK

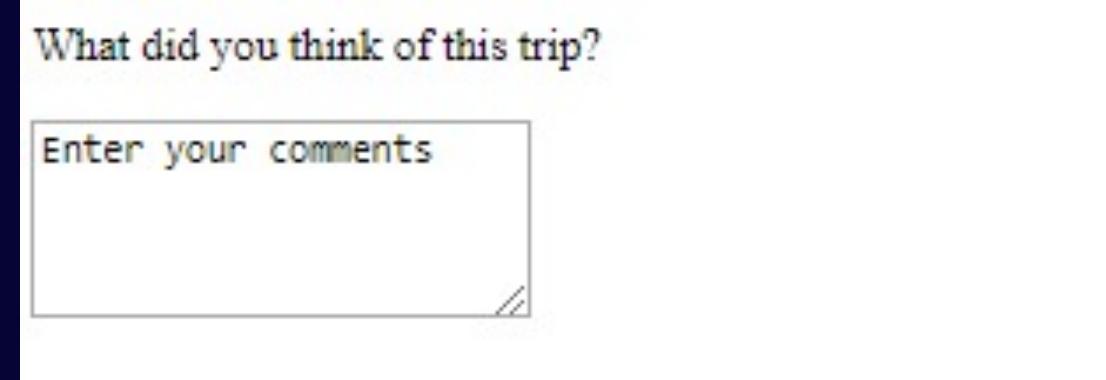
Password: .....

# TEXT AREA

The <textarea> element is used to create a multi-line text input. Unlike other input elements this is not an empty element. It should therefore have an opening and closing tag.

Any text that appears between the opening <textarea> and closing </textarea> tags will appear in the text box when the page loads.

```
<form>
  <p>What did you think of this trip?</p>
  <textarea name="comments" cols="20" rows="4">Enter your comments</textarea>
</form>
```



The image shows a screenshot of a web browser displaying a form. At the top, there is a question: "What did you think of this trip?". Below it is a text area with the placeholder text "Enter your comments". The text area has a standard gray border and a slightly darker gray background.

# RADIO BUTTON

**type="radio"**

Radio buttons allow users to pick just one of a number of options

**name**

The name attribute is sent to the server with the value of the option the user selects.

**value**

The value attribute determines the value that is sent to the server for the selected option. The value of each of the buttons in a group should be different.

**checked**

The checked attribute can be used to indicate which value should be selected when the page loads. The value of this attribute is checked.

# RADIO BUTTON

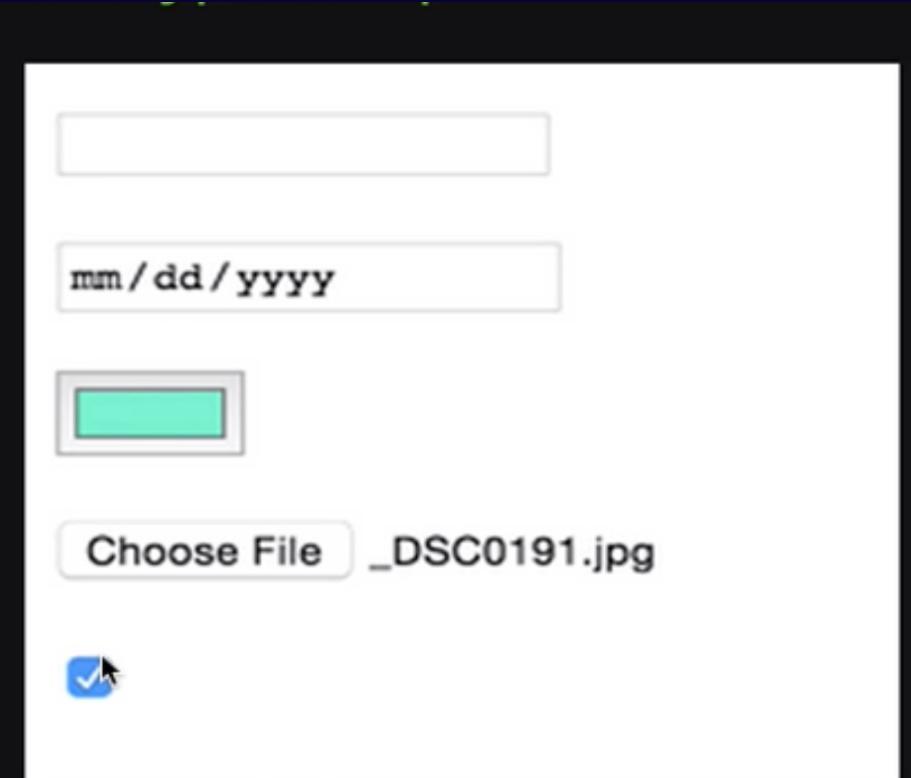
```
<form>
  <p>Please select your favorite food:
  <br/>
  <input type="radio" name="food" value="burger"
    checked="checked"/>Burger
  <input type="radio" name="food" value="Kabab"/>Kabab
  <input type="radio" name="food" value="Salad"/>Salad
  </p>
</form>
```

Please select your favorite food:  
● Burger ● Kabab ● Salad

# The <input> Tag

The input tag creates interactive controls. The “type” attribute determines the type of input

```
<input type="text">  
  
<input type="date">  
  
<input type="color">  
  
<input type="file">  
  
<input type="checkbox">
```



The image shows a web page with five input fields demonstrating different types:

- A text input field.
- A date input field with the placeholder "mm/dd/yyyy".
- A color input field with a green square preview.
- A file input field labeled "Choose File" with the file name "\_DSC0191.jpg" displayed next to it.
- A checkbox input field which is checked.

# checkbox

**type="checkbox"**

Checkboxes allow users to select(and unselect) one or more options in answer to a question.

**name**

The name attribute is sent to the server with the value of the option(s) the user selects.

**value**

The value attribute indicates the value sent to the server if this checkbox is checked.

**checked**

The checked attribute indicates that this box should be checked when the page loads. If used, its value should be checked.

# What are the differences between radio button and checkboxes?

- Each **checkbox** operates individually.
- Options are independent from each other.
- So users are allowed to select **multiple options** at once.
- **Radio buttons**, however, operate as a **group and are** used to allow user to select one from a set of options.
- So users are allowed to select only **one option**.

## CHECKBOXES

- red
- blue
- green

## RADIOBUTTONS

- red
- blue
- green

# Practice #12

- Create with input type checkbox example.

**Select your car (s)**

- mercedes
- bmw
- toyota

**Select one color**

- red
- blue
- green

# Solution

```
<h2>Select your car (s)</h2>
<input type="checkbox" name="option1" value="mercedes">mercedes <br>
<input type="checkbox" name="option2" value="bmw">bmw <br>
<input type="checkbox" name="option3" value="toyota">toyota <br>
<h2>Select one color</h2>
<input type="radio" name="option" value="red">red
<input type="radio" name="option" value="blue">blue
<input type="radio" name="option" value="green">green
```

# DROP DOWN LIST BOX

## <select>

A drop down list box(also known as a select box) allows users to select one option from a drop down list. The <select> element is used to create a drop down list box.

## name

The name attribute indicates the name of the form control being sent to the server, along with the value the user selected.

## <option>

The <option> element is used to specify the options that the user can select from.

## value

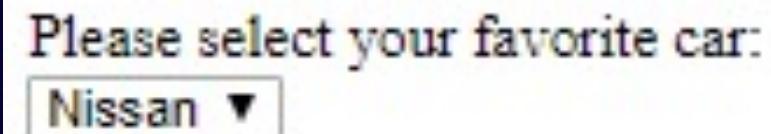
The <option> element uses the value attribute to indicate the value that is sent to the server along with the name of the control if this option is selected.

## selected

The selected attribute can be used to indicate the option that should be selected when the page loads.

# DROP DOWN LIST BOX

```
<form>
    <p>Please select your favorite car:</p>
    <br/>
    <select name="cars">
        <option value="honda">Honda</option>
        <option value="nissan" selected="selected">Nissan</option>
        <option value="ford">Ford</option>
    </select>
    </p>
</form>
```



# Practice #13

---

- Create a new folder Webstaurant
- Create new HTML file index.html
- Create the web page as shown in the picture

**Please select your favorite food!**

Beef  Burger  Salad

---

**Let us know how you want your food!**

Comments Here!

---

**Select your sauce!**

Sauces 

---

**Create an account with us in 3 easy steps!**

Creating an account allows you to track your order.

---

Email:

Username:

# Solution

```
<body>

    <h1>Welcome to Webstaurant!</h1>

    <h4>Please select your favorite food!</h4>

    <input type="checkbox" name="food" value="beef">Beef
    <input type="checkbox" name="food" value="burger">Burger
    <input type="checkbox" name="food" value="salad">Salad

    <hr>

    <h4>Let us know how you want your food!</h4>

    <textarea>Comments Here!</textarea>

    <hr>

    <h4>Select your sauce!</h4>

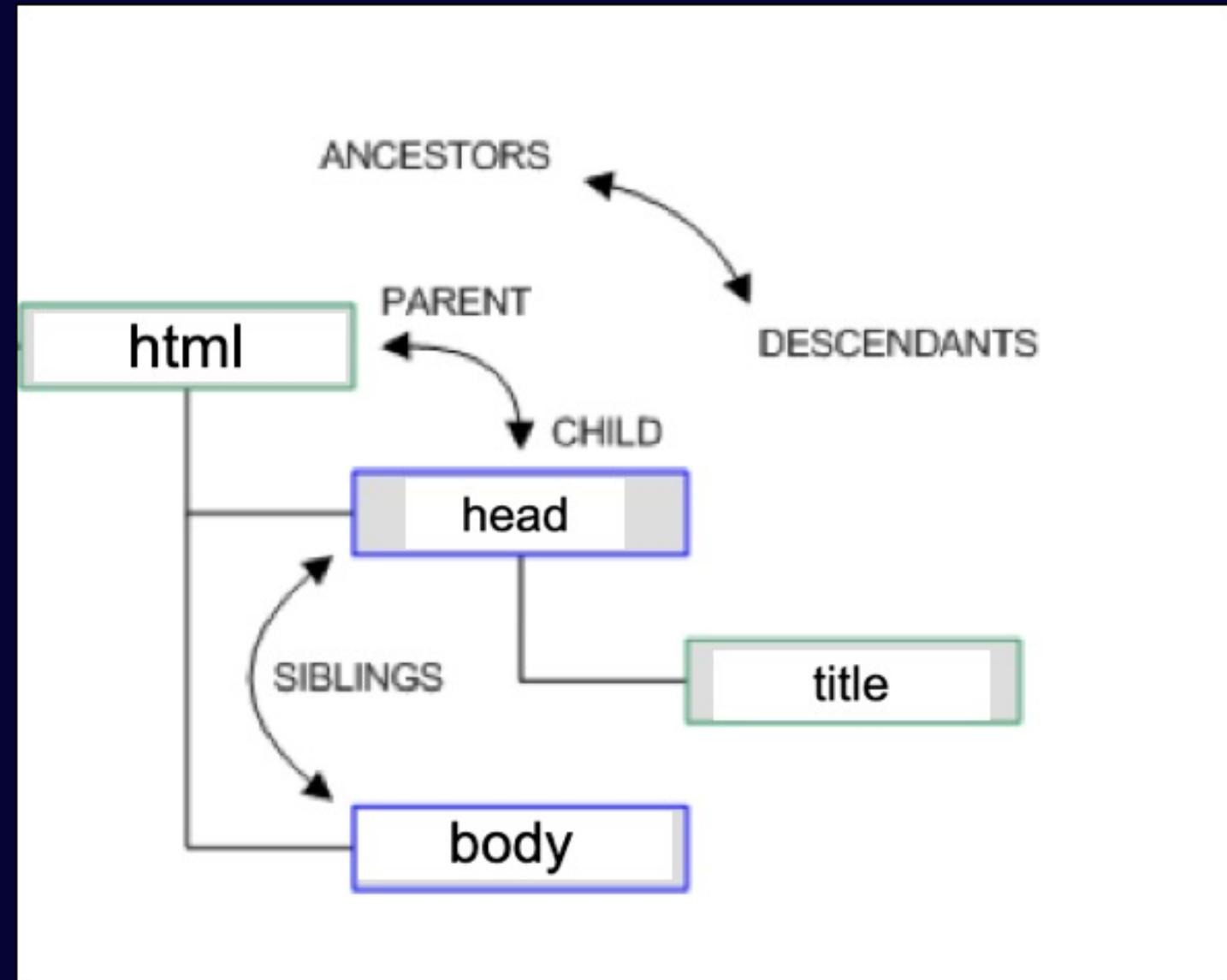
    <select>
        <option>Sauces</option>
        <option value="hot">Hot</option>
        <option value="mild">Mild</option>
        <option value="pepper">Pepper</option>
    </select>

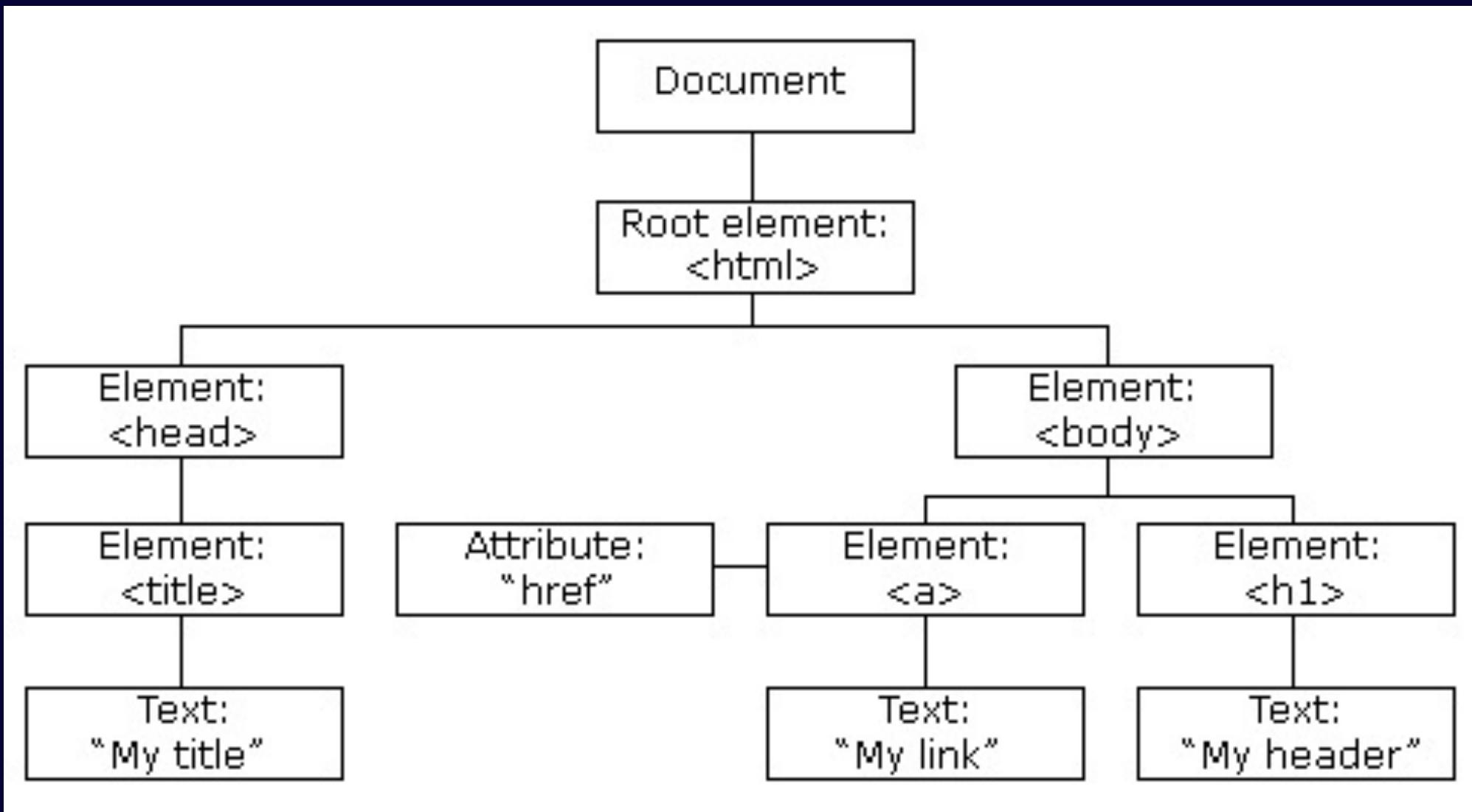
    <hr>
    <h3>Create an account with us in 3 easy steps!</h3>
    <p>Creating an account allows you to track your order.</p>
    <hr>
    Email: <input class="input-box" type="email" placeholder="Enter email">
    <br>
    Username: <input class="input-box" type="text" placeholder="Enter username">
    <br>
    Password: <input class="input-box" type="password" placeholder="Enter Password">
    <br>

    <a class="signup-button" href="#">Sign-Up</a>

</body>
```

## Parent and child relationship





Who is child,  
who is  
parent?

```
<h2>This is h2</h2>

<table>
  <thead>
    <tr>
      <td>Numbers</td>
      <td>Names</td>
      <td>Colors</td>
      <td>Fruits</td>
    </tr>
  </thead>

  <tbody>
    <tr>
      <td>441</td>
      <td>John</td>
      <td>Blue</td>
      <td>Orange</td>
    </tr>
    <tr>
      <td>553</td>
      <td>Jane</td>
      <td>Red</td>
      <td>Apple</td>
    </tr>
  </tbody>
</table>
```

## CSS ZenGarden – Now it might make more sense what css is.

- The whole purpose of this website is to show us what you can accomplish with different types of CSS Styles.
  - Every single one of these different versions have the EXACT same html back-bones.
  - Only different CSS styles has been applied to change the presentation.
- 
- <http://www.csszengarden.com/>

# Let's do some CSS styling to our Webstaurant web app!

## Welcome to Webstaurant!

Please select your favorite food!

Beef  Burger  Salad

---

Let us know how you want your food!

Comments Here!

---

Select your sauce!

Sauces 

---

Create an account with us in 3 easy steps!

Creating an account allows you to track your order.

---

Email:

Username:

Password:

**Sign-Up**

# Solution

```
<head>
    <title>Westaurant</title>
    <style type="text/css">
        body{
            text-align:center;
            background-color: lightgray;
            font-family: sans-serif;
        }

        hr{
            width: 300px;
        }

        .input-box{
            line-height: 20px;
            border-radius: 5px;
            margin: 5px 0 15px;
        }

        .signup-button{
            text-decoration: none;
            color: #fff;
            border: 1px solid black;
            padding: 10px 20px;
            display: inline-block;
            margin-top: 10px;
            border-radius: 5px;
            background-color: #78807a;
        }

        .signup-button:hover{
            text-decoration: none;
            color: black;
            border: 2px solid black;
            padding: 12px 25px;
            display: inline-block;
            margin-top: 10px;
            border-radius: 5px;
            background-color: #37f065;
            font-weight: bold;
            transition: 0.5s;
        }

        textarea{
            border-radius: 5px;
            line-height: 20px;
            height: 40px;
            width: 200px;
            border: 2px solid;
        }
    </style>
</head>

<body>

    <h1>Welcome to Webstaurant!</h1>

    <h4>Please select your favorite food!</h4>

    <input type="checkbox" name="food" value="beef">Beef
    <input type="checkbox" name="food" value="burger">Burger
    <input type="checkbox" name="food" value="salad">Salad

    <hr>

    <h4>Let us know how you want your food!</h4>

    <textarea>Comments Here!</textarea>

    <hr>

    <h4>Select your sauce!</h4>

    <select>
        <option>Sauces</option>
        <option value="hot">Hot</option>
        <option value="mild">Mild</option>
        <option value="pepper">Pepper</option>
    </select>

    <hr>
    <h3>Create an account with us in 3 easy steps!</h3>
    <p>Creating an account allows you to track your order.</p>
    <hr>
    Email: <input class="input-box" type="email" placeholder="Enter email">
    <br>
    Username: <input class="input-box" type="text" placeholder="Enter username">
    <br>
    Password: <input class="input-box" type="password" placeholder="Enter Password">
    <br>

    <a class="signup-button" href="#">Sign-Up</a>
</body>
```

# Practice #15

- Adding JavaScript to our page
- Let's add a simple alert to our Sign-up button so when it is clicked user gets notified!

# Solution

```
<a class="signup-button" href="#" onclick="alert('Thanks for signing up!')">Sign-Up</a>
```

# ARE YOU ABLE TO ANSWER QUESTIONS BELOW

- What is a web element?
- What is a tag?
- What is an attribute?
- What is an attribute value?
- Which element is parent, which one is child?
- What are HTML, CSS, and Javascript used for

