

Full Stack Development

React is best for single page application

href => is used to navigate in the same page to other pages.

nav => it is a semantic container in HTML5 used to define a block of navigation links

used to group navigation links logically and provide meaning to the document

<html> -> Root element

<head> -> It contains meta data like <title>, <meta>, <link>

<title> -> It set the title of the document

<body> -> It contains the content of the document

Works for the Day

Complete the 3 assignments

HTML 5 , CSS

Tailwind CSS

Why console and importance

Java Script :

It can be used in client side or service side

Powerful programming language it is equivalent to C and

C++ .

React and Angular from framework of JavaScript
or
libraries

HTML \Rightarrow Create static webpage

Javascript \Rightarrow Create dynamic

Note:

We can create script
+ inside the HTML body
+ outside the HTML body
+ separate file

Steps

Create \Rightarrow separate file and save as .js
In html file add script tag

with src

```
<script src="path">  
</script>
```

We can make alert message through
alert("Message")

We can make in separate file and add it in the
HTML file or through <script src="path">

JS datatypes

let var const

var (var num = 100)

Variable name num

Value is 100

(console.log(num) or alert(num))

Let

Note

We can print all three numbers in the same alert box using concatenation.

```
alert(" " + num + " " + b + " " + c)
```

Var Let → Block level scope

function scope

Const: value can't be changed

Ex:

```
function test() {
```

```
if (true) {
```

```
var a = 10;
```

```
let b = 20;
```

3

```
console.log(a);
```

```
console.log(b);
```

3

```
test();
```

Localscope globalscope

let

var

Reassignment ✓

Reassignment possible

Out

10

Error, Not defined

1) Do examples program for each of the following

1) if

2) if else

3) else if

4) else if ladder

5) Nested if

The Input:

No. of lemons in hand

Case 1:

No. of lemons in hand = 7

Expected output:

God 1 = 7 offered

God 2 = 7 offered need 7

God 3 = 7 offered need 7

Shortage: 14

Case 2:

No. of lemons in hand = 21

Ex O.P

God 1 = 7 off

God 2 = 7 off

God 3 = 7 off

sufficient

Case 3:

No. of lemons in hand = 15

Ex O.P

God 1 = 7

God 2 = 7

God 3 = ~~7~~ having 1 need 6

shortage: 6

Case 4:

No. of lemons in hand = 67

god 1 = 7 off

god 2 = 7 off

god 3 = 7 off

(21) $f(21 \text{ min})$

num = 21

sum =

$$\begin{array}{r} 37.5 \\ 15/2 \\ \hline 18.75 \end{array}$$

2.15

while(count > 7)

-91.5 10

1.1

31.5

12.75

5.625

$$\begin{array}{r} 18.75 \\ -15.00 \\ \hline 3.75 \end{array}$$

1

$$\frac{9}{15/2}, 15 - 1 = 14$$

8 while(count > 7)

 | god

 | result (count == 10)

 | god = 7

Ques

- 1) Sam having 75 candies, sam gives half of it to anu's friend angle. Since angle loves sam a lot she gives back half of her portion. calculate and display how many candies sam and angle have individually.

constraints:

Use one variable

use function only one time without argument, without return type

ES 6

ECMAScript 2015

New version of Javascript

CSS Style

<style>

button {

background-color: red;

color: white;

border-radius: 2px;

height: 10px;

width: 62px;

cursor: pointer;

border: none

Note:

In this method
we can only able to create
one button.

by using class we can
make separate style for
each button.

3

</style>

<style>

.class-name {

It same as above code

3

</style>

Arrow function - under es6

const ~~function.name~~ variable name = () => {
 return 100;
}

3

Arrow function

No function name it only has variable name

`document.getElementById("response").innerHTML = howareyou();`

↓ ↓ ↓ ↓
JS object Method It's an HTML element Property kind of function
 we call them by their id call

JS → JSON

System understand

The file we get from system will be JSON → JS

innerHTML

↓

property

* It's from ES6 for efficiency in terms of space until increase the readability, we can create function without name and its called as arrow function.

Simple Assign

Create simple calculator by getting two numbers as input and display addition, subtraction, product, quotient, remainder by creating individual Arrow function for the same

Ques

Create an array by taking array size and array element from the user. Extract all the perfect numbers and even prime numbers from the array.

Array

array.name.push(value) \Rightarrow add element in the beginning

array.name.pop() \Rightarrow delete element from the last

shift() \Rightarrow delete first element

• Shift(value) \Rightarrow add element at first

unshift(val1, val2, val3) -

push(val1, val2, val3, ...)

• splice(2, 2, 99)

index ↓
 replaceable value

No. of element

that should be deleted from the index

Ex:

arr = [1, 2, 3, 4, 5]

alert(arr) // 1, 2, 3, 4, 5

arr.splice(1, 2, 99)

alert(arr) // 1, 99, 4, 5

Slice()

It will slice the element from original array

slice(1, 4) \Rightarrow this one slice 1st, 2nd, 3rd index element
from array

oops

Object Oriented Programming Structure

Ex:

Birds → class

Crow } Parrot } Objects of class

Eating } → behaviours = one methods

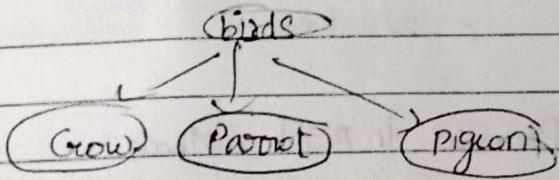
Speaking } ↓ core functions
flying }

colour } → properties

wings

legs

weight



class is a blueprint

without object there is no use of blueprint

Building a house first we need one model that is blueprint
without constructing house ,there is no use of blueprint

JS
Promises

It's an JavaScript (JS) objects.

Two states for promises:

- (i) Resolved (success) } They are callbacks
- (ii) Rejected (failure) }

Note:

Inverting a function

Keywords

then \rightarrow success resolve()
catch \rightarrow failure rejected()

Promise will be used when we are using API

Get time out function

Write a promise called Andhra, ~~B~~P, p1, p2, p3,
~~P1~~ \rightarrow A reached distance andhra to A 5000,
B reached andhra to B 2000, andhra to C 1000
c - reached

Promise in build methods

When there is more than one promise in order to
review then we can use promise inbuilt methods
according to the requirement.

Methods

- Promise.all([]) .
- Promise.any([]) .
- Promise.allSettled() .
- Promise.race() .

Promise.all

Once it sees a promise - false it will stop.

Promise.any

Gives the shortest true time promise, provided status should be true

Promise.allSettled

This one gives the keys and values in the dictionary

Will display ~~these three states~~ one among
in fulfilled
(i) Rejected
(ii) Pending

Promise.race

It will give the shortest value if there is no false

If there is any one false, it will show the ~~false~~ ^{first} value

Closure

It's like nested function
function inside another function

React JS

Netflix → React

Wikipedia → HTML

It's a library or framework of Javascript

Ex:

Netflix ↗ When we need user interaction
Amazon ↗ getting input from user.

HTML Website

Wikipedia

Youtube

2 important folders in React

(i) public

(ii) src

3 important files:

index.css

index.html

index.js

index.css

Note:

As of now don't touch index files.

Note:

Do initially write your code in App.js

DOM Document Object Model

React follows/
React contains V-Dom \Rightarrow Virtual DOM

Here Unlike HTML here once DOM gets created
the changes or manipulations what we do gets completed
and only that part will be re-rendered.

Whereas in HTML every time we make change
entire dom will be re-rendered

In Web application (created by react) each and everything
is called as components

Types of Components:

- (i) functional component
- (ii) class component

JSX

Javascript XML

Normally we write Javascript inside HTML

But in JSX we can write HTML in Javascript

Ex:

```
function App() {  
  return <h1>Hello world</h1>.  
}
```

3

2 para

1 image at top right

Props and States

Every component will have props and states.

Props:

It won't change

Ex: Brand

Name Tata's Bisleri

~~Brand~~

States:

It changes or We can change it.

Ex:

The water inside the Bisleri bottle.

Initial state - full

Current state may be half or $\frac{3}{4}$, or $\frac{1}{4}$ or empty

Flipkart Website

HomePage

Grocery

Mobile

Components

fashion.

Mobiles

Component Name : Mobiles

Props: Name, version, price

State: Discount, stock available

Parsing props between components.

two files:

parent.js

child.js

Hooks

Earlier in IT industries

state concept was not available with functional components.

Now, hooks is used to implement state in functional components.

(i) useState

(ii) useEffect

(iii) useRef

(iv) useContext

(v) useReducer

Best example for useState: counter clock

stating initial state as 0, we can incremented, decremented reseted using useState hook.

↳ useState:

↳ Returns a ^{set of} value