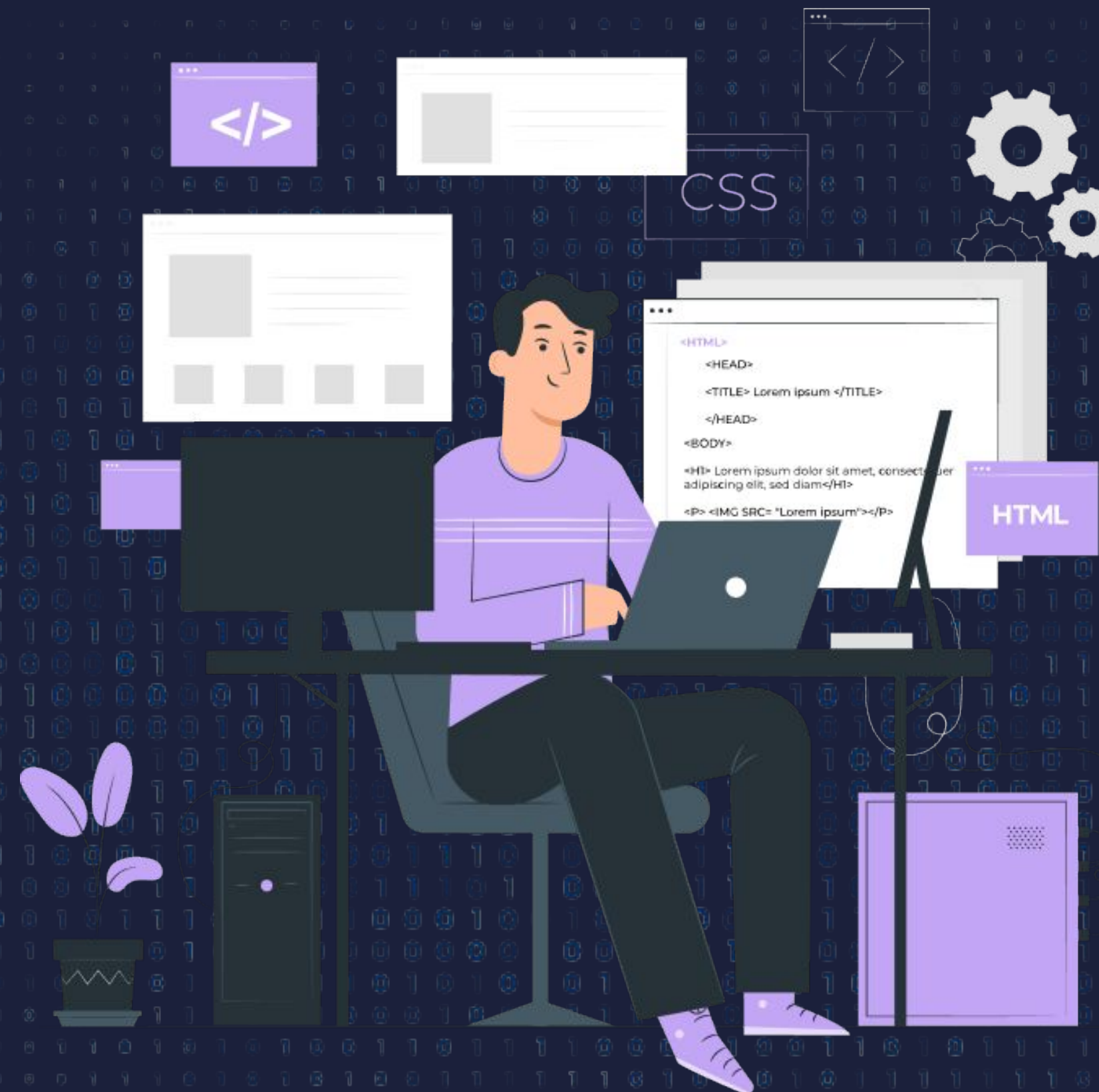




Object and Manipulating Object





Lecture CheckList

- Define Object in JavaScript
- Creating an object
- Manipulating values in an object



Define Object in JavaScript

In JavaScript, objects can be seen as a collection of properties. It is a collection of key-value pairs, where each key is a string or symbol that uniquely identifies a value. It is a fundamental data type and plays a crucial role in the language as it allows you to present complex data structures and manipulate them.

Example of object –

```
const user = {  
  name: "Subham",  
  lastName: "Sahu",  
  city: "Bangalore"  
}
```



Creating an object

There are majorly 3 ways to create an object in javascript:

1. By object literal
2. By creating an instance of Object directly (using new keyword)
3. By using an object constructor (using new keyword)



By object literal

// Syntax

```
let object = {name1: value1, name2:value2..... nameN: valueN}
```

```
let emp = {  
  id: 101,  
  name: "Alex",  
  salary: 10000,  
};  
console.log(emp.id + " " + emp.name + " " + emp.salary);  
// output - 101 Alex 10000
```



By creating an instance of an Object directly

```
// Syntax --  
var objectname = new Object()
```

```
const emp = new Object();  
emp.id = 101;  
emp.name = "Alex";  
emp.salary = 10000;  
  
console.log(emp.id + " " + emp.name + " " +  
emp.salary);  
// output - 101 Alex 10000
```

Creating an instance of an Object directly is not recommended due to the following reasons–

- Lack of encapsulation
- Limited Flexibility
- Inconsistent object creation



By using an object constructor

```
function Emp(id, name, salary) {  
  this.id = id;  
  (this.name = name), (this.salary = salary);  
}
```

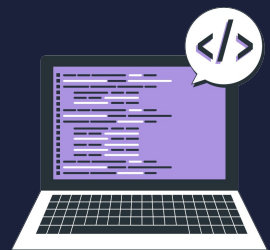
```
const emp = new Emp(101, "Alex", 10000);  
console.log(emp.id + " " + emp.name + " " +  
emp.salary);  
// Output - 101 Alex 10000
```



Manipulating values in an object

In data manipulation, we will cover the following -

- Accessing data
- Adding data
- Changing data
- Deleting data



Accessing data

In JavaScript objects, we can either use dot notation or square bracket notation to access object properties or alter values.

```
const emp = {  
  id: 101,  
  name: "Alex",  
  salary: "Doe",  
};  
console.log(emp.id);  
console.log(emp["name"]);  
// output -  
// 101  
// Alex
```



Adding data

It is simple to add additional key-value pairs to an existing object. Dot notation or square bracket notation can be used to accomplish that

```
//update --  
let emp = {  
  id: 101,  
  name: "Alex",  
  salary: 10000,};  
//Using dot notation  
emp.id = 102; // changing id for an emp  
console.log(emp);  
  
//Using brackets notation  
emp["name"] = "Sam"; // Changing name for an emp  
console.log(emp);  
  
/***** update object *****/  
{ id: 102, name: 'Alex', salary: 10000 }  
{ id: 102, name: 'Sam', salary: 10000 }
```




Deleting data

Data in an object can only be deleted with one method. It is done using the keyword `delete`.

```
let emp = {  
  id: 101,  
  name: "Alex",  
  salary: 10000 }  
emp.name = null;  
console.log(emp);  
delete emp.name;  
console.log(emp);  
/***** output *****/  
{ id: 101, name: null, salary: 10000 }  
{ id: 101, salary: 10000 }
```



Iterating Objects in Javascript

Data in an object can only be deleted with one method. It is done using the keyword `delete`.

In JavaScript, just like array an object can be also iterated. However, there are several approaches to loop over the properties of an object.

Which includes – `for...in`, `Object.keys()`, `Object.entries()`, `Object.getOwnPropertyNames()`.

Note – `Object.keys()`, `Object.entries()`, `Object.getOwnPropertyNames()` will studied in the object methods module.



For...in to iterate object

Data in an object can only be deleted with one method. It is done using the keyword `delete`.

```
const user = {  
  name: "Suham",  
  id: "1111",  
  depart: "Wed dev",  
};  
for (const key in user) {  
  console.log(user[key]);  
}  
// -- output --  
// Suham  
// 1111  
// Wed dev  
  
for (const key in user) {  
  console.log(`${key} : ${user[key]}`);  
}  
// -- output --  
// name : Suham  
// id : 1111  
// depart : Wed dev
```



▶ THANK YOU ◀