

### Notes :

1. Share a youtube link with the class for the next day's topic

### Week1 :

2. Introduction, Onboarding, Communication channels (Discord, Mail), Inspire
3. OS - Linux(Ubuntu),Windows & macOS
4. Tools(Installation)
  - a. VSCode
    - i. Extensions
  - b. Chrome
    - i. Extensions
  - c. Node / [Online Platform](#)
  - d. Git
5. Programming with Javascript
  - a. console.log

1. Print your name in given format

```
#  # ##### #  # #####
#  # #      #  # #      #
##### ##### #  # #      #
#  # #      #  # #      #
#  # ##### ##### #####
```

2. Javascript pyramid pattern

```
  *
 ***
*****
*****
*****
```

6. Git basics
  - a. Git add
  - b. Git status
  - c. Git commit
  - d. Git push
  - e. Git branch
7. Understanding the components of a computer
  - a. CPU

- b. RAM
- c. Processor
- 8. Bits & Bytes - Add, Sub, 2's complement

## **Week2(HTML) :**

- 9. Bhaukal
  - a. History
- 10. What is HTML? And use cases of HTML
- 11. HTML Tags & Attributes
  - a. <html>
  - b. <body>
  - c. <head>
  - d. <h1> to <h6>
  - e. <p>
  - f. <b>
  - g. <i>
  - h. <br/>
  - i. <hr/> [Day end]
  - j. Semantics
    - i. <header>
    - ii. <footer>
    - iii. <aside>
  - k. <img>
  - l. <button>
  - m. <a> & hover states
  - n. <div>
  - o. <span>
  - p. <ul> <ol> <li>[Day end]
  - q. <table>[Day end]
  - r. <input>
  - s. <form>[Day end]
- 12. What is a browser? & Type of browsers

## **Week3(CSS) :**

- 13. Bhaukal[Day end]
  - a. Maybe 3D, animations etc.
- 14. Play with font-size, color, background color
  - a. Basic Syntax with inline css
  - b. Assignment: Apply font-size, color, background on 3 different html tags
  - c. How do we avoid duplication?
- 15. Embedded css
  - a. Selector [Day end]
    - i. Tags
    - ii. Class
    - iii. Id
    - iv. Attribute selectors
  - b. Precedence

- c. Repeat the previous assignment without duplicating the design syntax
- 16. External CSS & linking [Day end]
- 17. Most common CSS properties
  - a. Padding, margin, border, Box model
  - b. Height, width, min & max[Day end]
  - c. Display
  - d. Position
  - e. letter-spacing, line-height
  - f. Opacity & Visibility
  - g. Background(Color & image)[Day end]

#### **Week4(Advance CSS) :**

- 18. Layout:
  - a. Grid
  - b. Flex
- 19. Pseudo elements
- 20. Learn How to Google & Importance of it
- 21. Bunch of live assignments
  - a. How to think designing of a page
  - b. Importance of responsive web pages
  - c. Media Query
- 22. Units of CSS Properties

#### **Week5(Practice):**

- 23. Create Youtube web page

#### **Week6(System Understanding):**

- 24. How a computer gets started?
- 25. The internet[day end]
  - a. <https://medium.com/@maneesha.wijasinghe1/what-happens-when-you-type-an-url-in-the-browser-and-press-enter-bb0aa2449c1a>
- 26. Browser developer tools[day end]
  - a. How to find css of any div element?
  - b. Console
  - c. Browser Versions
  - d. <https://caniuse.com/>
  - e. [Autoprefixer](#)
- 27. HTML & HTML5[day end]
- 28. CSS & CSS3[day end]
  - a. Cool Properties of CSS
    - i. Linear gradient
    - ii. Radial gradient
    - iii. Transform
    - iv. Clipath, boxshadow
    - v. And more...

#### **Week[7-9] (Basics of Programming with Javascript):**

- 29. Programming Bhaukal

30. Programming basics

- a. Syntax - [Js Syntax](#)
- b. Data Types - [Data Types](#)
- c. Variables - [var, let, const](#)
- d. Operators - [Operators](#)
- e. Ternary Operator - [Conditional \(ternary\) operator](#)
- f. Conditional statements (If, else, else if, switch, break, continue)
- g. Loops (for, for/in, for/of, while, do/while)
- h. Functions - [Functions](#)
- i. Array or Lists - [Array](#)
- j. Map, Filter, Reduce, forEach
- k. Classes and objects
- l. Exception handling - try catch

*Start one coding problem everyday*

**Week[10-11] (Javascript with HTML & CSS):**

- 31. Connecting Javascript, HTML & CSS together
- 32. Dom Manipulation using Vanilla Javascript[Day end]
- 33. What are events and importance?
- 34. Most common Events[Day end]
  - a. onload
  - b. onclick
  - c. onchange
  - d. scroll
  - e. resize
  - f. onmouseout
  - g. onmouseover
  - h. onkeydown
  - i. Onkeyup
- 35. Assignment: Guessing game
  - a. Generate a random number
- 36. Assignment: Form creation & validation
- 37. Higher-Order and Callback Functions
- 38. Promises
- 39. Asynchronous JS

**Week[12] (Getting started with React):**

- 40. Introduction/Bhaukal

- 41. ES5 vs ES6
- 42. Use React in Html, CSS & Js - Without Create React App
- 43. Fundamentals of React
  - a. Virtual Dom, Shadow Dom
  - b. React as a Library
- 44. Use React With Create React App
  - 1) Setup A Project in React
    - a) Architecture explanation
    - b) Codebase structure explanation
    - c) Dependencies
      - i) NPM
      - ii) Package.json

### **Week[13]**

- 2) What is components and how it work
- 3) What is state and how it work
- 4) Hooks (Basic)
  - a. useState
  - b. useEffect
  - c. useContext
- 5) What is props and how it work
- 6) How to use state and props together
- 7) Assignment: Create the gaming app using react

### **Week[14]**

- 8) Basics
  - a) Client
  - b) Server
  - c) Client-server communication
  - d) JSON
  - e) Rest APIs
- 9) APIs in React
- 10) Heavy application

### **Week[15]**

- 11) React Router / React Router Dom
- 12) Inspecting/developer's tools
- 13) Convert youtube clone into React
- 14) How to use localStorage & Cookie
- 15) Hooks (Advanced)
  - a. useReducer
  - b. useCallback
  - c. useMemo
  - d. useRef
  - e. useImperativeHandle
  - f. useEffect
  - g. useDebugValue

### **Week[16]**

- 16) Creating your own custom hooks
- 17) Forms - Formic and React Hooks Forms

### **Week[17]**

- 18) State Management via third-party libraries - Redux Saga
- 19) Caching, Cookies, local and session storage

### **Week[18]**

- 20) Styling : Styled Components, MaterialUI, Tailwind.css
- 21) React Testing
- 22) Styling / UI libraries
- 23) Handling APIs
- 24) Advanced React Topics
  - a. Higher-Order Components
  - b. Code-splitting
  - c. Refs
  - d. Context API
  - e. Server-side Rendering
  - f. React Suspense
  - g. React Server Components

### **Week[19-24]**

- 45. Basics of DS
- 46. Programming practice
- 47. Interview preparation
- 48. Project development

### **49. Practise Problem - Basic - [A Good Collection](#)**

- 1. Write a JavaScript program to compute the sum of the two given integers. If the two values are same, then returns triple their sum.
- 2. Write a JavaScript program to check two given numbers and return true if one of the number is 50 or if their sum is 50.
- 3. Write a JavaScript program to check whether a given positive number is a multiple of 3 or a multiple of 7.
- 4. Find if a given number is even or odd
- 5. Swap 2 number
- 6. Swap 2 number without using 3rd variable
- 7. Find the largest number among given 3 number
- 8. Reverse a number (+ve Integer)
- 9. Find largest number in an Array
- 10. Find second largest number in an Array
- 11. Find Sum of an array
- 12. Reverse an array(In place)
- 13. Side length of triangle is given check if its right angled or not.
- 14. Write a JavaScript program to find the area of a triangle where lengths of the three of its sides are 5, 6, 7
- 15. Write a JavaScript program to rotate the string 'w3resource' in right direction by periodically removing one letter from the end of the string and attaching it to the front.
- 16. Write a JavaScript program to determine whether a given year is a leap year in the Gregorian calendar.

17. Write a JavaScript program where the program takes a random integer between 1 to 10, the user is then prompted to input a guess number. If the user input matches with guess number, the program will display a message "Good Work" otherwise display a message "Not matched".
  18. Write a JavaScript program to convert temperatures to and from Celsius, Fahrenheit. [ Formula :  $c/5 = (f-32)/9$  [ where c = temperature in Celsius and f = temperature in Fahrenheit ]
  19. Write a JavaScript program to get the current date. - mm/dd/yyyy
  20. [Pattern Collections](#)
50. Languages

## **PAR102: Web Foundation**

- 1.
2. HTML5
3. CSS
  - a. Basic Selectors
  - b. Pseudo-classes and how to combine
  - c. Basics CSS
  - d. BOX Model
  - e. Flexbox
    - i. <https://flexboxfroggy.com/>
  - f. Grid
    - i. <https://www.gridgame.com/>
  - g. Display
  - h. Responsive web design
  - i. Positioning(relative, absolute)
  - j. Units(px, em, rem, vh....)
  - k. ...
4. Core Javascript
  - a. ES5 vs ES6
  - b. ...
5. Build A Page - [Youtube Clone](#)
6. React prerequisites
  - a. Variables in JS
  - b. Arrow Functions
  - c. Data Types and their methods
  - d. Dom Manipulation and events
  - e. Higher-Order and Callback Functions
  - f. Promises
  - g. Asynchronous JS
7. React
  - a. Use React in Html, CSS & Js - Without Create React App
  - b. Use React With Create React App
    - 25) Setup A Project in React
    - 26) What is components and how it work
    - 27) What is state and how it work

- 28) Hooks (Basic)
  - d. useState
  - e. useEffect
  - f. useContext
- 29) What is props and how it work
- 30) How to use state and props together
- 31) React Router / React Router Dom
- 32) Inspecting/developer's tools
- 33) Convert youtube clone into React
- 34) How to use localStorage & Cookie
- 35) Hooks (Advanced)
  - h. useReducer
  - i. useCallback
  - j. useMemo
  - k. useRef
  - l. useImperativeHandle
  - m. useLayoutEffect
  - n. useDebugValue
- 36) Creating your own custom hooks
- 37) Forms - Formic and React Hooks Forms
- 38) State Management via third-party libraries - Redux Saga
- 39) Caching, Cookies, local and session storage
- 40) Styling : Styled Components, MaterialUI, Tailwind.css
- 41) React Testing
- 42) Styling / UI libraries
- 43) Handling APIs
- 44) Advanced React Topics
  - h. Higher-Order Components
  - i. Code-splitting
  - j. Refs
  - k. Context API
  - l. Server-side Rendering
  - m. React Suspense
  - n. React Server Components