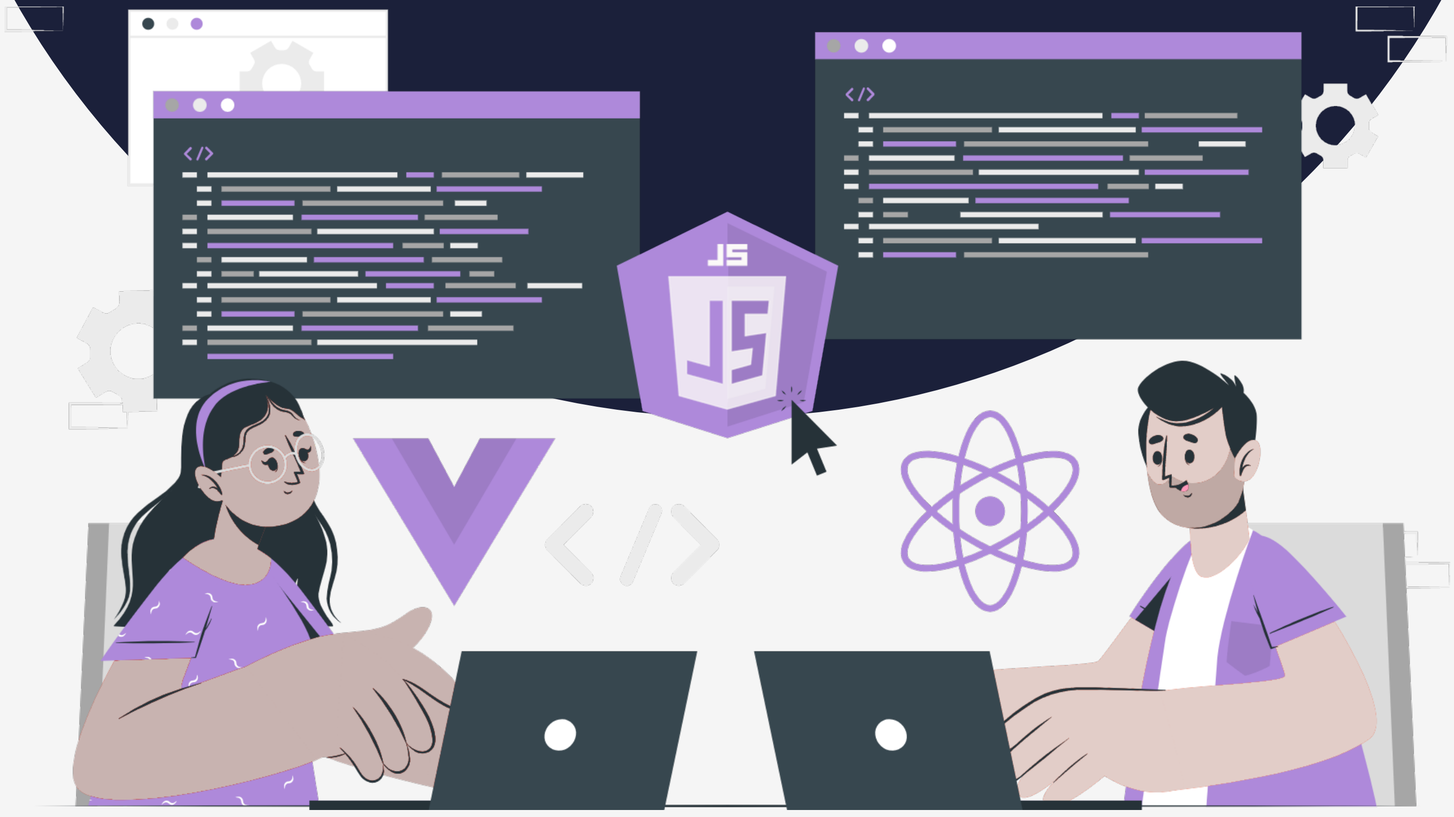
Lesson:

Creating an object and manipulating values in the object



PW SKILLS

List of content:

1. Creating an object
2. Manipulating values in an object

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)

Let us discuss each one of them here:

1. By object literal

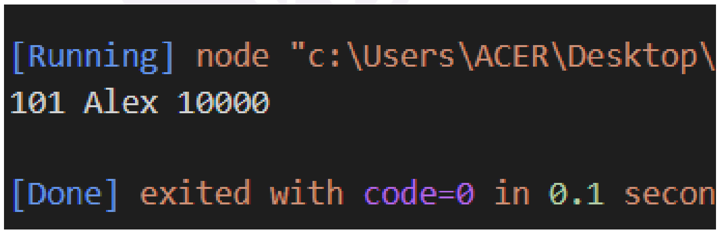
Syntax:

object={name1:value1, name2:value2.....name N:valueN}

As you can see, name and value are separated by a colon, this is how we create object with object literal method.

Let's see this simple example:

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



```
[Running] node "c:\Users\ACER\Desktop\
101 Alex 10000

[Done] exited with code=0 in 0.1 secon
```

2. By creating instance of object directly:

Syntax:

var objectname=new Object();

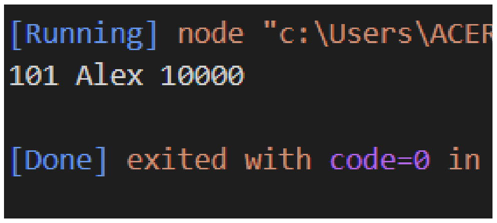
Here, **new** keyword is used to create object.

Let's see the example:

```
var emp=new Object();
  emp.id=101;
  emp.name="Alex";
  emp.salary=10000;
  console.log(emp.id+" "+emp.name+" "+emp.salary);
```

Here, we have used the new keyword to create the instance of the object emp. The individual values are assigned with the help of . operator with the object emp.

Output:



```
[Running] node "c:\Users\ACER
101 Alex 10000

[Done] exited with code=0 in
```

3. By using an object constructor

Here, we create a function with arguments.
Each argument value can be assigned in the current object by using **this** keyword.

this keyword refers to the current object.

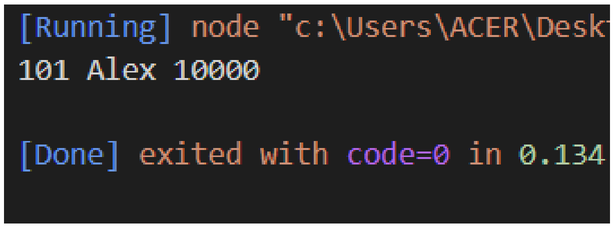
An example of creating object by object constructor:

```
function Emp(id,name,salary){
    this.id=id;
    this.name=name;
    this.salary=salary;
}
const e=new Emp(101,"Alex",10000);

console.log(e.id+" "+e.name+" "+e.salary);
```

Here, this keyword is used to assign values to the arguments id, name and salary.

Output:



```
[Running] node "c:\Users\ACER\Desk
101 Alex 10000

[Done] exited with code=0 in 0.134
```

Manipulating values in object

In data manipulation, we will cover the following:

- Accessing data

We can use either dot notation or square bracket notation to access object properties or alter values.

Since it is easier to read and use, dot notation is more commonly used.

Format for dot notation: objectName.propertyName.
Format for the square bracket notation: objectName['propertyName'].

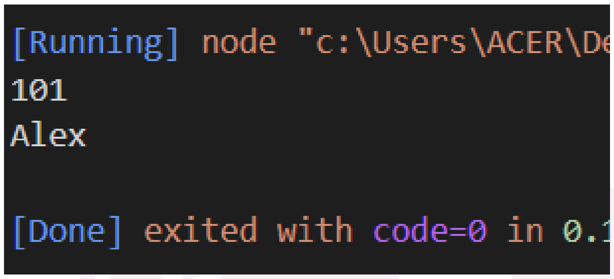
Use square bracket notation without quotes if you wish to use a variable as a property name.

Let us see an example:

```
const emp = {
  id: 101,
  name: 'Alex',
  salary: 'Doe'
}

console.log(emp.id);
console.log(obj['name']);
```

Output:



```
[Running] node "c:\Users\ACER\De
101
Alex

[Done] exited with code=0 in 0.1
```

- 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.

Let us see the example below to understand it better:

```
let emp = {
  id: 101,
  name: 'Alex',
  salary: 10000
}

emp.items = 10;
console.log(emp);

//Adding one more attribute to the emp object

emp['type'] = 'intern';
console.log(emp)
```

- Changing data

Sometimes all an object needs is a simple value change. We accomplish this using dot notation or square bracket notation, just as we do when adding new data.

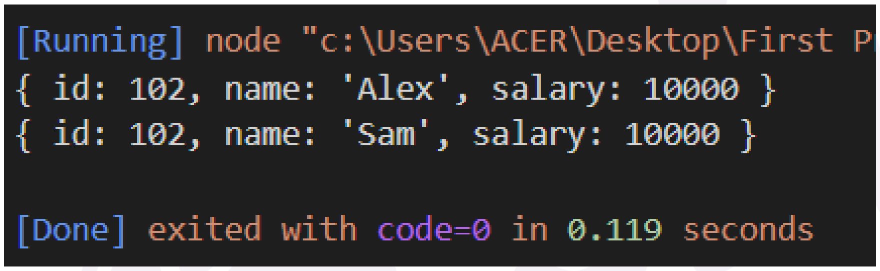
```
let emp = {
  id: 101,
  name: 'Alex',
  salary: 10000
}

//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)
```

Output:



```
[Running] node "c:\Users\ACER\Desktop\First P
{ id: 102, name: 'Alex', salary: 10000 }
{ id: 102, name: 'Sam', salary: 10000 }

[Done] exited with code=0 in 0.119 seconds
```

- Deleting data

Data in an object can only be deleted with one method. It is done using the keyword delete. Data is not lost if we use undefined or null; the key is kept but the value is altered.

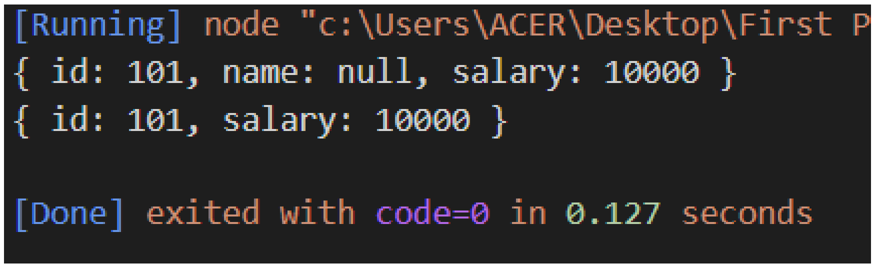
```
let emp = {
  id: 101,
  name: "Alex",
  salary: 10000
}

emp.name = null;

console.log(emp);

delete emp.name;
console.log(emp);
```

Output:



```
[Running] node "c:\Users\ACER\Desktop\First P
{ id: 101, name: null, salary: 10000 }
{ id: 101, salary: 10000 }

[Done] exited with code=0 in 0.127 seconds
```