JS (Part – 2)

1. DOM manipulation
    - Document Object Model
    - Tree structure: node: element, text, attribute, comment
    - document.getElementById();

```
<h1 id="abcd">Hello World</h1>
```

```
let abcd = document.getElementById("abcd");
console.log(abcd);
```

    - document.getElementsByClassName();

```
<h1 class="abcd">Hello World</h1>
```

```
let abcd = document.getElementsByClassName("abcd");
console.log(abcd);
```

    - document.querySelector();

```
<h1 class="abcd">Hello World</h1>
```

```
let abcd = document.querySelector("h1");
console.log(abcd);  //or console.dir(abcd);
```

    - document.querySelectorAll();

```
<h1 class="abcd">Hello World</h1>
```

```
let abcd = document.querySelectorAll("h1");
console.log(abcd);  //or console.dir(abcd);
```

    - change hello world to hello suhas (innerHTML, innerText, textContent)
    - sometimes if console.log(); does not show DOM in console: use console.dir();
    - innerHTML: used to change html
    - innerText: used to change text
    - textContent: used to change text

2. Attribute Manipulation (getAttribute, setAttribute, removeAttribute)
    - // here href is attribute
    -

```
<a href="">Download</a>
```

```
let a = document.querySelector("a");
a.href= "https://google.com";
```

- (Or) element.setAttribute(attribute name, attribute value);

```
let a = document.querySelector("a");
a.setAttribute("href", "https://google.com");
```

-

```
<img src="" alt="image" />
```

-

```
let img = document.querySelector("img");
img.setAttribute(
  "src",
  "https://unsplash.com/photos/
  street-scene-at-dusk-with-traditional-japanese-buildings-gjIIkr9-8qc"
);
```

- element.getAttribute(name);

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

```
let a = document.querySelector("a");
console.log(a.getAttribute("href"));
```

- element.removeAttribute(name);

```
let a = document.querySelector("a");
a.removeAttribute("href");
```

3. Dynamic DOM manipulation (createElement, append, appendChild, remove, removeChild prepend)
    - Create element
    - Append/ prepend element
    - Append means after body it will show
    - Prepend means after body first only it will show eh: <h1> <a> = after appending <a>, it will become <a> <h1>
    -

```
<h1>Hello World</h1>
```

    - Append

```
let h1 = document.createElement("h1");
h1.textContent= "Hello Suhas";
document.querySelector("body").append(h1);
//or document.body.append(h1);
```

Output:

Hello World

Hello Suhas

- Prepend

```
let h1 = document.createElement("h1");
h1.textContent= "Hello Suhas";
document.querySelector("body").prepend(h1);
//or document.body.prepend(h1);
```

Output:

Hello Suhas

Hello World

- Remove();
  Used to remove element in DOM

4. Style (.style, classList (add, remove. Toggle)
   - We can change css and its styles & properties
   - ```
     <h1>Hello World</h1>
     ```
   - ```
     let h1 = document.querySelector("h1");
     h1.style.color="red";
     ```

     Output:

     # Hello World
   - Using classList
     i. ```
        <style>
            .jiji{
                color: brown;
                background-color: aqua;
            }
        </style>
        ```
        ```

```
<h1>Hello World</h1>
```

```
let h1 = document.querySelector("h1");
h1.classList.add("jiji")
console.dir(h1);
```

Output:

Remove is used to remove
Toggle is used to add if not added and is removed when it is added

5. Events and events handling
    - Element.addEventListener("event name", function() { });
    - Click: (Single Click)

```
<p>Click me</p>
```

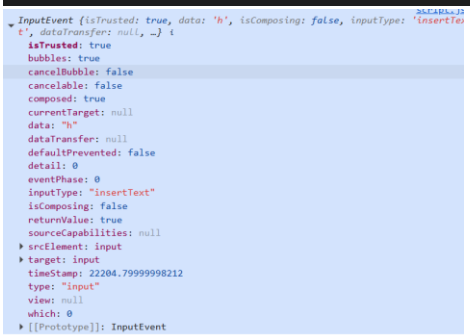
```
let p = document.querySelector("p");
p.addEventListener("click", function() {
    p.style.color="red";
})
```

    - (Double Click)

```
let p = document.querySelector("p");
p.addEventListener("dblclick", function() {
    p.style.color="red";
})
```

    - removeEventListener: is used to remove event added
    - input:

```
let inp = document.querySelector("input");
inp.addEventListener("input", function(dts) {
    console.log(dts);
}) //Output: every single character i type will be shown in object format
```



- in this eg: backspace when I click it is showing null

```
let inp = document.querySelector("input");
inp.addEventListener("input", function(dts) {
    console.log(dts.data);
})  //Output: every single character i type will be shown in console
```

- In this eg: when I click backspace it was showing null, after giving conditions, when I click backspace the data wont be showing as null

```
let inp = document.querySelector("input");
inp.addEventListener("input", function(dts) {
    if(dts.data != null) {
        console.log(dts.data);
    }
})  //Output: every single character i type will be shown in console
```

- Change:

```
<select>
  <option selected disabled>Choose Your device</option>
  <option value="samsung">Samsung</option>
  <option value="oppo">Oppo</option>
  <option value="apple">Apple</option>
</select>
```

```
let sel = document.querySelector("select");
sel.addEventListener("change", function(dts) {
    console.log(dts.target.value);
})
```

Output:

- Eg: 1 => selected device should display in text dynamically

```
<h1 id="syd">Select your device</h1>

<select>
  <option selected disabled>Choose Your device</option>
  <option value="samsung">Samsung</option>
  <option value="oppo">Oppo</option>
  <option value="apple">Apple</option>
</select>
```

```
let sel = document.querySelector("select");
let syd = document.querySelector("#syd");

sel.addEventListener("change", function(dts) {
    syd.textContent=`Your Selected Device: ${dts.target.value}`;
    console.log(dts.target.value);
})
```

Output:

- Keydown:

```
<div>
  <h1>Press any key</h1>
</div>
```

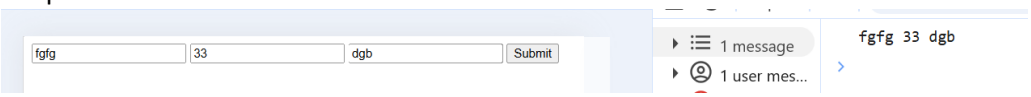
```
let h1 = document.querySelector("h1");

window.addEventListener("keydown", function (dts) {
  h1.textContent = dts.key;
});
```

- Submit:

```
<form>
  <input type="text" placeholder="name">
  <input type="text" placeholder="age">
  <input type="text" placeholder="place">
  <input type="submit">
</form>
```

```
let form = document.querySelector("form");
let inputs = document.querySelectorAll("input");

form.addEventListener("submit", function(dts) {
    dts.preventDefault();
    console.log(inputs[0].value, inputs[1].value, inputs[2].value);
})
```

Output:



- Eg: to show form values entered in screen

```
<form>
  <input type="text" placeholder="name">
  <input type="text" placeholder="age">
  <input type="text" placeholder="place">
  <input type="submit">
</form>

<h5>Hello</h5>
```

```
let form = document.querySelector("form");
let inputs = document.querySelectorAll("input");

let h5 = document.querySelector("h5");


form.addEventListener("submit", function(dts) {
    dts.preventDefault();
    h5.textContent= `Welcome ${inputs[0].value}, Age: ${inputs[1].value}, place: $
    {inputs[2].value}`;
    console.log(inputs[0].value, inputs[1].value, inputs[2].value);
})
```

Output:

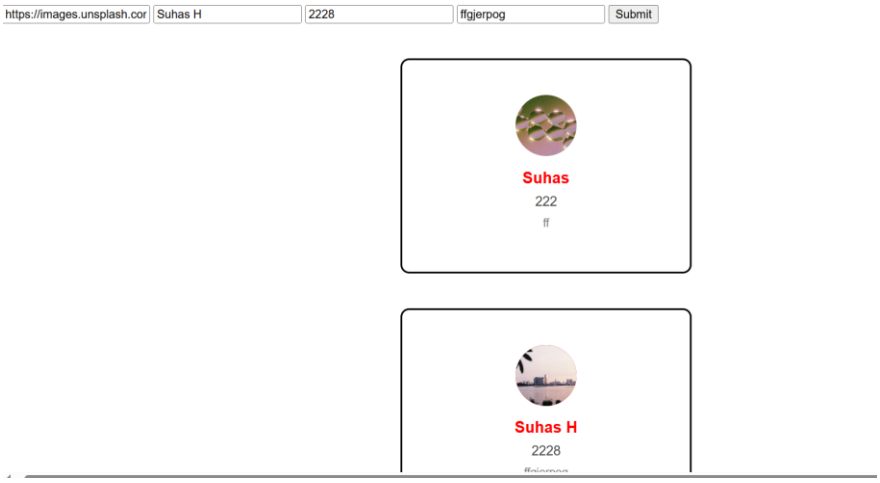
Welcome suhas, Age: 22, place: Bangalore

- Eg: when I click on submit the values will be shown below in card and if I add more details , the other added details will also be added.

```
<div id="main">
  <form>
    <input type="text" placeholder="profile pic" />
    <input type="text" placeholder="name" />
    <input type="text" placeholder="age" />
    <input type="text" placeholder="place" />
    <input type="submit" />
  </form>
</div>
```

```js
let form = document.querySelector("form");
let inputs = document.querySelectorAll("input");
let main = document.querySelector("#main");

form.addEventListener("submit", function(dts) {
    dts.preventDefault();

    let card = document.createElement("div");
    card.classList.add("card");

    let profile = document.createElement("div");
    profile.classList.add("profile");

    let img = document.createElement("img");
    img.setAttribute("src", inputs[0].value);

    let h3 = document.createElement("h3");
    h3.textContent= inputs[1].value;
    let h5 = document.createElement("h5");
    h5.textContent= inputs[2].value;
    let p = document.createElement("p");
    p.textContent = inputs[3].value;

    profile.appendChild(img);
    card.appendChild(profile);

    card.appendChild(h3);
    card.appendChild(h5);
    card.appendChild(p);

    main.appendChild(card);

})
```

Output:



- Mouseover: on hover
- Mouseout: on hover out

```html
<h1>Hello Suhas</h1>
```

```
let h1 = document.querySelector("h1");

h1.addEventListener("mouseover", function() {
    h1.style.color="red";
})

h1.addEventListener("mouseout", function() {
    h1.style.color="yellow";
})
```

Output:

Hello Suhas

Hello Suhas

- Mousemove:
- Event object, type, target, preventDefault

6. Event Bubbling
    - It searches for event listener, from child to parent.
    - If there is no event listener in child, and when child is clicked => it calls the parent event listener => if there is no event listener there also it will move to another parent event listener
    - Flow will be from bottom to top
7. Event capturing
    - When the parent event event listener is clicked, it checks for true value in child event listener +. If it is present the event listener will execute and if child event listener has also has event listener => that will also executed also false value will be executed
    - Flow will be from top to bottom

8. Forms and Form validation
    - If I don't enter characters more than 1, it shows error

```html
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Key Press</title>
  <style>
    #hide {
      display: none;
      color: red;
    }
  </style>
</head>

<body>
  <form>
    <input type="text" placeholder="name" id="name" />
    <small id="hide">Please enter your name between 2 to 50 characters</small>
    <br>
    <input type="email" placeholder="email" id="email" />
    <br>
    <input type="password" placeholder="password" id="password" />
    <br>
    <select>
      <option value="male">Male</option>
      <option value="female">Female</option>
      <option value="others">Others</option>
    </select>
    <br>
    <textarea placeholder="Tell us about ur self"></textarea>
    <br />
    <input type="submit" />
  </form>

  <script src="script.js"></script>
</body>
```

-

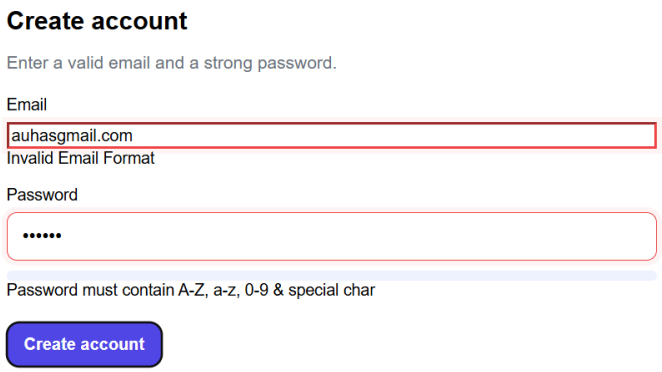
```js
let nm = document.querySelector("#name");
let form = document.querySelector("form");

form.addEventListener("submit", function (dts) {
  dts.preventDefault();

  if (nm.value.length <= 2) {
    document.querySelector("#hide").style.display = "initial";
  } else {
    document.querySelector("#hide").style.display = "none";
  }
});
```

- minLength, maxLength
- pattern: “[a-z]{3,8}”
- required
- autocomplete
- regex:
    i. used to check email validation, phone, name etc
    ii. Returns true or false

```
let email = document.querySelector("#email");
let password = document.querySelector("#password");
let form = document.querySelector("form");

form.addEventListener("submit", function (e) {
    e.preventDefault();

    const reEmail = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    const rePassword = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$%^&*(),.?":{}|<>]).{8,}$/;

    let emails = reEmail.test(email.value);
    let pass = rePassword.test(password.value);

    // reset messages
    document.querySelector("#emailMsg").textContent = "";
    document.querySelector("#pwMsg").textContent = "";

    // Email validation
    if (!emails) {
        document.querySelector("#emailMsg").textContent = "Invalid Email Format";
        email.classList.add("invalid");
    } else {
        email.classList.remove("invalid");
        email.classList.add("valid");
    }

    // Password validation
    if (!pass) {
        document.querySelector("#pwMsg").textContent = "Password must contain A-Z, a-z, 0-9 & special char";
        password.classList.add("invalid");
    } else {
        password.classList.remove("invalid");
        password.classList.add("valid");
    }
});
```

```
 <form>
   <!-- EMAIL FIELD -->
   <div class="field">
     <label for="email">Email</label>
     <input id="email" placeholder="you@example.com" required />
     <div class="msg" id="emailMsg"></div>
   </div>

   <!-- PASSWORD FIELD -->
   <div class="field">
     <label for="password">Password</label>
     <input id="password" type="password" placeholder="Create a password" required />

     <div class="strength" id="pwStrength"><i></i></div>

     <div class="msg" id="pwMsg"></div>
   </div>

   <!-- SUBMIT -->
   <div class="actions">
     <button type="submit" id="submitBtn">Create account</button>
     <div class="msg" id="formMsg"></div>
   </div>
 </form>
</section>
```

Ouput:



9. Timers and intervals
    - setTimeout(function() {}, time value);
        i. executes once
    - setInterval(function() {}, time value);
        i. executes again and again
    - clearInterval();
        - used to clear the interval
    -

```
let count = 10;

let interval = setInterval(() => {
    if(count >=1) {
        count--;
        console.log(count);
    } else {
        clearInterval(interval);
    }
}, 1000);
```

    Output:

```
8
7
6
5
4
3
2
1
0
```

10. Local storage, session storage
    - ~5MB
    - Set: localStorage.setItem(“name”, “Suhas”);
    - Get: localStorage.getItem(“name”);
    - Update: localStorage.setItem(“name”, “Hello”); //suhas to hello
    - Remove: localStorage.removeItem(“name”);

- Set: sessionStorage.setItem(“name”, “Suhas”);
- Get: sessionStorage.getItem(“name”);
- Update: sessionStorage.setItem(“name”, “Hello”); //suhas to hello
- Remove: sessionStorage.removeItem(“name”);

11. Cookies
    - 4kb
    - Temporary small data
    - When reload, it goes to the server along with data and brings the data
    - document.cookie=”email=suhas@gmail.com”;