DEDICATED TO INNOVATION

www.bacancy.com

BACANCY®

Customer satisfaction is our highest priority.

Employee Matters

Core Values

Sustainability

Excellence

Innovation

BACANCY

Javascript

In this course

Agenda...

- Basic Introduction to javascript
- ES6 features
- Var, let and const
- regex, string and functions
- Arrays and Array funtions
- Conditional and Ternary operators
- What is DOM?
- What is DOM Manipulation

In this course

What is Javascript?

- JavaScript (JS) is a **high-level, interpreted** programming language primarily used to make web pages **interactive**.
- **Key Features of JavaScript:**

✔ **Client-Side Scripting** – Runs in the browser to enhance user experience
✔ **Dynamic & Event-Driven** – Responds to user actions (clicks, inputs, etc.)
✔ **Versatile** – Used for front-end (React, Angular) & back-end (Node.js)
✔ **Works with HTML & CSS** – Manipulates DOM to change content/styles

In this course

What is ES6

- **ES6 (ECMAScript 2015)** is a major update to JavaScript, introducing **modern syntax and powerful features** to make coding easier and more efficient.

Key ES6 features

1. `let` and `const` (Block-scoped variables)
2. Arrow functions (`=>`)
3. Template literals (`` `${}` `` syntax)
4. Default parameters
5. Destructuring assignment

In this course

6. Rest and spread operators (`...`)
7. Classes
8. Modules (`import` / `export`)
9. Promises
10. Enhanced object literals
11. For...of loop
12. Map and Set data structures
13. Symbol type
14. Generators
15. `includes()` for arrays and strings
16. Optional chaining (`?.`) *(introduced in later ES versions, often included)*

Var, Let and Const

JavaScript has **3 ways to declare variables**:

- var (old way - ES5 and before)
- const (Modern way ES6)
- let (Modern way ES6)

var

- `var` is function-scoped
- Can be **redeclared** and **updated**
- Gets hoisted (moved to the top of its scope)

Example

```
var x = 10;
var x = 20; // No error
console.log(x); // Output: 20
```

```
function test() {
  if (true) {
    var a = 5;
  }
  console.log(a); // Output: 5 (accessible outside block)
}
test();
```

let

- Introduced in ES6
- **Block-scoped**
- Cannot be redeclared in the same scope
- Can be updated

Example

```
let y = 10;
y = 15; // OK
// let y = 20; ❌ SyntaxError: Identifier 'y' has already been declared
```

```
if (true) {
  let z = 5;
  console.log(z); // Output: 5
}
// console.log(z); // ❌ ReferenceError: z is not defined
```

const

- Introduced in ES6
- **Block-scoped**
- Cannot be redeclared in the same scope
- Can be updated

Example

```
const PI = 3.14;
// PI = 3.14159; ❌ TypeError: Assignment to constant variable

const user = { name: "John" };
user.name = "Jane"; // ✅ Allowed (object properties can be changed)
```

Regular expression (RegEx)

RegEx stands for **Regular Expression,** It's a **pattern** used to match character combinations in strings, Helpful for **validation**, **searching**, or **replacing** text

Common Use Cases:

- Email or phone number validation
- Find/replace text in a string
- Check for specific patterns

Regular expression (RegEx)

- **Example 1: Test if a string is an email**

```
const email = "test@example.com";
const pattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
console.log(pattern.test(email)); // Output: true
```

- **Example 2: Replace all digits**

```
const str = "Order123";
const newStr = str.replace(/\d/g, "*");
console.log(newStr); // Output: Order***
```

String

A sequence of characters enclosed in " ", ' ', or backticks `

Method	Description	Example
length	Returns string length	"hello".length → 5
toUpperCase()	Converts to uppercase	"hello".toUpperCase() → "HELLO"
toLowerCase()	Converts to lowercase	"HELLO".toLowerCase() → "hello"
includes()	Checks for substring	"JavaScript".includes("Script")

In this course

Method	Description	Example
slice(start, end)	Extracts part of a string	"abcdef".slice(1,4) → "bcd"
replace()	Replaces part of a string	"Hi John".replace("John", "Jane")

Example

```
const msg = "Welcome to JavaScript!";
console.log(msg.includes("Java")); // true
console.log(msg.slice(11, 21));    // "JavaScript"
```

Functions

A block of code designed to **perform a task,** Can be reused multiple times

Example:

```
function greet(name) {
  return "Hello, " + name;
}
console.log(greet("Alice")); // Output: Hello, Alice
```

Arrow Function (ES6+):

```
const greet = (name) => "Hello, " + name;
console.log(greet("Bob")); // Output: Hello, Bob
```

Function Use Case: Add Two Numbers

```
function add(a, b) {
  return a + b;
}
console.log(add(2, 3)); // Output: 5
```

Array :-

An **Array** is a special variable that can hold **multiple values** in a single variable. Arrays are **ordered**, and values are accessed using **index numbers** (starting from `0`).

Example:-

```
let fruits = ["Apple", "Banana", "Orange"];
console.log(fruits[0]); // Output: Apple
console.log(fruits.length); // Output: 3
```

Common array methods :-

In this course

Method	Description	Example
push()	Adds item to **end**	fruits.push("Mango")
pop()	Removes **last** item	fruits.pop()
shift()	Removes **first** item	fruits.shift()
unshift()	Adds item to **start**	fruits.unshift("Kiwi")
indexOf()	Finds the index of an item	fruits.indexOf("Banana")
includes()	Checks if item exists	fruits.includes("Apple")
length	Gets the total number of items	fruits.length

Looping through array

Example: Using `for` loop

```
let colors = ["Red", "Green", "Blue"];
for(let i = 0; i < colors.length; i++) {
  console.log(colors[i]);
}
```

```
Red
Green
Blue
```

Array Higher-Order Functions

These functions take other functions as arguments and are commonly used for cleaner code.

`forEach()` – Loop through array

```
 let numbers = [1, 2, 3];
numbers.forEach(num => console.log(num));
```

`map()` – Create a new array by transforming each element

```
let nums = [1, 2, 3];
let squared = nums.map(n => n * n);
console.log(squared); // [1, 4, 9]
```

In this course

`filter()` – Filter elements based on condition

```
let nums = [1, 2, 3, 4];
let even = nums.filter(n => n % 2 === 0);
console.log(even); // [2, 4]
```

`reduce()` – Combine all elements into a single value

```
let nums = [1, 2, 3, 4];
let sum = nums.reduce((total, current) => total + current, 0);
console.log(sum); // 10
```

Conditional & Ternary Operators in JavaScript

Conditional statements let you **make decisions** in your code based on conditions (true or false).

Statement	Use when
if	You want to do something **only if** a condition is true
if...else	You want **one of two** actions
if...else if	You have **multiple conditions**
switch	You want to check **multiple fixed values**

Conditional & Ternary Operators in JavaScript

if...else

```
let age = 20;

if (age >= 18) {
  console.log("You are an adult");
} else {
  console.log("You are a minor");
}
```

✅ Output: `You are an adult`

Example: `if...else if...else`

```
llet score = 85;

if (score >= 90) {
  console.log("Grade A");
} else if (score >= 75) {
  console.log("Grade B");
} else {
  console.log("Grade C");
}
```

Output: `Grade B`

Switch case

The `switch` statement is used to **perform different actions** based on **different values** of a variable or expression. It's a cleaner alternative to writing many `if...else if` statements.

```
let day = 3;
let dayName;

switch(day) {
  case 1:
    dayName = "Monday";
    break;
  case 2:
    dayName = "Tuesday";
    break;
```

```
case 3:
    dayName = "Wednesday";
    break;
  case 4:
    dayName = "Thursday";
    break;
  case 5:
    dayName = "Friday";
    break;
  default:
    dayName = "Weekend";
}

console.log(dayName); // Output: Wednesday
```

Ternary Operators

A shorter version of `if...else`

Syntax: condition ? expressionIfTrue : expressionIfFalse;

Example:

```
let isLoggedIn = true;
let message = isLoggedIn ? "Welcome back!" : "Please log in.";
console.log(message);
```

Output: `Welcome back!`

DOM and DOM manipulation in javascript

DOM stands for **Document Object Model**.

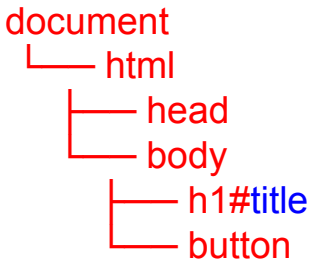
It is a **tree-like structure** that represents your web page.

With the DOM, JavaScript can **access and manipulate** HTML and CSS content **dynamically**.

Example:

```
<!DOCTYPE html>
<html>
  <body>
    <h1 id="title">Hello World</h1>
    <button onclick="changeText()">Click Me</button>
  </body>
</html>
```

DOM tree representation



DOM Manipulation

DOM Manipulation refers to using JavaScript to **access, change, add, or delete elements** and content on a web page dynamically.

- It's how JavaScript **interacts** with the structure and content of an HTML document.
- This allows developers to create **interactive and dynamic** websites.

Why Use DOM Manipulation?

- Change text, images, or styles on the fly
- Respond to user actions (like button clicks)
- Create or remove elements dynamically

Example: Change Text on Button Click

HTML

```
<h1 id="title">Hello!</h1>

<button onclick="changeHeading()">Click Me</button>
```

Javascript

```
function changeHeading() {

  document.getElementById("title").innerText = "Welcome to JavaScript!";

}
```

This changes the <h1> content when the button is clicked.

DOM Manipulation Methods

Method	Purpose
getElementById("id")	Select element by ID
getElementsByClassName("class")	Select elements by class
querySelector("selector")	Select first matching element
createElement("tag")	Create new HTML element
appendChild(element)	Add element to the DOM
removeChild(element)	Remove element from DOM
`innerText` / `innerHTML`	Get or set element content
setAttribute("attr", "value")	Set HTML attributes
style.property	Modify CSS styles

Example:

```
let p = document.createElement("p");
p.innerText = "This is a new paragraph!";
document.body.appendChild(p);
```

Tasks

Task 1:

- Write a function getPositiveNumbers(arr) that returns only positive numbers.
- Write a function getSquaredEvens(arr) that: Filters even numbers and returns an array with their squares.

Task 2:

Write a function getFee(isMember) that:

- Returns '$2.00' if isMember is true, else '$10.00'.
- Use a ternary operator.

Task 3 :

Build a **To-Do List** using JavaScript:

- Add tasks via an input field and "Add" button.
- Display tasks as a list.
- Each task should have a "Delete" button that removes the item.
- Bonus: Add "Mark as Done" toggle with strikethrough.
- Use DOM functions

Q&A

Thank You