

### 01 . Intro to javascript

- Js is a plAnd it allows developers to add dynamic and interactive effects to any webpage.
- We also use it to manipulate the content or the CSS, load data from remote servers ,and really build entire applications in the browser,
- There has been a huge update to the language in 2015, which is officially called ES2015, but most people just call it ES6,

### Conditional operator

```
syntax: condition:true
statement : false
statement
const age = 88;
age >= 18 ? consol e.l -
og( "you can") :conso -
le.l og ("you wont ");
```

### Template literals

```
const firstname =
"Sathiya";
const firstnamel = " -
Sat hiy aPr iya n";
const $new = ${firs tname
} is a guy ;
consol e.l og( $new);
```

Template literals - use variable in a string , use multiple line is easy

### p1-use stright mode

```
"use strict";
-> Use certian things and visible errors
-> and like that write more secure code. And when secure, I mean that strict mode makes it easier for us developers to avoid accidental error
```

### Array methods

```
// Adds
push - add an element in a end
unshift(): Adds elements to the beginning of an array.
//Remove
pop - remove last element of an array
shift(): Removes and returns the first element of an array.
//Find index
indexOf(): Returns the index of the first occurrence of a specified element ; minus value mean does not exit
//Check array exit or not
includes(): Checks if an array contains a specified element. - immutable
```

### Link js file

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

Link from another page , Script in same page simple use script tag

### Value and variable

Camal case

### Type Conversion and Coercion

nan - not a valid number or not a number  
Js - convert only string , boolean , number not other type

### True valuse and false value

5 falsy values in js  
0, "undefined , null , nan

### Version of javascript



### p2-Function declaration and expression (cont)

```
> // Anonymous function -
expressions is a vaule
const me = function (birthYear) {
  const age = 2029 - birthYear;
  return age;
};
```

### Object

```
// and it's called the
object literal Syntax
const spObject = {
  fir stName: " sat -
hiy a",
  las tName: " sp",
  age: 27,
  des tin ita tion:
" stu den t",
  habits: ["pr odu -
cti vit y", "time
manage men t"],
```

Always there is no way of giving these elements a name. **Apple** so we can't reference them by name, but only by their order number in which they appear in the array.

So we solve that problem we have another data structure in JavaScript, which is object

object is for unstructured data  
array is for ordered data

### p2-Function declaration and expression

```
"use strict";
// Function declar ation
function calAge 1(b -
irt hYear) {
  const age = 2029 -
birthYear;
  return age;
}
consol e.l og( cal Age -
1(2 003));
```



By sathiyapriyan\_j

Published 13th June, 2023.

Last updated 14th June, 2023.

Page 1 of 2.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>

### 3 ways of declaring a javascript

let, const is a modern way of declaring variable var is a old way of declarig of js

```
let age = 0 // Varaibale declare
age = 1 // it is a mutated variable
or reassign variable
```

let vs const -> const is a best practice .

The reason for this is that it's a good practice

to have as little variable mutations or variable changes as possible because changing variables introduces a potential bug

```
var is completely avoid
or declare nothing - bad practice
```

### Equality operator

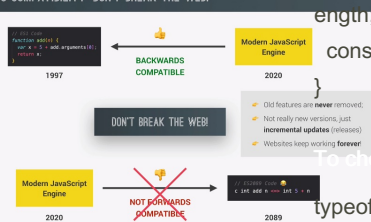
The strict equality operator (===) compares two values without type coercion

### Equality operator (cont)

while the loose equality operator (==) compares values with type coercion in JavaScript.is full of really weird rules and behaviors. this can introduce many hard to find bugs into our code. clean code

### Version-pic-2

BACKWARDS COMPATIBILITY: DON'T BREAK THE WEB!



### Put into any via loops (cont)

```
> for (let count = 0; count <=
years.length; count++) {
  console.log((age[count] =
calAge(years[count]]));
  newArray.push(age);
}
console.log("--");
console.log(newArray);
console.log(calAge(2003));
for (let count = 0; count < array.l
engine; count++) {
  const element = array[count];
```

### Check what kind of data type

typeof

### Data types

THE 7 PRIMITIVE DATA TYPES

1. Number: Floating point numbers. Used for decimals and integers.
2. String: Sequence of characters. Used for text.
3. Boolean: Logical type that can only be true or false. Used for taking decisions.
4. Undefined: Value taken by a variable that is not yet defined ('empty value').
5. Null: Also means 'empty value'.
6. Symbol (ES2015): Value that is unique and cannot be changed. (Not useful for now)
7. BigInt (ES2020): Larger integers than the Number type can hold.

JavaScript has dynamic typing. We do not have to manually define the data type of the value stored in a variable. Instead, data types are determined automatically.

### NULL is undefined data type

### operator

Check c notes  
Operator precedence So basically the order in which operators are executed.

### p3-Arrow functions

```
syntax : let func =
(arg1, arg2, ..., argN)
=> expression
It is introuded in ES6
vesrion
```

### Javascript dot vs bracket

<https://codepen.io/pen/tour/welcome/start?editors=1010>

### Put into any via loops

```
const yearsfa = [1991,
2007, 1967, 2020];
const age = [];
const newArray = [];
function calAge ($var) {
  return 2023 - $var;
}
```

### Get input from webpage

```
const $colour =
prompt(`Tell your
most favourite colour`);
consol e.l og( $co
lour);
```

### Version-pic-3

HOW TO USE MODERN JAVASCRIPT TODAY

- During development: Simply use the latest Google Chrome
- During production: Use Babel to transpile and polyfill your code (converting back to ES5 to ensure browser compatibility for all users).
- ES5: Fully supported in all browsers (down to IE 9 from 2011). Ready to be used today.
- ES6/ES2015: Well supported in all modern browsers. No support in older browsers. Can use most features in production with transpiling and polyfilling.
- ES2020: Can already use some features in production with transpiling and polyfilling.
- ESNext: Future versions of the language (new feature proposals that are not yet in production).

<https://kangax.github.io/compat-table/es6/>

### Calling function inside another

```
function cutFruit(fruit) {
  return fruit * 4;
}
function fruitP roc ess or
const appleP = cutFru
const orangesP = cutFr
const juice = Juice with
{oranges} oranges;
return juice;
}
```



By sathiyapriyan\_j

Published 13th June, 2023.

Last updated 14th June, 2023.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>