**Java Script**

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Learn JavaScript

**The programming language of the Web**.

JavaScript is easy to learn.

This tutorial covers everything from basic JavaScript up to the latest 2025 version.

<!DOCTYPE html>

<html>

<body>

<h2>My First JavaScript</h2>

<button type="button"

onclick="document.getElementById('demo').innerHTML = Date()">

Click me to display Date and Time.</button>

<p id="demo"></p>

</body>

</html>

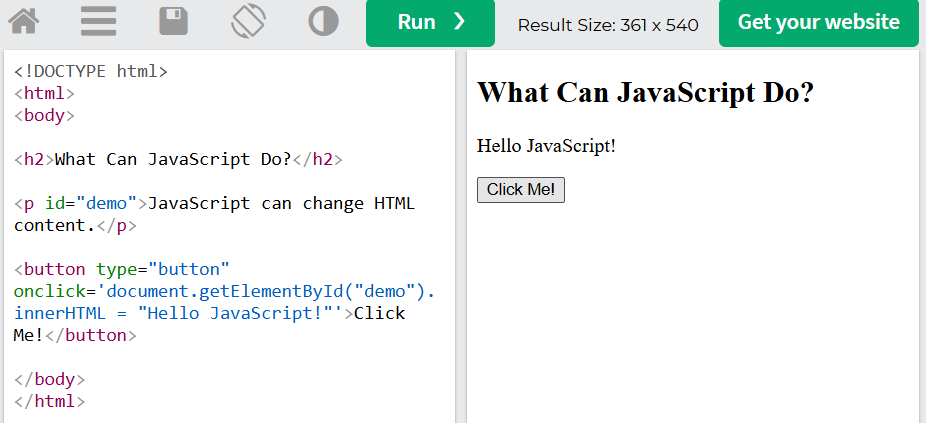
O/P:

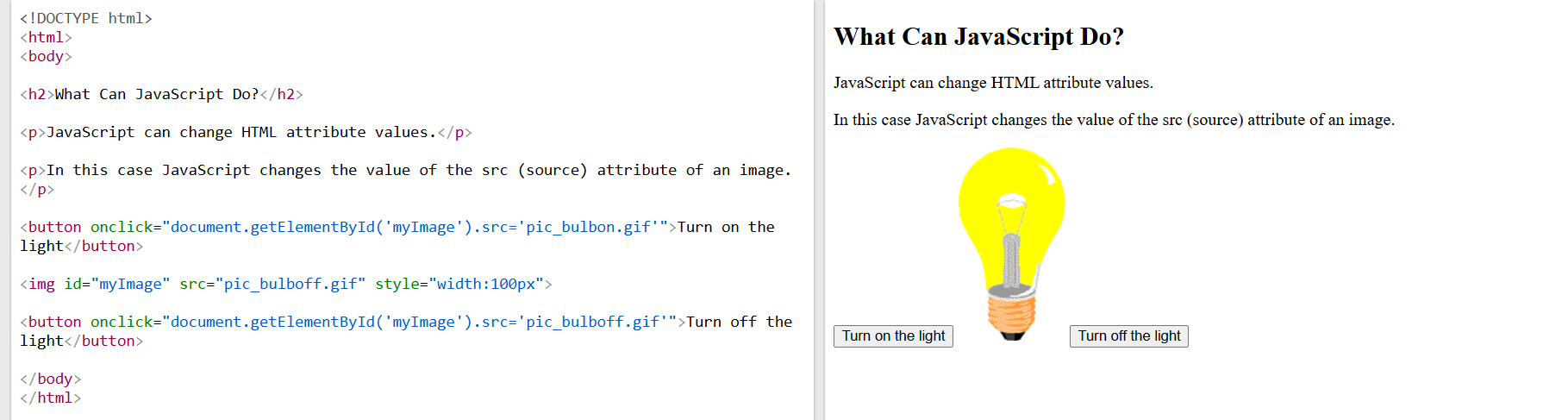
My First JavaScript

Click me to display Date and Time (After clicking here   
Mon Oct 06 2025 20:47:10 GMT+0530 (India Standard Time)

alert("Hello"); (Pop up)

alert("world");





🡪2+3

5

🡪alert(2+3);

undefined

🡪typeof(23);

'number'

🡪typeof("sumanth");

'string'

🡪 typeof(true);

'boolean'

Q)Swap of two numbers

function test() {

var a = "3";

var b = "8";

/\*\*\*\*\*\*\*\*\*\*\*Do not change the code above 👆\*\*\*\*\*\*\*/

//Write your code on lines 7 - 9:

var c=a;

a=b;

b=c;

/\*\*\*\*\*\*\*\*\*\*\*Do not change the code below 👇\*\*\*\*\*\*\*/

console.log("a is " + a);

console.log("b is " + b);

}

test();

## Naming and Naming Coventions Variables = Data Containers

**JavaScript variables** are containers for data.

JavaScript variables can be **declared** in 4 ways:

### Modern JavaScript

* Using let
* Using const

### Older JavaScript

* Using var (Not Recommended)
* Automatically (Not Recommended)

**<script>**

**let x = 5;**

**let y = 6;**

**let z = x + y;**

**document.getElementById("demo").innerHTML = "The value of z is " + z;**

**</script>**

**Concatination :**

**“a”+”b” = “ab”**

**🡪var message="Hello ";**

**🡪var name="sumanth";**

**alert(message+name) o/P: Hello sumanth**

**Length of a string**

🡪name.length o/p: 7

**🡪 characters remaining**

var tweet= prompt("Compose your tweet");

var tweetCount=tweet.length

alert("you have written "+ tweetCount + " characters ,you have " + (140-tweetCount) + " characters remaning");

**Slicing - var.slice(x,y)**

🡪var name="Angela";

name.slice(1,3); o/p:”ng”

name.slice(5,6) o/p:”a”

🡪var tweet= prompt("Compose your tweet");

var tweetUnder140=tweet.slice(0,140)

alert(tweetUnder140)

(or)

🡪alert(prompt("Compose your tweet").slice(0,140));

**toUppercase()**

🡪var name="Sumanth"

name=name.toUpperCase();

nam=name.toLowerCase();

**🡪 First letter capital sumanth o/p: Sumanth**

var name=prompt("What is your name?");

var firstChar=name.slice(0,1);

var upperCaseFirstChar=firstChar.toUpperCase();

var restOfName=name.slice(1,name.length);

var capitalisedName=upperCaseFirstChar+restOfName;

alert("Hello " + capitalisedName)

**🡪 // suppose sumaNth to Sumanth**

var name=prompt("What is your name?");

var firstChar=name.slice(0,1);

var upperCaseFirstChar=firstChar.toUpperCase();

var restOfName=name.slice(1,name.length);

restOfName=restOfName.toLowerCase();

var capitalisedName=upperCaseFirstChar+restOfName;

alert("Hello " + capitalisedName)

**Numbers**

Math.floor(4.33); O/P: 4

**Math.round(2.5); O/P:3**

**Functions**

[**https://stanford.edu/~cpiech/karel/ide.html**](https://stanford.edu/~cpiech/karel/ide.html)

function main(){

putBeeper();

next();

next();

next();

next();

}

function next(){

move();

turnLeft();

move();

putBeeper();

turnRight();

}

**🡪Life in Weeks Coding Exercise**

**function lifeInWeeks(age) {**

**var days = 365;**

**var weeks = 52;**

**var months = 12;**

**var remainingYears = 90 - age;**

**var remainingDays = remainingYears \* days;**

**var remainingWeeks = remainingYears \* weeks;**

**var remainingMonths = remainingYears \* months;**

**console.log("You have " + remainingDays + " days, " + remainingWeeks + " weeks, and " + remainingMonths + " months left.");**

**}**

**lifeInWeeks(56);**

**🡪**

**function bmiCalculator(weight,height){**

**var bmi=weight/(height\*height)**

**var round=Math.round(bmi);**

**return round;**

**}**

**var new1=bmiCalculator(65,1.8);**

**new1();**

**Random Number Generation -** Range from 0 inclusive up to but not including 1.

**🡪var n1=Math.random();**

**n1=n1\*6;**

**n1=Math.floor(n1);**

**console.log(n1);**

🡪

**prompt("What is your name?")**

**prompt("What is their name?")**

**var loveScore=Math.random()\*100;**

**loveScore=Math.floor(loveScore+1);**

**alert("Your love score is "+ loveScore + "%");**

**Control Statements**

🡪var age=prompt("enter your age:")

if (age>=18){

alert("He is adult");

}

else{

alert("he is kid");

}

## JavaScript Arithmetic Operators

Arithmetic operators perform arithmetic on numbers (literals or variables).

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| \*\* | Exponentiation ([ES2016](https://www.w3schools.com/js/js_2016.asp)) |
| / | Division |
| % | Modulus (Remainder) |
| ++ | Increment |
| -- | Decrement |

**Comparators and Equality**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | equal to | x == 5 |
| === | equal value and equal type | x === 5 |
| != | not equal | x != 5 |
| !== | not equal value or not equal type | x !== 5 |
| > | greater than | x > 5 |
| < | less than | x < 5 |
| >= | greater than or equal to | x >= 5 |
| <= | less than or equal to | x <= 5 |

JavaScript Assignment Operators

Assignment operators assign values to JavaScript variables.

Given that **x = 10** and **y = 5**, the table below explains the assignment operators:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Example** | **Same As** | **Result** |
| = | x = y | x = y | x = 5 |
| += | x += y | x = x + y | x = 15 |
| -= | x -= y | x = x - y | x = 5 |
| \*= | x \*= y | x = x \* y | x = 50 |
| \*\*= | x \*\*= y | x = x \*\* y | x = 100000 |
| /= | x /= y | x = x / y | x = 2 |
| %= | x %= y | x = x % y | x = 0 |
| : | x: 45 | size.x = 45 | x = 45 |

Logical Assignment Operators

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Result** |
| &&= | true &&= 10 | x = 10 |
| ||= | false ||= 10 | x = 10 |
| ??= | null ??= 10 | x = 10 |

**🡪 ==**

var a="1";

var b=1;

if (a==b){

console.log("equal");

}

else{

console.log("not equal");

}

O/P: equal

**🡪 ===**

var a="1";

var b=1;

if (a===b){

console.log("equal");

}

else{

console.log("not equal");

}

O/P:not equal

🡺

function isLeap(year) {

if ((year%4==0 && year%100!=0) || (year%400==0)){

return "Leap year.";

}

else{

return "Not leap year.";

}

}

**Combining Operaotrs**

**Arrays**

🡪Var myEggs=eggs[1];

🡪Var eggs=[11,22,33,44,66,77,88];

Console.log(eggs[0]);

🡪var ouput=[]

output.push(1);

output o/p;[1]

output.push(2); o/p:[1,2]

🡪

function whosPaying(names) {

var numberOfPeople = names.length;

var randomIndex = Math.floor(Math.random() \* numberOfPeople);

var randomPerson = names[randomIndex];

return randomPerson + " is going to buy lunch today!";

whosPaying(["Angela", "Ben", "Jenny", "Michael", "Chloe"]);

**Control Statements**

**While Loop**

🡪Var i=1;

While(i<5){

console.log(i);

i++;

}

🡪

var numberOfBottles=99

while (numberOfBottles>=0){

}

console.log( numberOfBottles+" bottles of beer on the wall, "+numberOfBottles+" bottles of beer.");

numberOfBottles--;

console.log("Take one down and pass it around, "+numberOfBottles+" bottles of beer on the wall.");

}

1. 🡪var numberOfBottles = 99
2. while (numberOfBottles >= 0) {
3. var bottleWord = "bottle";
4. if (numberOfBottles === 1) {
5. bottleWord = "bottles";
6. }
7. console.log(numberOfBottles + " " + bottleWord + " of beer on the wall");
8. console.log(numberOfBottles + " " + bottleWord + " of beer,");
9. console.log("Take one down, pass it around,");
10. numberOfBottles--;
11. console.log(numberOfBottles + " " + bottleWord + " of beer on the wall.");
12. }

**For Loop**

**🡪for (var i=0;i<2;i++){**

**// Do something**

**}**

🡪

function fibonacciGenerator(n) {

if (n <= 0) {

return [];

}

if (n === 1) {

return [0];

}

// n >= 2

var output = [0, 1];

for (var i = 2; i < n; i++) {

output.push(output[i - 1] + output[i - 2]);

}

return output;

}

In JavaScript, the terms "inline," "internal," and "external" refer to different methods of including and organizing JavaScript code within a web project.

1. Inline JavaScript:

Inline JavaScript involves embedding JavaScript code directly within an HTML element's attributes, typically as an event handler. This is often used for simple, single-purpose actions tied to a specific element.

Code

**<button onclick="alert('Hello from inline JavaScript!');">Click Me</button>**

2. Internal JavaScript:

Internal JavaScript involves placing JavaScript code within <script> tags directly inside an HTML file, either within the <head> or <body> section. This method is suitable for scripts that are specific to a single HTML page.

Code

<!DOCTYPE html>  
<html>  
<head>  
 <title>Internal JavaScript Example</title>  
 <script>  
 // Internal JavaScript code  
 console.log("Hello from internal JavaScript!");  
 </script>  
</head>  
<body>  
 <h1>My Web Page</h1>  
</body>  
</html>

3. External JavaScript:

External JavaScript involves storing JavaScript code in a separate .js file, which is then linked to the HTML document using the src attribute of the <script> tag. This is the preferred method for larger projects due to its advantages in organization, reusability, and maintainability.

Example of external.js:

JavaScript

*// external.js*  
console.log("Hello from external JavaScript!");

Example of HTML linking to external.js:

Code

<!DOCTYPE html>  
<html>  
<head>  
 <title>External JavaScript Example</title>  
 <script src="external.js"></script>  
</head>  
<body>  
 <h1>My Web Page</h1>  
</body>  
</html>

Key Differences and Use Cases:

* **Inline:**

Best for very small, element-specific actions. Can make HTML less readable if overused.

* **Internal:**

Suitable for scripts specific to a single page. Can become less organized for complex projects.

* **External:**

Recommended for most projects. Promotes code organization, reusability across multiple pages, improved maintainability, and better browser caching. It also separates concerns, keeping HTML, CSS, and JavaScript in their respective files.



Example

The following example changes the content (the innerHTML) of the <p> element with id="demo":

🡪<h2>My First Page</h2>

<p id="demo"></p>

<script>

document.getElementById("demo").innerHTML = "Hello World!";

</script>

**Selecting HTML with Javascript**

🡪document.firstElementChild.lastElementChild.querySelector("ul").lastElementChild.innerHTML = "Angela";

🡪document.getElementsByTagName("li");

🡪document.getElementsByTagName("li").length; o/p:3

🡪document.getElementsByTagName("li")[2].style.color="purple";

🡪document.getElementsByClassName("btn");

🡪document.getElementById("title");

🡪document.getElementById("title").innerHTML="Good Bye";

🡪document.querySelector("h1");

🡪document.querySelector("#title");

🡪document.querySelector(".btn");

🡪document.querySelector("li a");

🡪document.querySelector("li").style.color="red";

O/P:

* [Google](https://www.google.com/)

🡪document.querySelector("h1").style.color="red";

# Hello

🡪document.querySelector("h1").style.fontSize="10rem";

🡪document.querySelector("h1").style.padding="30%";

🡪document.querySelector("button").style.backgroundColor="yellow";

🡪document.querySelector("a");

document.querySelector("a").attributes;

document.querySelector("a").getAttribute("href");

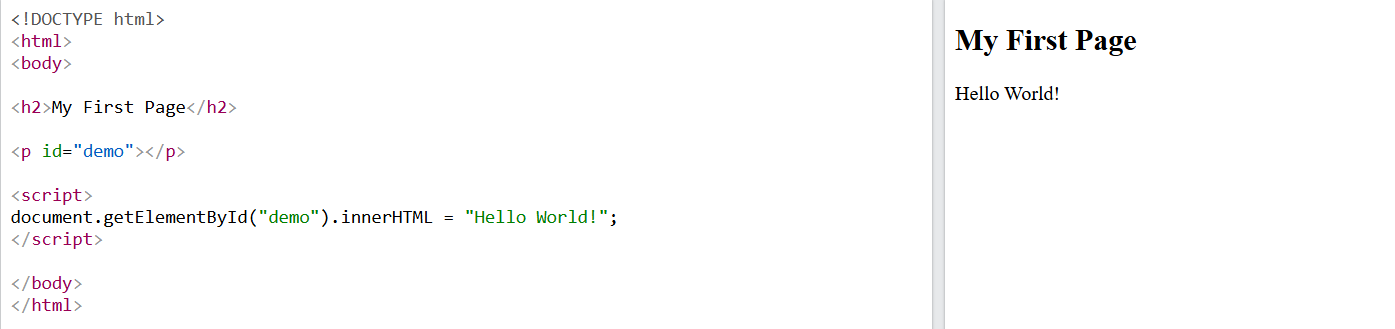
🡪$0.innerHTML="Angle Script";

🡪$0.addEventListener("click",function(){

console.log("I got clicked");

});

🡪



The HTML DOM Document Object

The document object represents your web page.

If you want to access any element in an HTML page, you always start with accessing the document object.

Below are some examples of how you can use the document object to access and manipulate HTML.

Finding HTML Elements

|  |  |
| --- | --- |
| **Method** | **Description** |
| document.getElementById(*id*) | Find an element by element id |
| document.getElementsByTagName(*name*) | Find elements by tag name |
| document.getElementsByClassName(*name*) | Find elements by class name |

Changing HTML Elements

|  |  |
| --- | --- |
| **Property** | **Description** |
| *element*.innerHTML =  *new html content* | Change the inner HTML of an element |
| *element*.*attribute = new value* | Change the attribute value of an HTML element |
| *element*.style.*property = new style* | Change the style of an HTML element |
| **Method** | **Description** |
| *element*.setAttribute*(attribute, value)* | Change the attribute value of an HTML element |

Adding and Deleting Elements

|  |  |
| --- | --- |
| **Method** | **Description** |
| document.createElement(*element*) | Create an HTML element |
| document.removeChild(*element*) | Remove an HTML element |
| document.appendChild(*element*) | Add an HTML element |
| document.replaceChild(*new, old*) | Replace an HTML element |
| document.write(*text*) | Write into the HTML output stream |

Finding HTML Elements

Often, with JavaScript, you want to manipulate HTML elements.

To do so, you have to find the elements first. There are several ways to do this:

* Finding HTML elements by id
* Finding HTML elements by tag name
* Finding HTML elements by class name
* Finding HTML elements by CSS selectors
* Finding HTML elements by HTML object collections

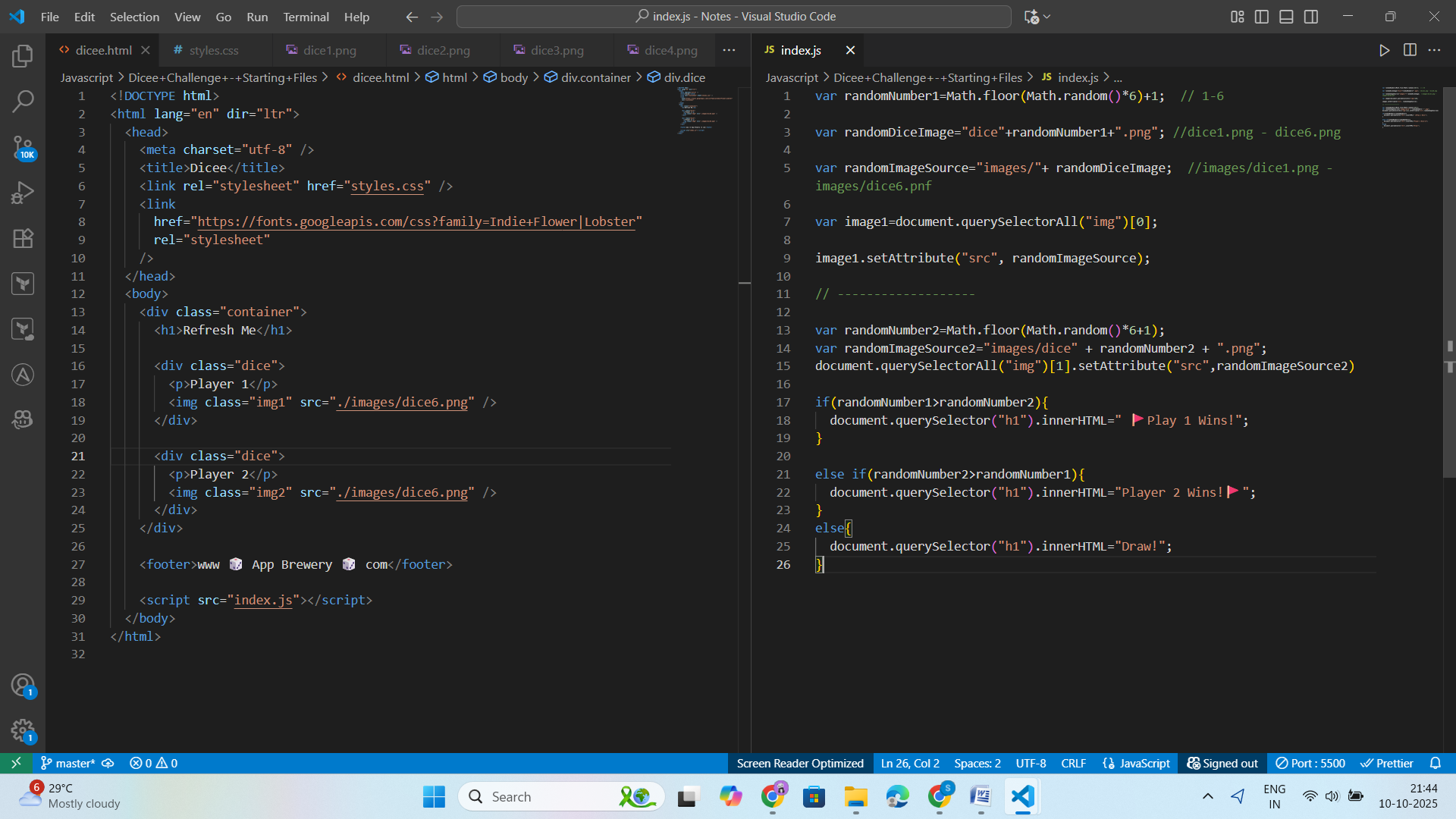
Finding HTML Element by Id

The easiest way to find an HTML element in the DOM, is by using the element id.

This example finds the element with id="intro":



**The Separation of Concerns: Structure vs Style vs Behaviour**





🡪

document.querySelector("button").addEventListener("click",handleClick);

function handleClick(){

  alert("I got clicked!");

}

🡪

var numberOfDrumButtons=document.querySelectorAll(".drum").length;

for  (var i=0; i<numberOfDrumButtons; i++){

document.querySelectorAll(".drum")[i].addEventListener("click",function (){

  alert("I got clicked!");

});

}

**🡪Higher Order Functions and Passing Functions as Arguments**

function add(num1, num2) {

return num1 + num2;

}

function subtract(num1, num2) {

return num1 - num2;

}

function multiply(num1, num2) {

return num1 \* num2;

}

function divide(num1, num2) {

return num1 / num2;

}

function calculator(num1, num2, operator) {

return operator(num1, num2);

}

🡪calculator(4,5,add);

9

**🡪It is used to debug**

debugger;

calculator(3,4,multiply);

🡪palysounds javascript

**var audio = new Audio('audio\_file.mp3');**

**audio.play();**

🡪Background image

**background-image: url("cat-front.png");**

**🡪Switch**

var buttonInnerHTML=this.innerHTML;

    switch (buttonInnerHTML) {

        case "w":

            var tom1 = new Audio("sounds/tom-1.mp3");

            tom1.play();

            break;

        case "a":

            var tom2 = new Audio("sounds/tom-2.mp3");

            tom2.play();

            break;

        case "s":

            var tom3 = new Audio("sounds/tom-3.mp3");

            tom3.play();

            break;

        case "d":

            var tom4 = new Audio("sounds/tom-4.mp3");

            tom4.play();

            break;

        case "j":

            var snare = new Audio("sounds/snare.mp3");

            snare.play();

            break;

        case "k":

            var crash = new Audio("sounds/crash.mp3");

            crash.play();

            break;

        case "l":

            var kick = new Audio("sounds/kick-bass.mp3");

            kick.play();

            break;

        default: console.log(buttonInnerHTML);

    }

**🡪// Constructor Functions**

**function HouseKeeper (yeatsOfExperience,name,cleaningRepertoire){**

**this.yeatsOfExperience=yeatsOfExperience;**

**this.name=name;**

**this.cleaningRepertoire=cleaningRepertoire;**

**}**

**var houseKeeper1=new HouseKeeper(9,"Tom","bedroom");**

**console.log(houseKeeper1);**

**O/P:** **HouseKeeper {yeatsOfExperience: 9, name: 'Tom', cleaningRepertoire: 'bedroom'}**

**🡪Objects, their Methods and the Dot Notation**

**🡪Keyboard keyPress**

document.addEventListener("keypress",function(){

    alert("Key was pressed!");

});

**🡪Callaback Function**

**🡪-------------------**

// Detecting Button press

var numberOfDrumButtons=document.querySelectorAll(".drum").length;

for  (var i=0; i<numberOfDrumButtons; i++){

document.querySelectorAll(".drum")[i].addEventListener("click",function (){

    var buttonInnerHTML=this.innerHTML;

    makeSound(buttonInnerHTML);

    buttonAnimation(buttonInnerHTML);

});

}

// Detecting keyboard Press

    document.addEventListener("keypress",function(event){

    makeSound(event.key);

    buttonAnimation(event.key);

});

    function makeSound(key){

        switch (key) {

            case "w":

                var tom1 = new Audio("sounds/tom-1.mp3");

                tom1.play();

                break;

            case "a":

                var tom2 = new Audio("sounds/tom-2.mp3");

                tom2.play();

                break;

            case "s":

                var tom3 = new Audio("sounds/tom-3.mp3");

                tom3.play();

                break;

            case "d":

                var tom4 = new Audio("sounds/tom-4.mp3");

                tom4.play();

                break;

            case "j":

                var snare = new Audio("sounds/snare.mp3");

                snare.play();

                break;

            case "k":

                var crash = new Audio("sounds/crash.mp3");

                crash.play();

                break;

            case "l":

                var kick = new Audio("sounds/kick-bass.mp3");

                kick.play();

                break;

            default: console.log(buttonInnerHTML);

    }

    }

    // Animation Button press

    function buttonAnimation(currentKey){

        var activeButton=document.querySelector("."+currentKey);

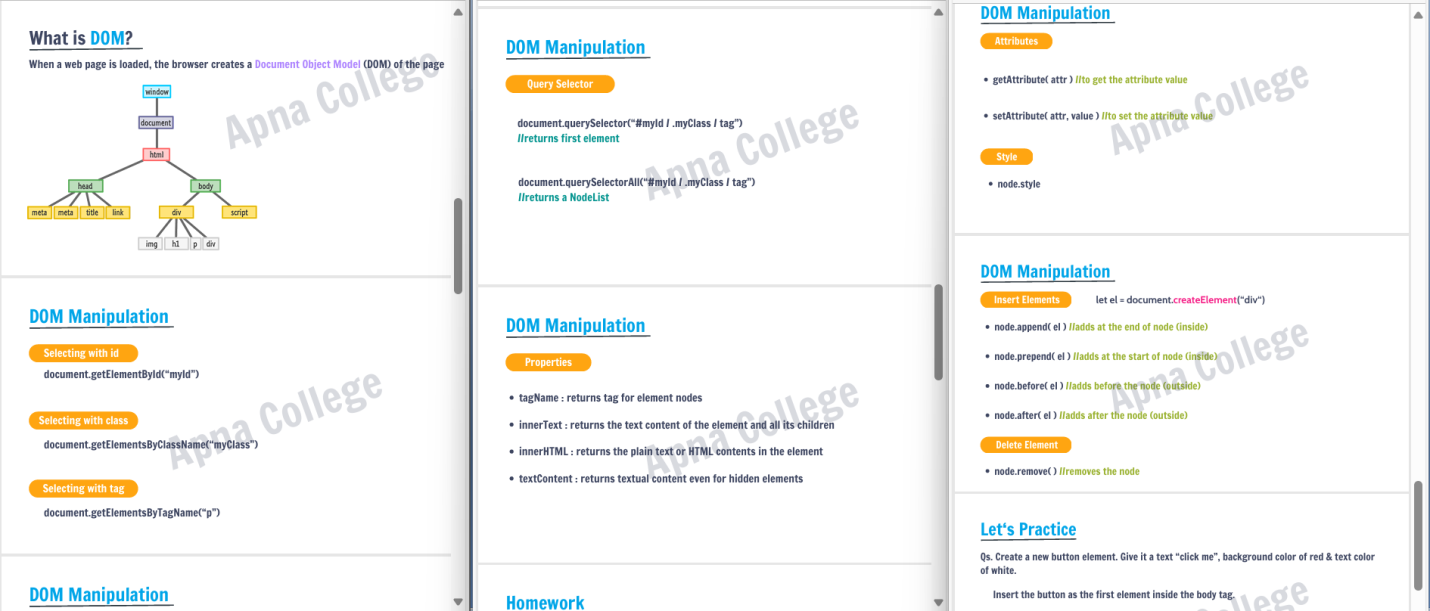
        activeButton.classList.add("pressed");

        setTimeout(function(){

            activeButton.classList.remove("pressed");

    }, 100);

    }

****