Introduction to ReactJS

A Student-Run Short Course (SRC) conducted by the Student Academic Council in collaboration with the Technical Council.

Instructors:

- Reuben Devanesan
- Ananthu JP
- Kanishk Singhal

Logistics

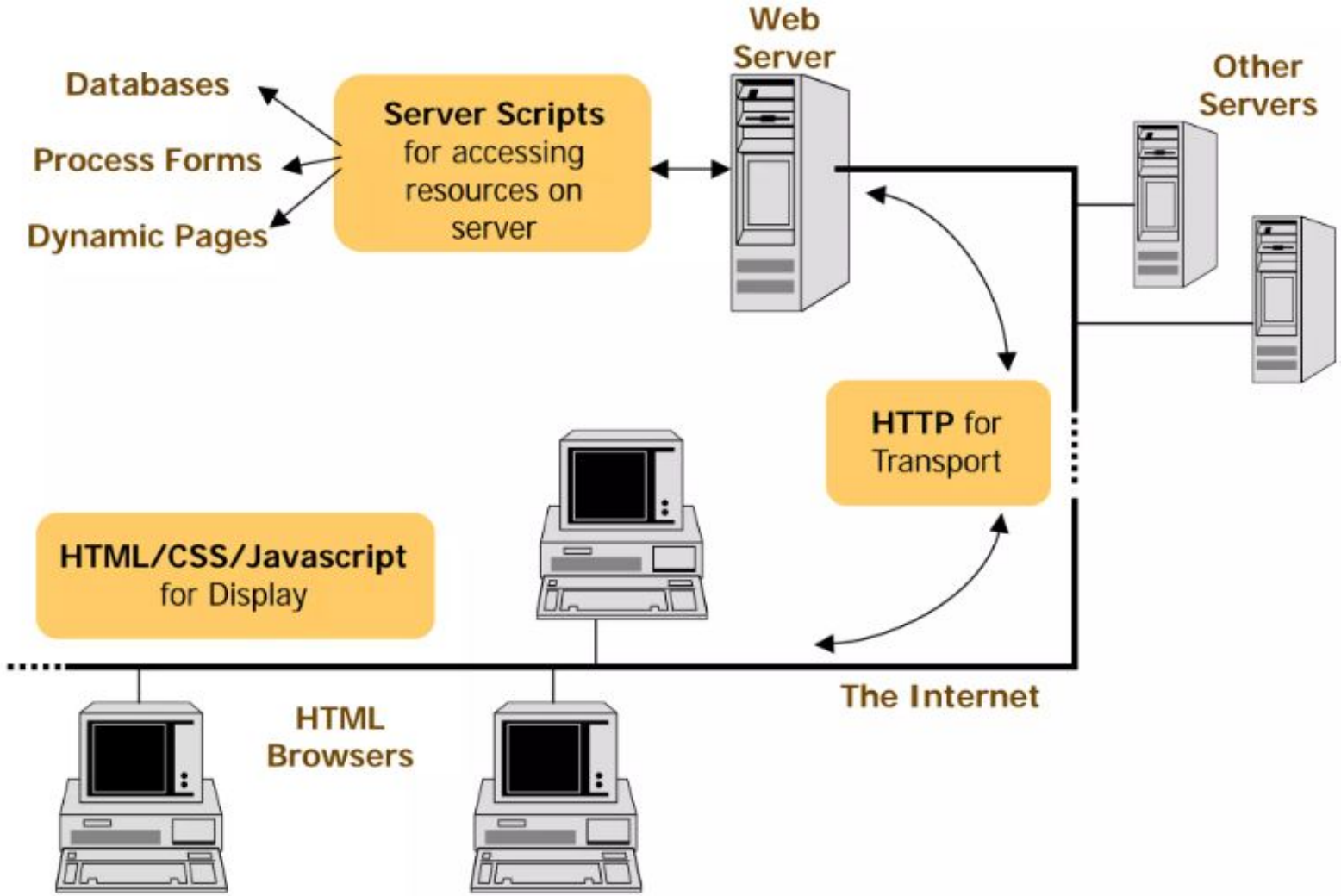
- 16th Oct to 4th Nov, 10 PM to 11 PM. AB 1/101, Learning Theatre.
- All important updates will be communicated through Google Classroom.
- All codes will be uploaded to the GitHub repository. Link: https://github.com/Reuben27/ReactJS-SRC
- Part I: https://github.com/Reuben27/Web-Development-SRC
- P/F course. Attendance compulsory.

What is Web Development? Why learn it?

- Creating, building and maintaining websites. It can range from working on a single page static site to managing complex web applications such as ecommerce sites, social networking sites, CMS, etc.
- Around 15.1 billion web enabled devices. Expected to almost double by 2030.
- Demand is still high for web developers in 2023. AI tools ain't perfect but can acts as tools to being productive.
- For any framework you use in the future, basics of how the web works, HTML, CSS is going to be required.

How the web works?

- The World Wide Web is all about communication between **clients and servers**.
  - Clients - browsers such as Chrome, Firefox.
  - Servers - computers hosted on the cloud.
- HyperText Transfer Protocol **(HTTP)** connects you and your website request to the remote server that houses all website data. It's an internet protocol for transmitting hypermedia documents such as HTML. Foundation of data communication for the web.
- Domain Name System **(DNS)** is a hostname for IP address translation service. Protocol for message exchange between clients and servers.


Web Server
Other Servers
Databases
Process Forms
Dynamic Pages
Server Scripts for accessing resources on server
HTTP for Transport
HTML/CSS/Javascript for Display
HTML Browsers
The Internet

What's the Difference?

HTML

Hypertext Markup Language

Create the structure

- Controls the layout of the content
- Provides structure for the web page design
- The fundamental building block of any web page

CSS

Cascading Style Sheet

Stylize the website

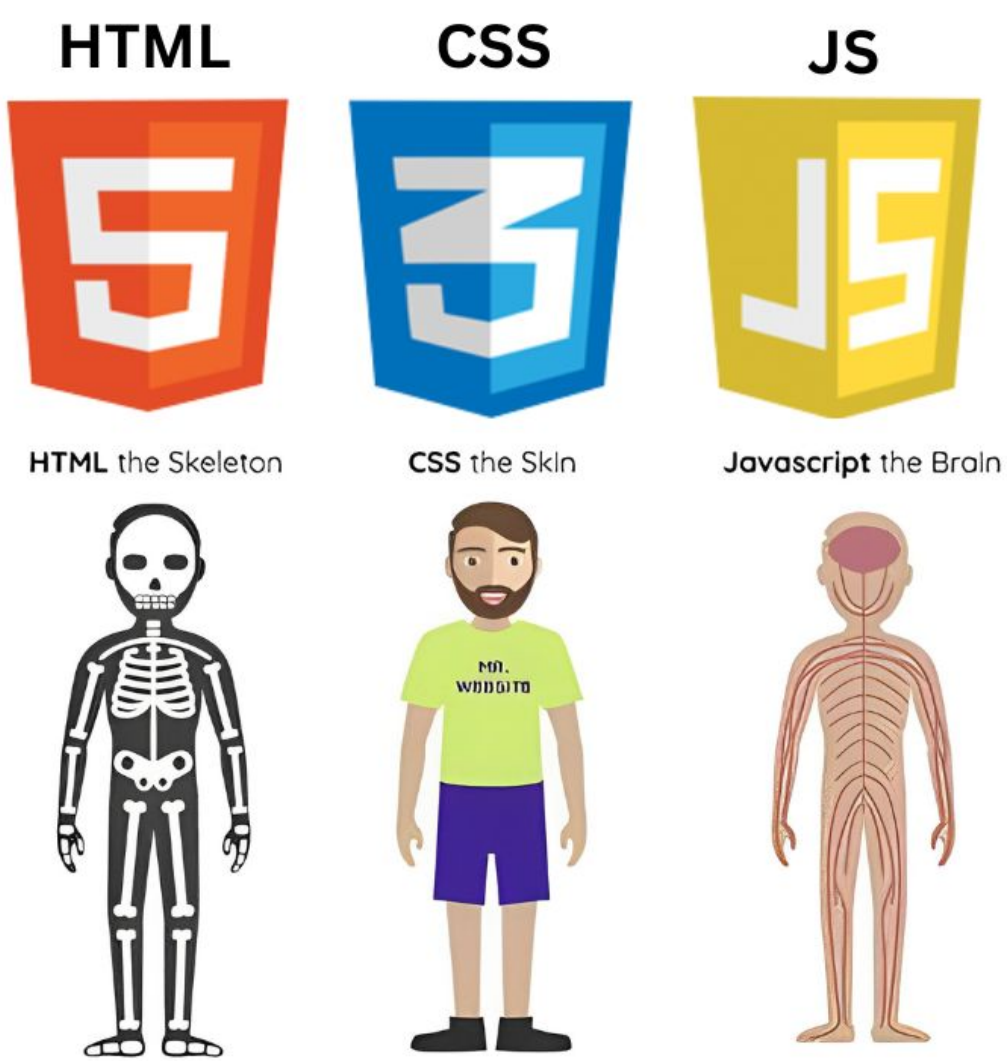
- Applies style to the web page elements
- Targets various screen sizes to make web pages responsive
- Primarily handles the "look and feel" of a web page

Javascript

Increase interactivity

- Adds interactivity to a web page
- Handles complex functions and features
- Programmatic code which enhances functionality

In Summary:


HTML
CSS
JS
HTML the Skeleton
CSS the Skin
Javascript the Brain

HTML (HyperText Markup Language)

- HTML is not a programming language but is a markup language. Markup languages prepare a structure for the data or prepare the look or design of a page. These are presentational languages and it doesn't include any kind of logic or algorithm
- A markup language is a set of markup tags.
- The tags in the HTML language are not displayed in browser, but the browser uses these tags to interpret the content of the page.
- HTML is not case sensitive.

Structure

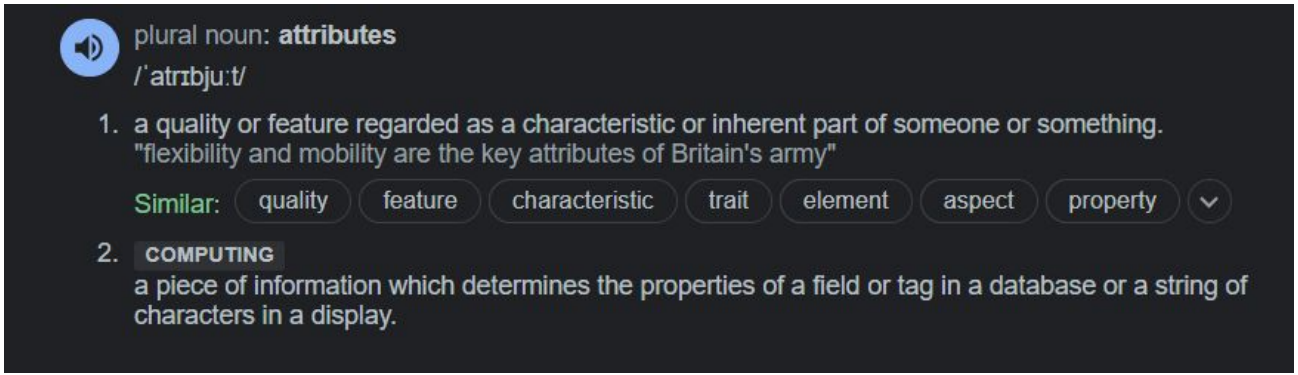
- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

```
<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

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

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

</body>

</html>
```

VS Shortcut for adding Structure (! or Shift+1)

HTML Attributes

- Attributes provide additional information about HTML elements. Always specified in the start tags.
- <img src="img_girl.jpg" width="500" height="600">



CSS : Casacading Style Sheet

1.You can add CSS to your webpage by:-

- Making a separate CSS file and linking it to the HTML file, or
- Adding CSS content using the style attribute/tag.

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

```
<h1 style="font-weight: bold; font-size: 42px;">
  I will be bold and 42 pixels!
</h1>

<p style="color: blue;">I will be blue!</p>
```

2. If you wish to add some style/effect to a certain element in the HTML file, you need to tag that element before applying changes to it.

There are different ways to tag an element like:

- Using Universal Selector. {Not used commonly}(*)
- Using element selector. For eg:- button, img, h1 etc.
- Using id and class tags.(#id {...}, .class {....})

3. Order of specificity: Inline styles>ID>class>element.

CSS Selector

Priority Order

Inline Style>ID>Class>Element

1 Element

```
div{
    height: ;
    background-color: ;
}
```

2 Class

```
.element-class{
    height: ;
    background-color: ;
}
```

3 ID

```
#element-id{
    height: ;
    background-color: ;
}
```

4 Inline Style

```
<h1 style="font-weight: bold; font-size: 42px;">
  I will be bold and 42 pixels!
</h1>

<p style="color: blue;">I will be blue!</p>
```

SELECTORS
CASCADE
SPECIFICITY
INHERITANCE
IMPORT
ATTRIBUTE SELECTORS
PSEUDO-CLASSES
PSEUDO-ELEMENTS
COLORS
UNITS
URL
CALC
Z-INDEX

BACKGROUNDS
COMMENTS
CUSTOM PROPERTIES
FONTS
TYPOGRAPHY
BOX MODEL
BORDER
PADDING
MARGIN
BOX SIZING
DISPLAY
POSITIONING
FLOATING AND CLEARING

CENTERING
LISTS
MEDIA QUERIES AND
RESPONSIVE DESIGN
FEATURE QUERIES
FILTERS
TRANSFORMS
TRANSITIONS
ANIMATIONS
NORMALIZING CSS
ERROR HANDLING
VENDOR PREFIXES
CSS FOR PRINT

JAVASCRIPT - BASICS

BASICS | Variables

1. console.log()

Object that writes a message to the browser's console,

2. alert

A method of the window object that displays a message box with a specified message and an OK button.

BASICS | Variables

1. var

Declarations are globally scoped or function scoped.
It can be updated or re-declared.

2. let

Declarations are block scoped.
Can be updated but not redeclared

3. const

It cannot be updated or redeclared

BASICS | Data Types

1. Undefined

A variable that has not been assigned a value.

2. Null

A variable that has been explicitly assigned the value null.

3. String

A sequence of characters enclosed in quotes.

4. Number

A numeric value.

5. Boolean

A logical value that can be either true or false.

BASICS | Arrays and Objects

Arrays

Syntax `const array_name = [item1, item2, ...];`

1. Add/Remove

push *- Add element to the end*
pop *- Remove element at the end*
shift *-Add element to the start*
unshift *- Remove element at the start*

2. Sort,Reverse

A numeric value.

Objects

Objects are variables too. But objects can contain many values.

Syntax

```
const object_name = {label1: value, label2, value, label3: function()};
```

BASICS | String Methods

1. Case

toUppercase *- Change the case to upper*

toLowercase *- Change the case to lower*

2. Substring

slice(*start,end****)*** *-*

split(*delimiter****)*** *-*

3. Find and Replace

find(*char****)*** *-*

replace(*searchValue,replaceValue****)*** *-*

replaceAll(*searchValue,replaceValue****)*** *-*

BASICS | Loops

1. for

Regular *for loop.*

```
for (expression 1; expression 2;
expression 3) {
  // code block to be executed
}
```

for of: *It loops through the values of an iterable object..*

```
for (variable of iterable) {
  // code block to be executed
}
```

2. while

The while loop loops through a block of code as long as a specified condition is true.

```
while (condition) {
  // code block to be executed
}
```

BASICS | Conditional Statements

1. if/else

if…else statement is used to execute a statement if a specified condition is truthy. If the condition is falsy, another statement in the optional else clause will be executed

```
if (condition) {
  //  block of code to be executed if the condition is true
} else {
  //  block of code to be executed if the condition is false
}
```

2. switch/case

Perform different actions based on different conditions

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

BASICS | Functions

The function is a block of code designed to perform a particular task.

Syntax

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

Thank you!