

JavaScript Refresher | Complete React Course in Hindi #3

Before continuing our journey in React, let's recall some basic concepts of JavaScript.

JavaScript:

It allows client-side scripting to create completely dynamic web pages. It was designed to make web pages alive. With the help of Nodejs, Javascript can be used as a backend language.

Two ways to add JavaScript In HTML:

1. **Using `<script>` Tag** : Adding `<script></script>` tag and writing our JS code inside it.
2. **External File** : Adding a File with .js extension through link. The syntax is `<script src = "js/tut2.js"></script>`

Variables in JavaScript:

Variables are the memory location to store data. To declare a variable in Javascript, we use the following keywords:

- var
- let
- const

console.log():

This is used to print any defined variable or simply the text.

Arithmetic Operators:

These are used to perform any arithmetic function on a variable. Some examples are Addition, Subtraction, Multiplication, etc.

Datatypes in JavaScript:

There are two types of data types:

1. **Primitive:** Undefined, null, Number, String, Boolean, Symbol.
2. **Reference:** Arrays and Objects.

Functions in JavaScript:

It is a group of reusable codes which can be called anytime in the program. They are used to perform operations in the program. This eliminates the need to rewrite the same code. For Example:

```
function hello(){  
  console.log("Hey! I am Harry. ");  
}
```

On calling the above function by its name it will print "Hey! I am Harry." in the console.

Objects in JavaScript:

It can hold multiple values, arrays, functions, etc. For Example:

```
let obja = {  
  r: 34,  
  m: 64,  
  func: function myfunc(number) { console.log("The number is " + number) }  
}
```

Now, we can access each element of the Object individually. For example:

- **Accessing element:** To access the value of 'm' we will use obja.m
- **Calling function from the objects:** We can pass any number in the function, suppose 45. To call this function we can similarly use obja.func(45).

Events in JavaScript:

When the user interacts with the components in the page then the changes in the state of an object are termed as an event. There are many types of events like On clicked, On load, mouse over, mouse out, etc.

To fire the event, we need to listen to the event first for that we use an `addEventListener`. For example:

```
document.addEventListener("click", function click(){
  console.log("clicked");
  let conf = confirm("Are you Sure?");
  console.log(conf);
});
```

In the above code, On invoking the click event the click function is fired which shows the confirm modal to the user and prints clicked in the console.

Strings in JavaScript:

String stores text inside double and single quotes. For example:

```
let myString = "Harry is a good boy";
```

String methods:

- Length of a String
- Index of
- Last Index of
- Slice
- Replace and much more

Arrays in JavaScript:

Arrays are objects to store multiple values in a single variable. Example:

```
let arr = [1, 2, 3, "Harry"]
```

Array Methods:

- length
- forEach: arr.forEach(function)
- unshift, pop, push, toString, and so on.

Loops:

They are used to run a given set of codes until a certain condition is satisfied. Some examples of Loops are: For loop, Do while loop, while loop, etc.

Conditionals in JavaScript:

They are used to perform actions, like executing the code, based on the specific condition. Some types of conditionals are: If, else, else if, and switch case.

break and continue:

- **continue:** At a certain condition being met, It skips the current iteration and jumps to the second value.
- **break:** It stops the iteration when a certain condition is met.

Strict mode in JavaScript:

Strict mode imposed many restrictions in your JavaScript code. To execute your JavaScript in strict mode type "use strict".The advantage of using strict mode is that It ensures that a coding error doesn't occur in your program.

Arrow function:

Why do we need an arrow function? Let's understand with an example.

'This' keyword is a window object. For example, If you write `console.log(this);` it will be printed as a windows object. But inside a normal function, this windows object is termed as undefined. For example:

```
function myFunc(number){  
  console.log(this);  
}  
myfunc()
```

To overcome this problem we use the Arrow function.

```
let myFunc = (number)=> { console.log(this); }
```

In this case [object window] will be printed in the console.

Call back function:

It is the type of function which is invoked when a particular task has been completed. For example, `setTimeout` uses a callback function.

```
setTimeout(() => { console.log("Hooray") }, 3000);  
console.log("This is a script. ")
```

In the above code, "This is a script" will be printed in the console and after 3 seconds call back function is fired, and hence "Hooray" will be printed.

Asynchronous non-blocking IO model:

You might be wondering that you have written: "Hooray" before "This is a script" but still Javascript has executed it first. Why?

This happens because Javascript works on an Asynchronous non-blocking IO model, which means it doesn't block the second process due to the pending process.