

LOCUS 2026

22nd National Technological Festival

**Day 2: JavaScript Essentials
(Frontend Programming)**

Syllabus:



LOCUS 2026
22nd National Technological Festival

- JS Syntax: Data Types, Variables, Functions, Operators
- DOM Manipulation: Selecting Elements, Event Listeners
- Control Structures: If/Else, Loops
- Using Github Copilot for Small JS Tasks
- Lab: Add Interactivity(Example: Color Switcher, Calculator)

JavaScript



LOCUS 2026
22nd National Technological Festival

- It is the programming language of the web.
- High-Level: Abstracts away complex machine details.
- Single-Threaded: Does one thing at a time (with a clever event loop).
- Garbage-Collected: Manages memory for you.
- Multi-Paradigm: Supports different programming styles (procedural, object-oriented).
- Dynamic: Flexible and forgiving (types can change).

Where did JavaScript come from?



LOCUS 2026
22nd National Technological Festival

- 1995: Created by Brendan Eich at Netscape in just 10 days!
- Initially called "Mocha," then "LiveScript," then "JavaScript" to ride the popularity of Java (they are very different!).
- 1997: Standardized as ECMAScript (ES) to ensure compatibility across browsers.
- Now: The undisputed king of client-side scripting and a major player on the server side.

Client Side VS Server Side



LOCUS 2026
22nd National Technological Festival

Client Side

- Runs on the user's computer.
- Makes web pages interactive (animations, form validation, pop-ups).
- Manipulates the DOM (the content you see).
- This is the classic role of JS.

Server Side

- Runs on a remote server.
- Handles business logic, database operations, and user authentication.
- Made possible by environments like Node.js.

Data Types



LOCUS 2026
22nd National Technological Festival

- Data type refers to the kind of value a variable holds
- Different data types in JS:
 1. Number
 2. String
 3. Boolean
 4. Undefined
 5. Null
 6. Object
 7. Symbol
 8. BigInt

Data Types



LOCUS 2026
22nd National Technological Festival

Undefined vs Null

```
● ● ●  
  
let a;  
console.log(typeof(a)); //undefined  
  
let b = null;  
console.log(typeof(b)); //null
```

Data Types



LOCUS 2026
22nd National Technological Festival

Undefined vs Null



```
let b = null;  
console.log(typeof(b)); //object
```

Data Types



LOCUS 2026
22nd National Technological Festival

Objects

- It holds key-value pairs.
- The key is the label for the data while the value is the actual data you want to store.

Data Types



LOCUS 2026
22nd National Technological Festival

Objects

```
● ● ●  
  
let book = {  
    title: "The Great Gatsby",  
    author: "F. Scott Fitzgerald",  
    year: 1925  
};
```

Data Types



LOCUS 2026
22nd National Technological Festival

Objects



```
let book = {  
    title: "The Great Gatsby",  
    author: "F. Scott Fitzgerald",  
    year: 1925  
};
```

```
console.log(book.title); //The Great Gatsby  
console.log(book["year"]); //1925
```

Data Types



LOCUS 2026
22nd National Technological Festival

Dynamic Assignment



```
let value = "Alice";
let key = "Name";
let obj = {[key]:value};

console.log(obj);
//{Name: "Alice"}
```

Data Types



LOCUS 2026
22nd National Technological Festival

Object Destructuring



```
const person = { name: "Alice", age: 30, city: "New York" };

const { name, age } = person;

console.log(name); // Alice
console.log(age); // 30
```

Data Types



LOCUS 2026
22nd National Technological Festival

Object Destructuring



```
let person = { name: "Alice", age: 30, city: "New York" };

let { name: personName, age: personAge } = person;

console.log(personName); // Alice
console.log(personAge); // 30
```

Data Types



LOCUS 2026
22nd National Technological Festival

Symbol

- A special type of value in JS that is always unique and cannot be changed
- Used to create unique labels or identifiers for properties.

Data Types



LOCUS 2026
22nd National Technological Festival

Symbol



```
const symbol1 = Symbol('mySymbol');
const symbol2 = Symbol('mySymbol');
console.log(symbol1 === symbol2); //false
```

Data Types



LOCUS 2026
22nd National Technological Festival

BigInt

- Used for very large numbers that exceed the limit of the Number type

Array



LOCUS 2026
22nd National Technological Festival

- Collection of values of same data type



```
const colors = ["red", "green", "blue"];
```

Variables



LOCUS 2026
22nd National Technological Festival

- It acts as a container for storing a value
- It can be defined with the “var”, “let” or “const” keywords.
- You can reassign values to variables that were defined using the “let” keyword, but you cannot do the same to the variables defined using “const” keyword.
- “var” does not have a block scope. So a variable declared using the “var” keyword can be used outside the block.

Function



LOCUS 2026
22nd National Technological Festival

- They are basically reusable pieces of code that perform a specific task

```
● ● ●  
  
function greet() {  
    console.log("Hello,  
Fellows!");  
  
greet(); //Hello, Fellows!
```

Function



LOCUS 2026
22nd National Technological Festival



```
function add(num1, num2){  
    return num1 + num2;  
}
```

```
console.log(add(3, 4));//7
```

Function



LOCUS 2026
22nd National Technological Festival

- Another way to write functions: **Arrow Function**



```
let add = (num1, num2) => {  
    return num1 + num2;  
}
```

```
console.log(add(3, 4));//7
```

Template Literals



LOCUS 2026
22nd National Technological Festival

- Template literals let you embed variables and expressions inside strings easily.
- Use backticks (` `) instead of single or double quotes.

Template Literals



LOCUS 2026
22nd National Technological Festival



```
const name = 'Alex';
const msg = `Hello, ${name}!`;
// Output: Hello, Alex!
```

Template Literals



LOCUS 2026
22nd National Technological Festival



```
let name = 'Pragalbha';

const intro = `Hi
I am
${name}`;

console.log(intro);
```

Template Literals



LOCUS 2026
22nd National Technological Festival



Arithmetic Operators



LOCUS 2026
22nd National Technological Festival

Operator	Description	Example
<code>+</code>	Addition	<code>a + b</code>
<code>-</code>	Subtraction	<code>a - b</code>
<code>*</code>	Multiplication	<code>a * b</code>
<code>/</code>	Division	<code>a / b</code>
<code>%</code>	Modulus (Remainder)	<code>a % b</code>
<code>**</code>	Exponentiation	<code>a ** b</code>
<code>++</code>	Increment	<code>a++</code>
<code>--</code>	Decrement	<code>a--</code>

Assignment Operators



LOCUS 2026
22nd National Technological Festival

Operator	Description	Example
=	Assign	x = y
+=	Add and assign	x += y
-=	Subtract and assign	x -= y
*=	Multiply and assign	x *= y
/=	Divide and assign	x /= y
%=	Modulus and assign	x %= y
**=	Exponent and assign	x **= y

Relational Operators



LOCUS 2026
22nd National Technological Festival

Operator	Description	Example
<code>==</code>	Equal to	<code>x == y</code>
<code>===</code>	Equal value and type	<code>x === y</code>
<code>!=</code>	Not equal to	<code>x != y</code>
<code>!==</code>	Not equal value or type	<code>x !== y</code>
<code>></code>	Greater than	<code>x > y</code>
<code><</code>	Less than	<code>x < y</code>
<code>>=</code>	Greater than or equal to	<code>x >= y</code>
<code><=</code>	Less than or equal to	<code>x <= y</code>

Strict Equals



LOCUS 2026
22nd National Technological Festival



```
let a = 1;
let b = '1';
console.log(a == b); //true
console.log(a === b); //false
```

Bitwise Operators



LOCUS 2026
22nd National Technological Festival

Operator	Description	Example
&	AND	a & b
	OR	a b
^	XOR	a ^ b
~	NOT	~a
<<	Left shift	a << 2
>>	Right shift	a >> 2
>>>	Zero-fill right shift	a >>> 2

Ternary Operators



LOCUS 2026
22nd National Technological Festival



```
(condition)? true: false;
```

The Console



LOCUS 2026
22nd National Technological Festival

- The console is an essential tool for debugging.
- Open it in your browser with F12 or Ctrl+Shift+I.
- `console.log()`: Print general information.
- `console.error()`: Print an error message.
- `console.table()`: Display objects or arrays in a table.

Control Structures



LOCUS 2026
22nd National Technological Festival

- Control statements manage the flow of a program based on conditions or loops.
- They include conditional (if, switch) and looping (for, while) statements.
- They also include jump statements like break, continue, and return to alter execution.

Control Structures



LOCUS 2026
22nd National Technological Festival

If-Else Ladder

```
● ● ●

let a = 10;
let b = 5;

if(a >= b)
{
    console.log(` ${a} ` + " is greater than " + ` ${b} `);
}
else
{
    console.log(` ${b} ` + " is greater than " + ` ${a} `)
}

//10 is greater than 5
```

Control Structures



LOCUS 2026
22nd National Technological Festival

For Loop



```
for (initialization; condition; increment or decrement) {  
    // code block to be executed  
}
```

Control Structures



LOCUS 2026
22nd National Technological Festival

For Loop



```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

//1 2 3 4 5

Control Structures



LOCUS 2026
22nd National Technological Festival

While Loop



```
while (condition) {  
    // code block to be executed  
}
```

Control Structures



LOCUS 2026
22nd National Technological Festival

While Loop

```
● ● ●  
  
let i = 0;  
while (i < 5) {  
    console.log(i);  
    i++;  
}  
  
//0 1 2 3 4
```

Control Structures



LOCUS 2026
22nd National Technological Festival

Do While Loop



```
do {  
    // code block to be executed  
} while (condition);
```

Control Structures



LOCUS 2026
22nd National Technological Festival

Do While Loop

```
● ● ●  
let i = 5;  
do {  
    console.log(i);  
    i++;  
} while (i < 5);  
  
//5
```

Control Structures



LOCUS 2026
22nd National Technological Festival

- For...In
- For...Of
- Switch Case
- Break
- Continue
- Return

Scope



LOCUS 2026
22nd National Technological Festival

- Scope determines where variables can be accessed.
- Types of scope: Global, Function, Block



Global Scope

- Variables declared outside any function are global.

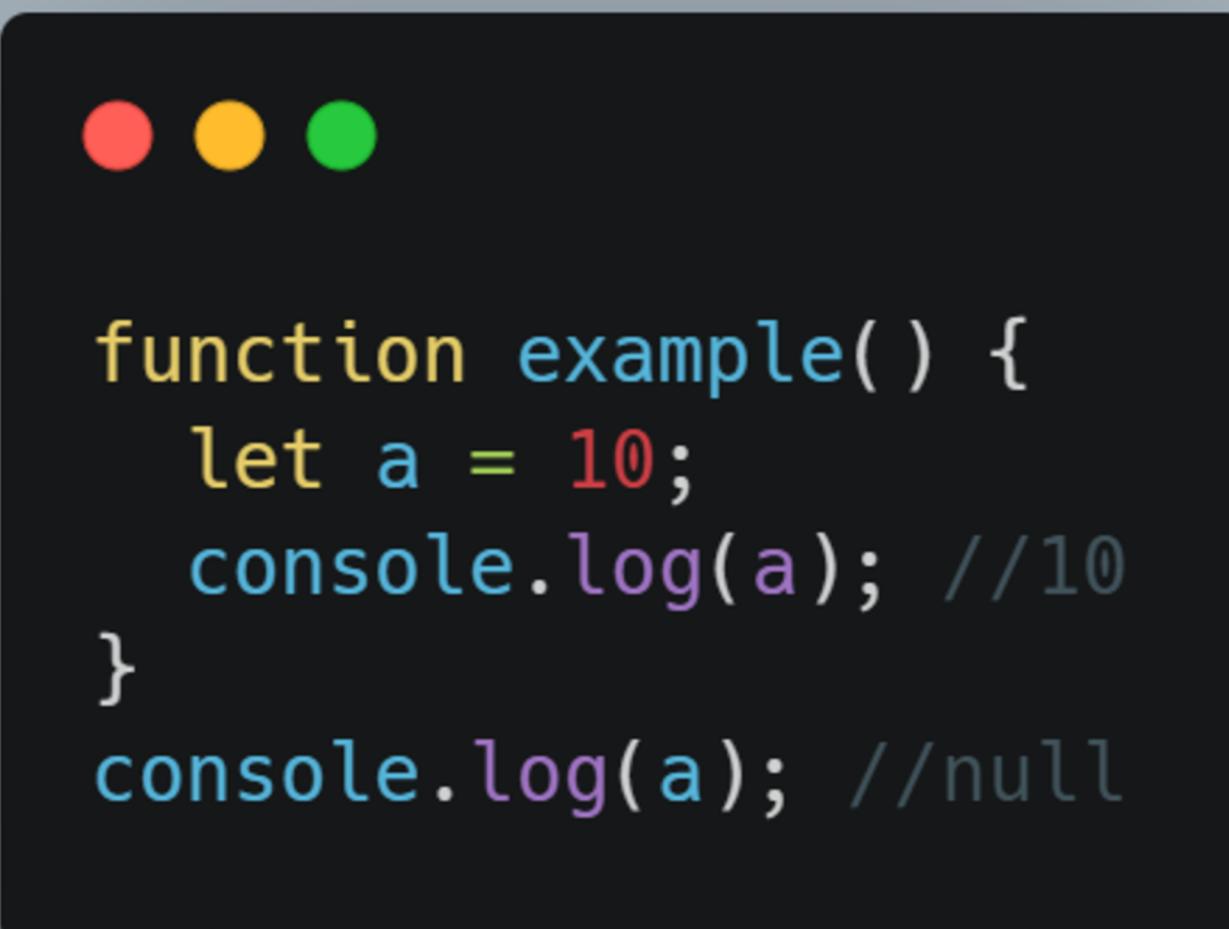


```
let x = 10;  
function show( ) {  
    console.log(x); //10  
}
```



Function Scope

- Any variable declared with var, let, or const inside a function is only accessible inside that function.



A dark gray rectangular box with rounded corners, resembling a terminal window, containing three colored dots (red, yellow, green) at the top left. Below them is a block of JavaScript code.

```
function example() {
  let a = 10;
  console.log(a); //10
}
console.log(a); //null
```



Block Scope

- Variables declared with let or const inside {} blocks (like if, for, while) are only accessible in that block.

```
● ● ●

if (true) {
  const b = 20;
  console.log(b); //20
}
console.log(b); //null
```

Mini Exercise



LOCUS 2026
22nd National Technological Festival

Write a function that takes one argument (a number) and prints to the console whether that number is "even" or "odd".

Document Object Model



LOCUS 2026
22nd National Technological Festival

WEBSITE STRUCTURE



HTML
The Skeleton



CSS
The Skin



Java Script
The Brains

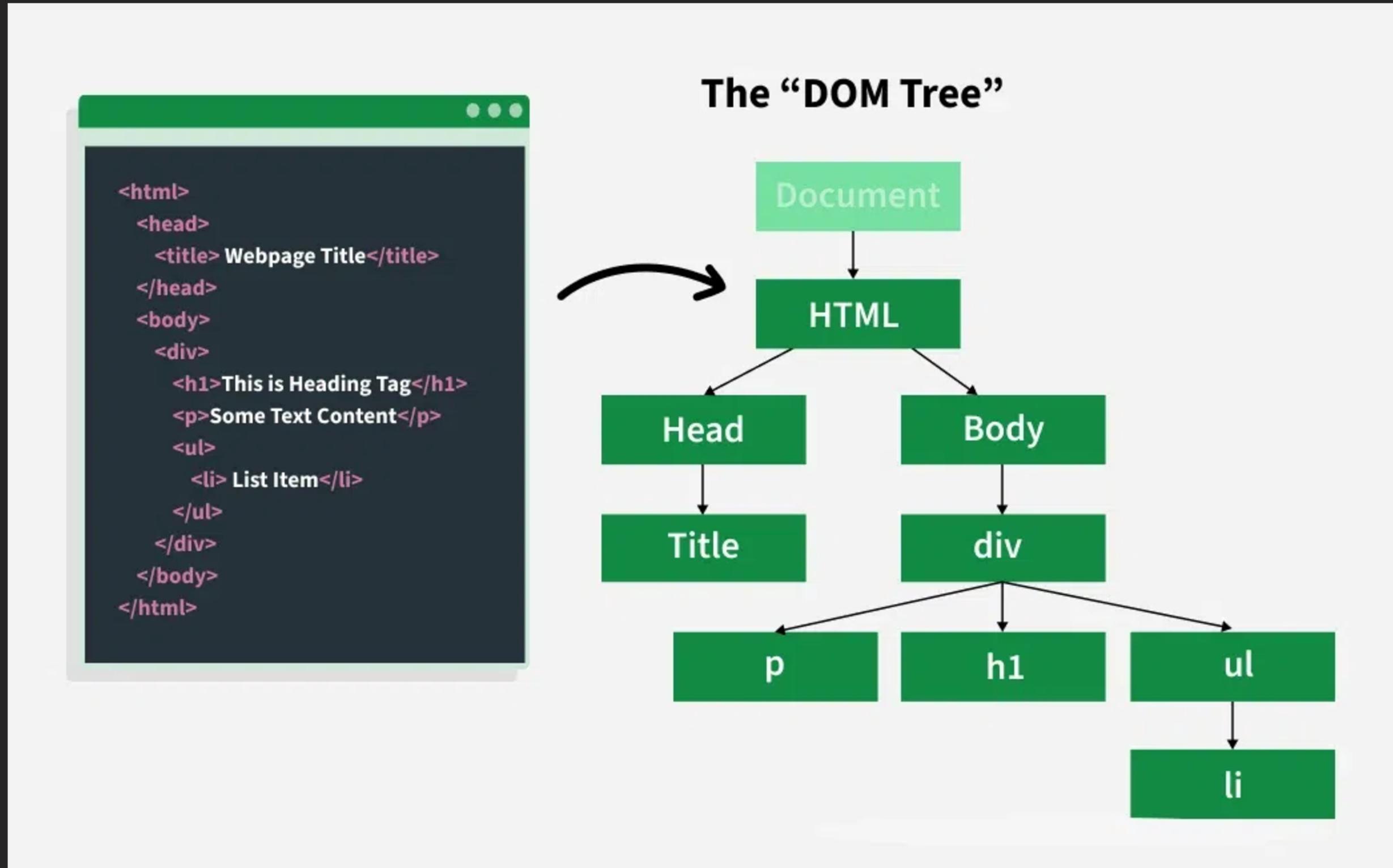
Document Object Model



LOCUS 2026
22nd National Technological Festival

It's a programming interface that lets us interact with HTML documents.

- Root Element
- Container Element
- Child Element





DOM Manipulation:

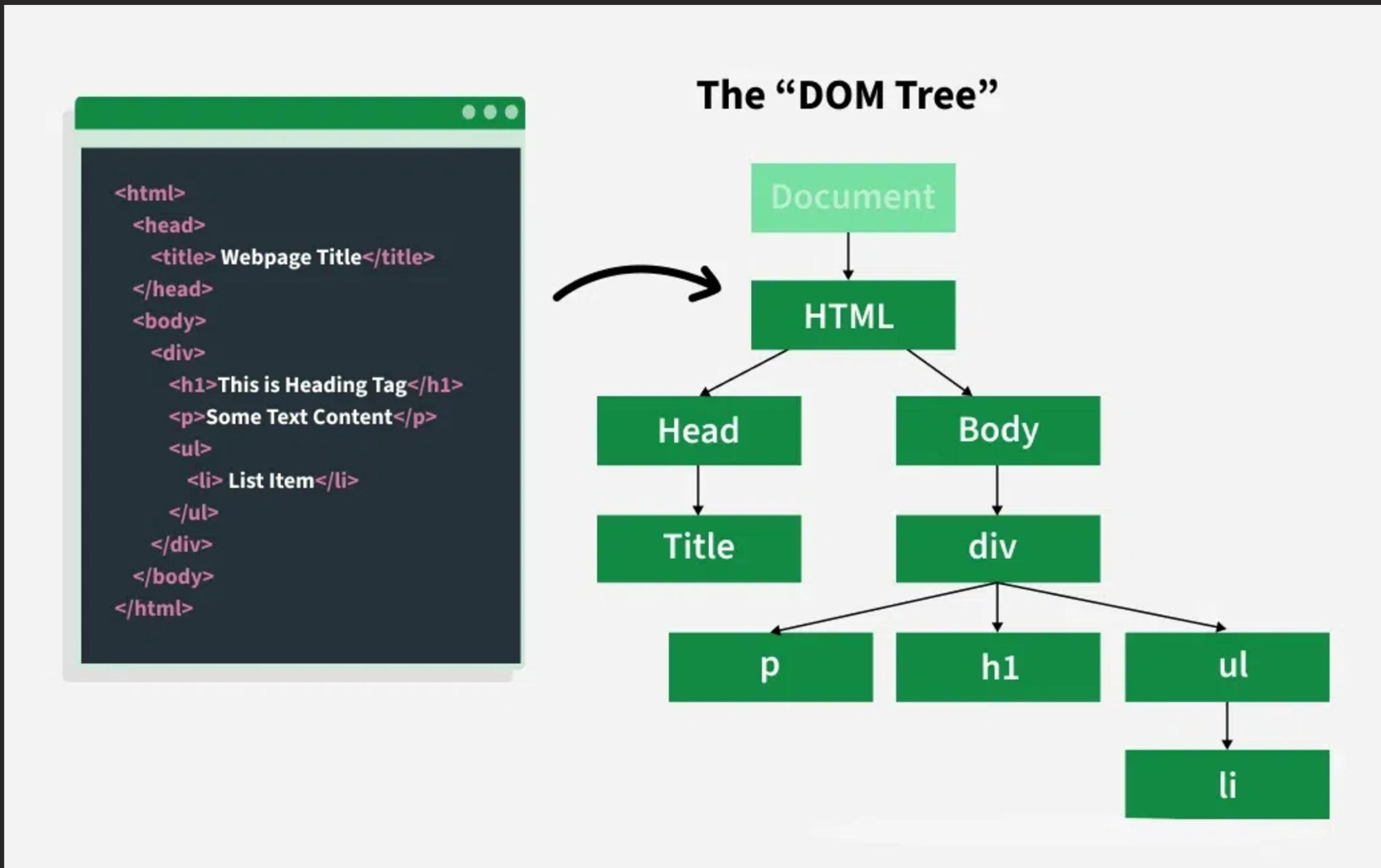
DOM manipulation means:

- using JavaScript to **find**, **change**, **create**, or **remove** elements on the web page after it has loaded to make it more interactive.

We usually perform DOM manipulation by selecting, modifying or adding new elements or even deleting element.



DOM Manipulation:



Selecting Elements



LOCUS 2026
22nd National Technological Festival

- `getElementById('id')`: selects element by ID.
- `querySelector('selector')`: selects first matching element.
- `getElementsByClassName('class')`: selects all elements with that class.
- `querySelectorAll('selector')`: selects all matching elements.

Modifying Elements



LOCUS 2026
22nd National Technological Festival

Once you've selected an element, you can change it by:

- `element.innerText` → Changes visible text.
- `element.textContent` → Includes hidden text too.
- `element.innerHTML` → Replaces HTML inside.
- `element.style.property` → Changes inline CSS.
- `element.setAttribute("name", "value")` → Updates HTML attribute.

Adding & Removing Elements



LOCUS 2026
22nd National Technological Festival

- **Create a new element**

```
document.createElement('p')
```

- **Add content**

```
element.innerText = 'Hello World'
```

- **Append to parent element**

```
parent.appendChild(newElement)
```

- **Remove an element**

```
element.remove()
```

Event Listeners:



LOCUS 2026
22nd National Technological Festival

- An event listener is a function that runs in response to an event.
- Events can be triggered by:

User actions (click, scroll, type, hover)

Browser actions (page load, resize, etc.)

- Basic syntax of it is :

```
element.addEventListener('click', function(e){  
    // do something here  
});
```

Above, the **click** is the name of event and it can be other like mouseover

Event Listeners + DOM :



LOCUS 2026
22nd National Technological Festival

- Let's do small exercise that combines our knowledge of DOM manipulation and event listener.
- **Goal : Create a button on an HTML page. When the button is clicked, it changes the text and color of a heading.**

Hints:

- select the button and the heading.
- Add a click event listener to the button
- Inside the listener, change the innerText and style.color of the heading.

Using GitHub Copilot for Small JS Tasks



LOCUS 2026
22nd National Technological Festival

- An AI pair programmer by GitHub & OpenAI.
- Helps you write code faster by suggesting lines or entire functions.
- Works inside VS Code and other supported editors

Why Use Github Copilot for Small JS Tasks?



LOCUS 2026
22nd National Technological Festival

- Speeds up simple coding (e.g., loops, DOM manipulation).
- Great for learning patterns and syntax.
- Offers autocomplete-style suggestions as you type.

Examples of Small JS Tasks



LOCUS 2026
22nd National Technological Festival

- Creating a function to reverse a string
- Adding a click event to a button
- Generating a random number
- Fetching data from an API
- Toggling a CSS class

Using Github Copilot



LOCUS 2026
22nd National Technological Festival

- Start typing a comment or function name
- Copilot shows suggestions (grayed text)
- Press Tab to accept, or keep typing for new suggestions
- Use clear comments (// Create a function to...)
- Don't blindly accept suggestions: read & understand first
- Great for learning best practices, not a replacement for learning

Node.js(short-intro)



LOCUS 2026
22nd National Technological Festival

- Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.
- It's a program that lets you run JavaScript outside of a web browser.
- This allows JS to be used for:
 - Building fast and scalable backend servers.
 - Creating command-line tools.
 - Automating development tasks.



LOCUS 2026
22nd National Technological Festival

Any Questions?