

① Hoisting - It is a JS behavior in which a function or a variable can be used before the declaration.

↳ Type - Variable  
function.

② Event Bubbling - It allows the DOM elements to be nested inside each other.

by default event bubble is true.

⇒ Bottom to up the bubbles will come up.

```
<div>
  Parent
  <div> child
    <div> Grand child
      </div>
    </div>
  </div>
```

\* In such a case, if the handler of the child is clicked, then the handler of the parent will also work as if it were clicked too.

③ let, var & const.

⇒ var - declarations are globally scoped.

let & const - block scoped.

var - variables can be updated and re-declared within the scope.

let - variables can be updated but not re-declared.

const - can neither be updated nor re-declared.

④ Handle Exceptions -

```
try {  
    =  
} catch (err) {  
    =  
} finally {  
    =  
}
```

promise.resolve.  
promise.reject.

⑤ Append value in array  $\Rightarrow$  `myArr[arr.length]=myval;`

⑥ Variable typing = used to assign a value of some type and can be used to assign the different ~~type~~ value.

eg `a=120;`  
`a="Name";`

⑦ Delete operator - used to delete all the objects & variables but not the variable declared with var keyword.

⑧ Null - used to represent no value.

⑨ Undefined - has been declared but not assigned any value.

⑩ Void - prevent page from refreshing.

- ① This operator - refers to the owner of the object.
- ② callback - it is a plain javascript function passed to some method as an argument or the option.

Some callbacks are events that are called to give the user a chance to react when a certain state is triggered.

- ⑬ Closure - created whenever a variable that is defined outside the current scope is accessed from within the scope.

- ⑭ HTML elements can be accessed in JS -

- getElementById ( ' ' );
- getElementByClass ( );
- getElementByTagName ( );
- querySelector ( );

- ⑮ == → comparison operator  
=== → strictly check (value & type).

- ⑯ Typed language - Dynamically - can hold multiple values.  
Statically - can hold single value

- ⑰ Data types - undclined
- |         |        |
|---------|--------|
| Null    | Symbol |
| Boolean | number |
| String  |        |

18) Create array in JS

```
var x = [ ];
```

```
var x = [1, 2, 3];
```

19) Yes, we can assign the anonymous function to a variable and then pass it as an argument to the another function.

20) Scope - Global - can be used anywhere in the code.  
Local - Only used in the function.

21) Every method - it check whether all the elements of the array satisfy the conditions.

22) Some method - it check whether some element of the array satisfy the condition or not.

23) Map - Key, value pairs.

- set - to set value to the map key.
- get - to get value from the map of key
- size

we can print key & value with the help of "for of" loop.

```
→ for (let [key, value] of mymap)
  console.log(key, value);
  }
```

```
for (let key of mymap.keys()) {
  console.log(key);
}
|
for (let value of mymap.values()) {
  console.log(value);
}
}
```

24) Spread operator -

it allows an iterable to expand in places where any no. of arguments are expected.

Whenever we have to pass the parameter and we don't know how many we are expecting.

```
arr = [...myArr].
```

25) filter unique value from array -

we can use set along with spread operator.

```
const users = ['ram', 'mark', 'ram'];
```

```
const unique = [...new Set(users)];
```

26) Map - returns - value

27) ES6 features -

→ let & const keyword

→ Default Parameters

→ Arrow functions

→ Template literals

→ Object literals

→ Rest & spread operators

→ Destructuring

→ Modules, classes, generators & iterators

→ Promises.

28) Prototype - allows us to add new properties & method to the object constructors.

29) Template literals -

Are string with the embedded code and variables inside them.

→ use with backticks ` `

30) array.unshift()

→ adds the new element to the beginning of the array.

31) array.shift() -

→ removes the first element from the beginning of the array.

32) array.pop() -

→ removes the last element from the end of the array

33) array.push() -

→ adds the new element to the end of the array.

34) IIFE -

Self invoking functions -

function invoked automatically in the code.

```
(function() {  
  console.log('Hi');  
})();
```

### 35 Empty an Array in JS

- ① `arrayList = [];`
- ② `arrayList.length = 0;`
- ③ `arrayList.splice (0, arrayList.length);`
- ④ `while (arrayList.length) {`  
    `arrayList.pop ();`  
}