Skyline
Advanced Technology Services

JavaScript Essentials

Syntax, Control Structures, Data Structures, and Functions

Brandon Stryker, Instructor

Brandon Stryker

Presenter
Skyline ATS

Bill Heller

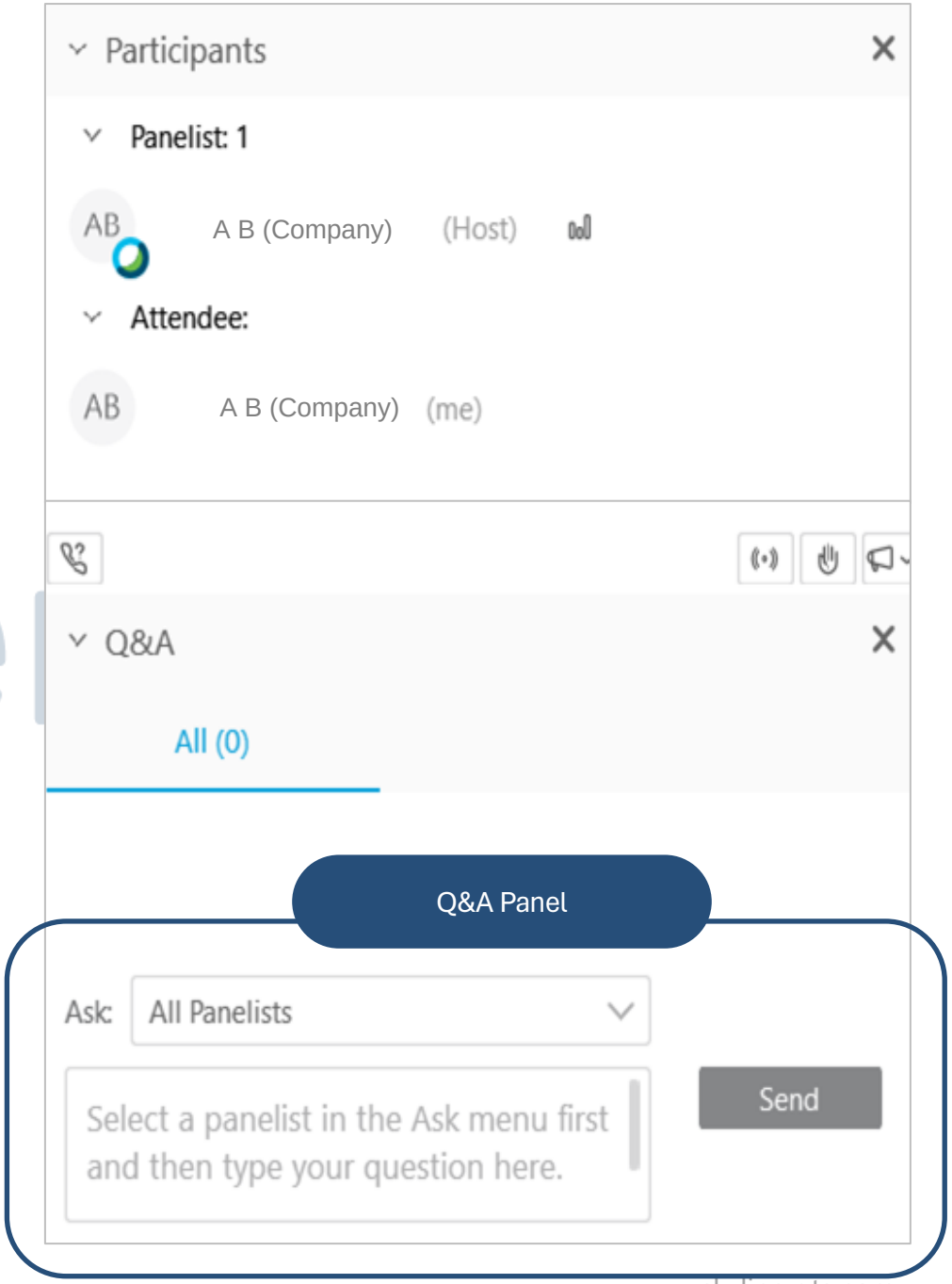
Moderator
Skyline ATS

Tech Session Logistics

Ask our experts a question:

Type your question into the Q&A panel and click **Send**.

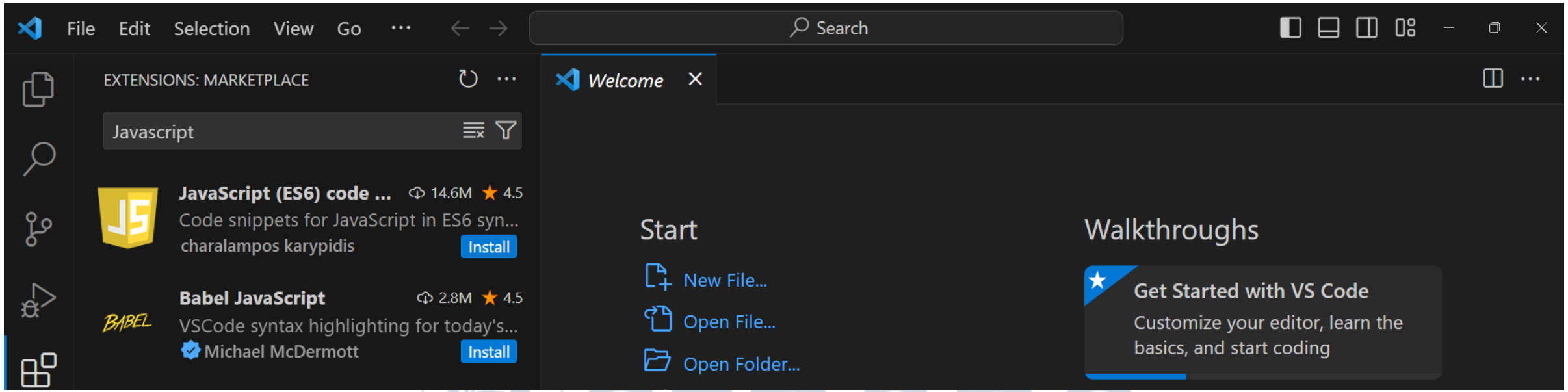
There will also be a brief Q&A session when we wrap up.



If you are having audio issues, please use the chat window and one of our panelists will assist you.

Our Agenda

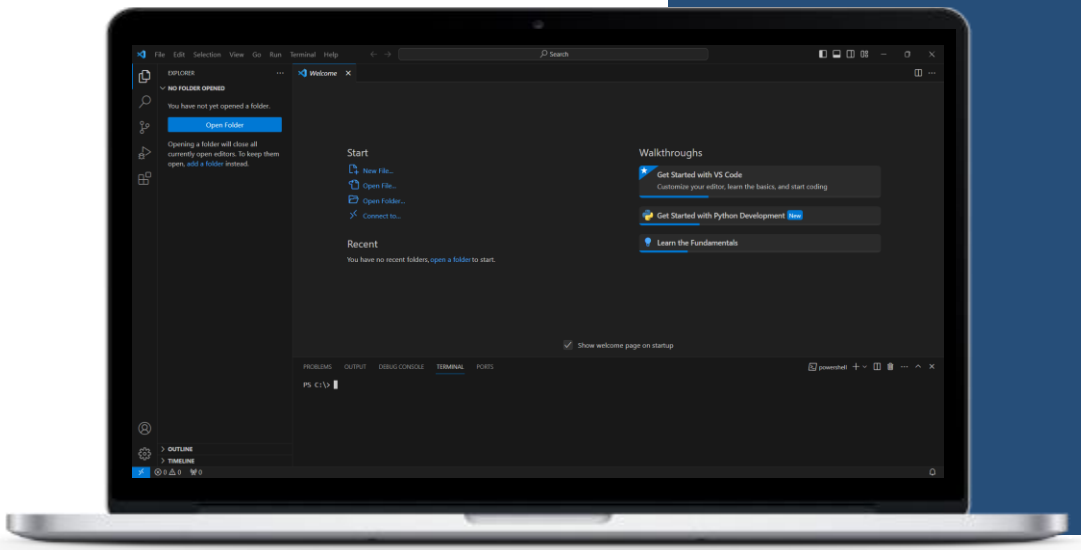
- **Getting Started**
  IDEs, Output, and Running a Script
- **Variables**
  Name and Store Data
- **Data Types**
  Booleans, Integers, Strings, Arrays, and Objects
- **Control Structures**
  Conditionals, Statements and Loops
- **Functions**
  Named, Reusable Blocks of Code
- Next Steps to Continue Learning



Getting Started

Using and Integrated Development Environment (IDE), running code, and generating/observing output.

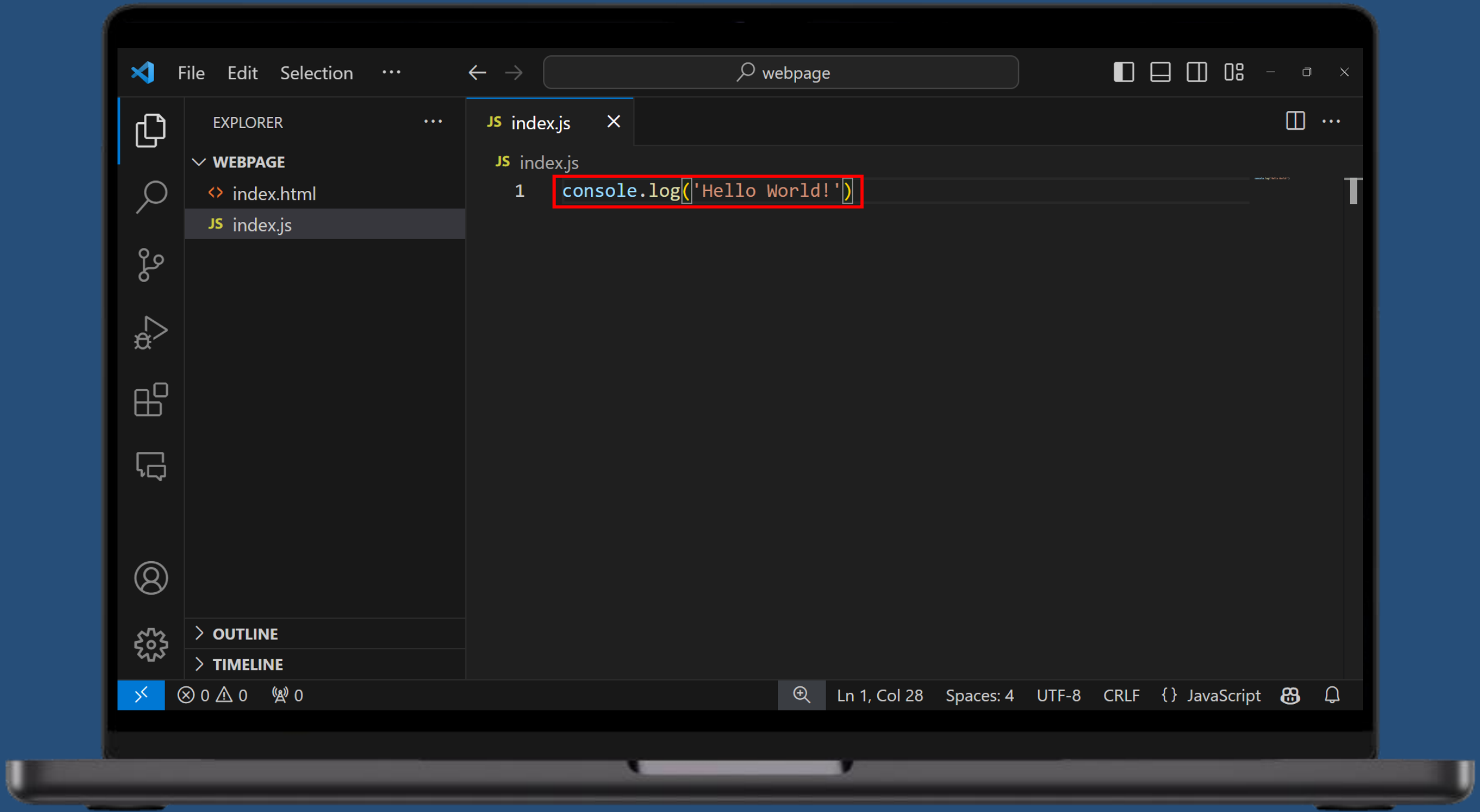
Integrated Development Environment (IDE)



And IDE is a text editor with advanced tooling enabling you to write, run, and debug code all in one program.

IDEs Facilitate Coding

With access to a huge collection of plug-ins and integrations, IDEs, like Visual Studio Code, allow you to customize your development environment. This makes coding easier and greatly improves the entire development process.


File
Edit
Selection
webpage
EXPLORER
WEBPAGE
index.html
index.js
index.js
index.js
1
console.log('Hello World!')
OUTLINE
TIMELINE
Ln 1, Col 28
Spaces: 4
UTF-8
CRLF
JavaScript

Run JS Code in a Web Browser

index.html > ...

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>My First Web Page</title>
5      <script src="./index.js"></script>
6    </head>
7    <body>
8      <h1>Welcome to my first web page!</h1>
9      <p>This is a paragraph of text.</p>
10   </body>
11 </html>
```

If you are building a webpage, you can use the <script> tag to run JS code, either from local files or a remote endpoint.

JS Was Built for Websites

Historically, JS was used primarily for manipulating elements and making requests in a website.

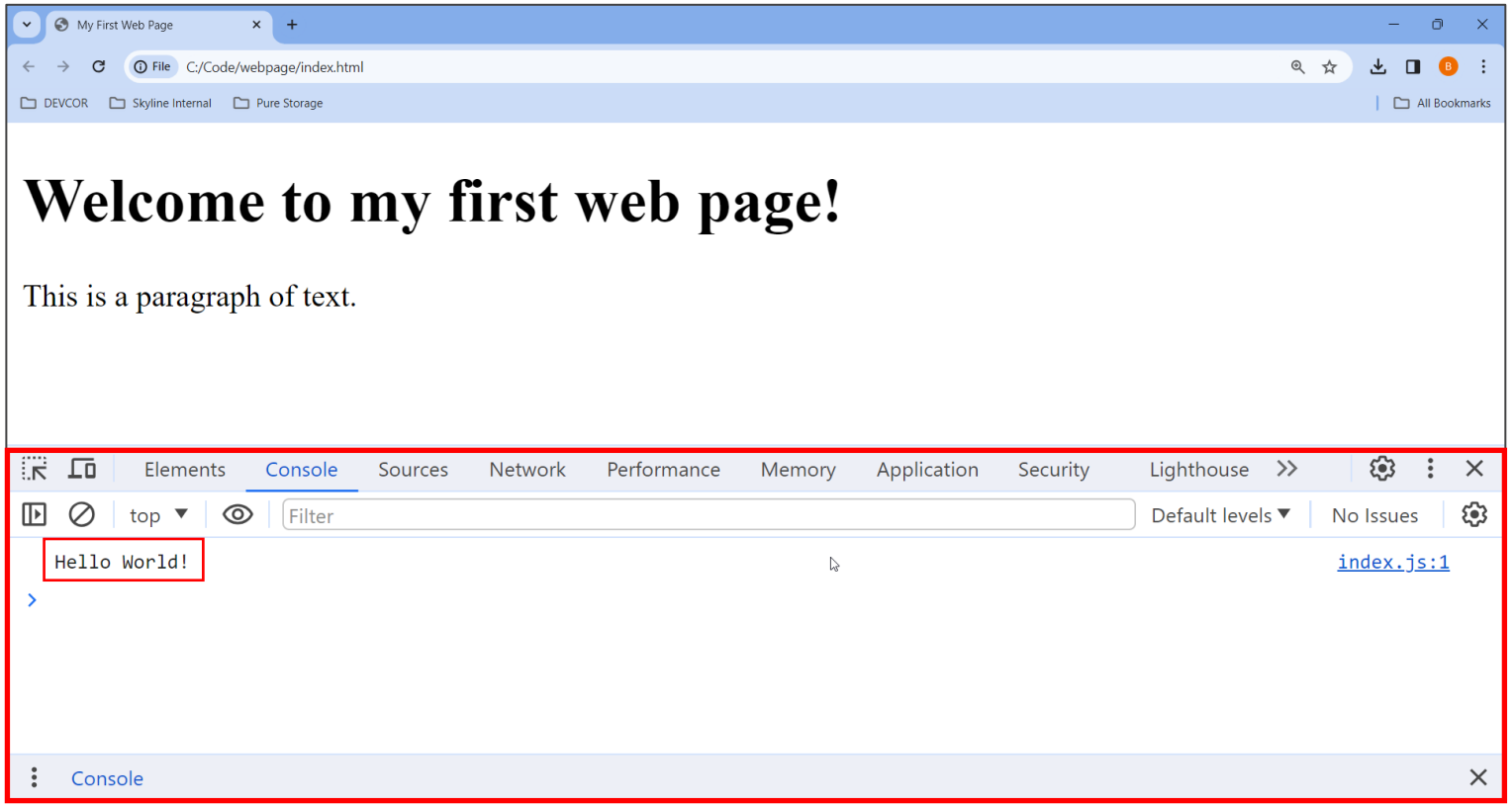
While its initial design catered to browser environments, JavaScript's functionality has vastly extended beyond web page scripts.

The browser serves as the execution environment for JavaScript, interpreting and running scripts that provide interactivity and functionality to web pages.

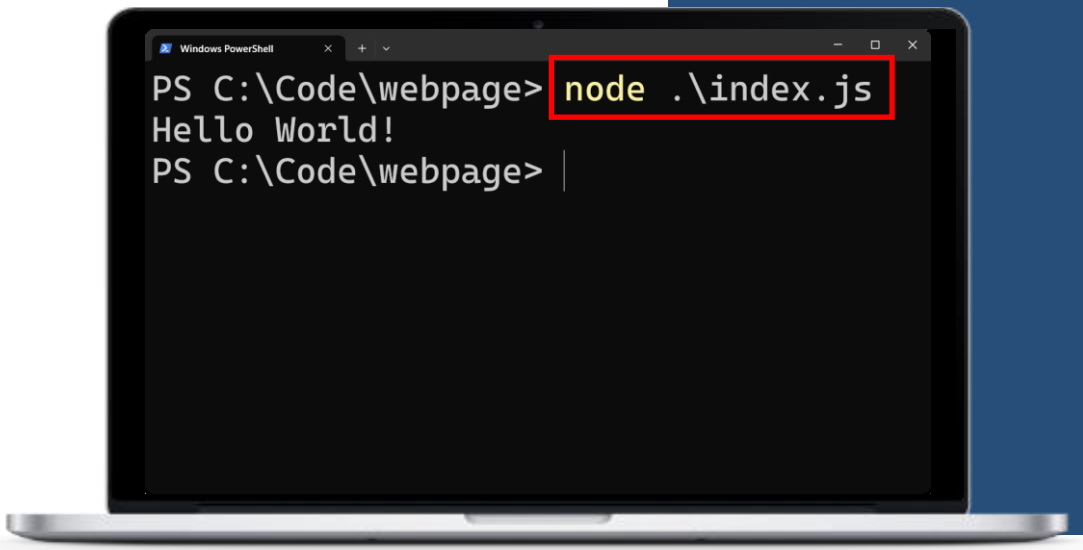
The Browser

Modern browsers have a suite of monitoring and debugging tools that aid with the development process.

You can view output, inspect elements, run inline JS, and much more.



The Node.js Framework



Node.js is a runtime environment that allows JavaScript to be executed on the server side, enabling the development of scalable network applications outside of the browser context.

Command Line

Developers can execute JavaScript files, manage packages, and interact with Node.js through CLI commands.

```
const MAX_CONNECTIONS = 5;
let currentConnections = 1;
var isServerRunning = true;
```

Variables

Storing data in named memory space which can be used or manipulated.

Coding Without Variables is Challenging

```
console.log(20 * 10); // Outputs: 200
```

In this line of code, how can you identify that we are calculating the area of a rectangle? How can you keep track of which number is length or width?

Using Variables Clarifies Code

In this code block, we have labeled which value is the length and width. We have also named the value that is the result of the calculation.

Clear and accurate naming of variables makes you code more readable, easier to understand, and easier to maintain.

```
let length = 20;
let width = 10;
let area = length * width;
console.log(area); // Outputs: 200
```

Variable Declaration

The keywords **let**, **const**, and **var** enable the creation of variables that hold information, such as numbers, text, or more complex data. Each keyword comes with its own rules and intended usage.

```
let count = 1;
count += 1; // Now count is 2
```

let is used to create a variable that might change later, like the score in a game or the number of likes on a photo.

```
const MAX_USERS = 100;
// Changing value will throw Error
```

const is used to declare variables whose values do not change, like a birthday. Once it's set, it stays the same.

Best Practice

```
var userName = "Alice";
userName = "Bob"; // var allows reassignment
```

var also creates a variable, but it's an older way of doing it. Think of it like putting a note in a big room; it's there, but not as neatly organized as with **let** or **const**.

Antiquated

Data Types

Storing data in named memory space which can be used or manipulated.

```javascript
// Boolean
let isActive = true;

// Integer
let age = 30;

// String
let name = "Alice";

// Array
let colors = ["red", "green", "blue"];

// Object
let person = {
    firstName: "Alice",
    lastName: "Smith",
    age: 30
};
```

Data Types

Booleans, Numbers, Strings, Arrays, and Objects

Data types dictate how data can be used and what operations are available for them, forming the foundation of JavaScript programming.

Boolean

A Boolean represents a logical entity and can have two values: true or false.

Booleans are commonly used in conditional statements to control program flow.

```
let isAvailable = true;
console.log(isAvailable); // Output: true

// Toggling boolean value
isAvailable = !isAvailable;
console.log(isAvailable); // Output: false
```

Number

JavaScript uses the Number data type for both integer and floating-point numbers.

Numbers are essential for calculations, measurements, and any numerical operations.

```
// Define an integer
let items = 5;

// Define a float
let pricePerItem = 2.99;

// Perform an arithmetic operation
let totalCost = items * pricePerItem;

console.log(totalCost); // Outputs: 14.95
```

String

A String is a sequence of characters used to represent text.

Strings are used for storing and manipulating text such as names, messages, or any textual data.

```
let greeting = "Hello, World!";
console.log(greeting); // Output: Hello, World!

// Concatenation
greeting += " How are you?";
console.log(greeting); // Output: Hello, World! How are you?

// Getting a substring
console.log(greeting.substring(0, 5)); // Output: Hello
```

Array

An Array is an ordered collection of items, which can be of any data type.

Arrays are used for storing lists of data, like a collection of names or a series of numbers.

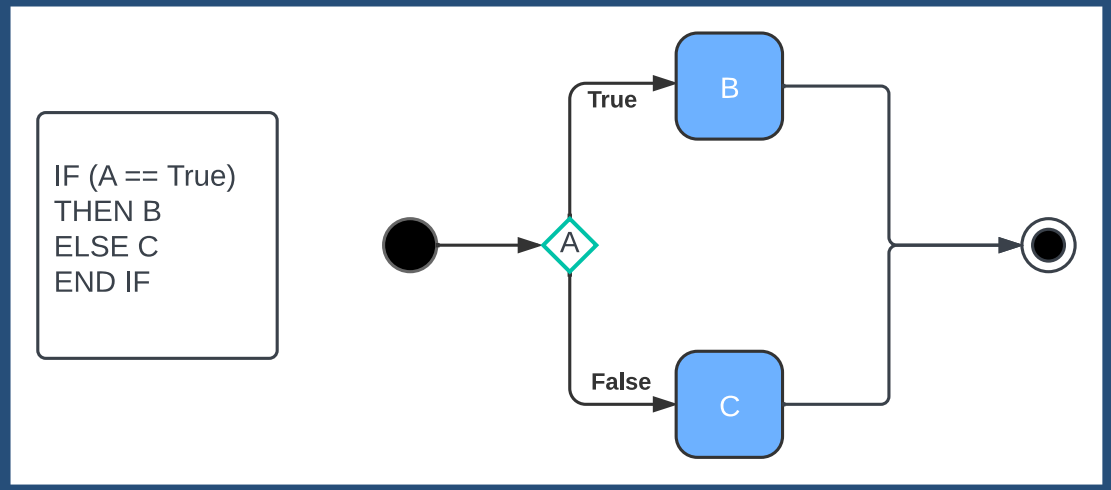
```
let colors = ["red", "green", "blue"];
console.log(colors); // Output: ['red', 'green', 'blue']

// Adding an item
colors.push("yellow");
console.log(colors); // Output: ['red', 'green', 'blue', 'yellow']

// Removing the last item
colors.pop();
console.log(colors); // Output: ['red', 'green', 'blue']

// Finding the index of an item
console.log(colors.indexOf("green")); // Output: 1
```

Object

An Object is a collection of related data and/or functionality consisting of key-value pairs.

Objects can represent more complex data structures like a person with properties (name, age).

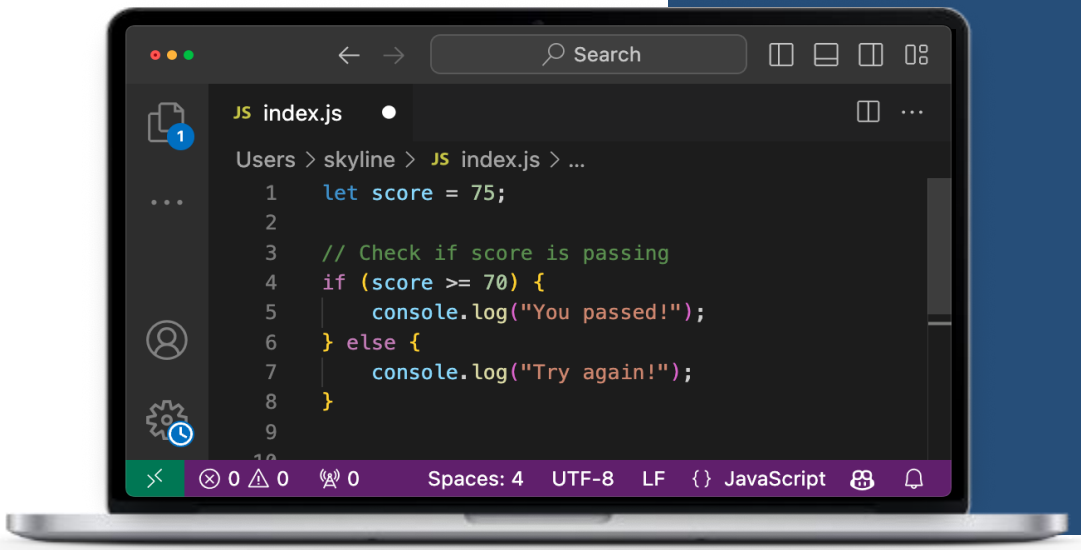
```
let person = {
    name: "Alice",
    age: 30
};
console.log(person); // Output: { name: 'Alice', age: 30 }

// Accessing properties
console.log(person.name); // Output: Alice

// Adding a new property
person.email = "alice@example.com";
console.log(person); // Output: { name: 'Alice', age: 30, email: 'alice@example.com' }

// Deleting a property
delete person.age;
console.log(person); // Output: { name: 'Alice', email: 'alice@example.com' }
```



Conditional Statements

Determine what code should run based on statements evaluating to true or false

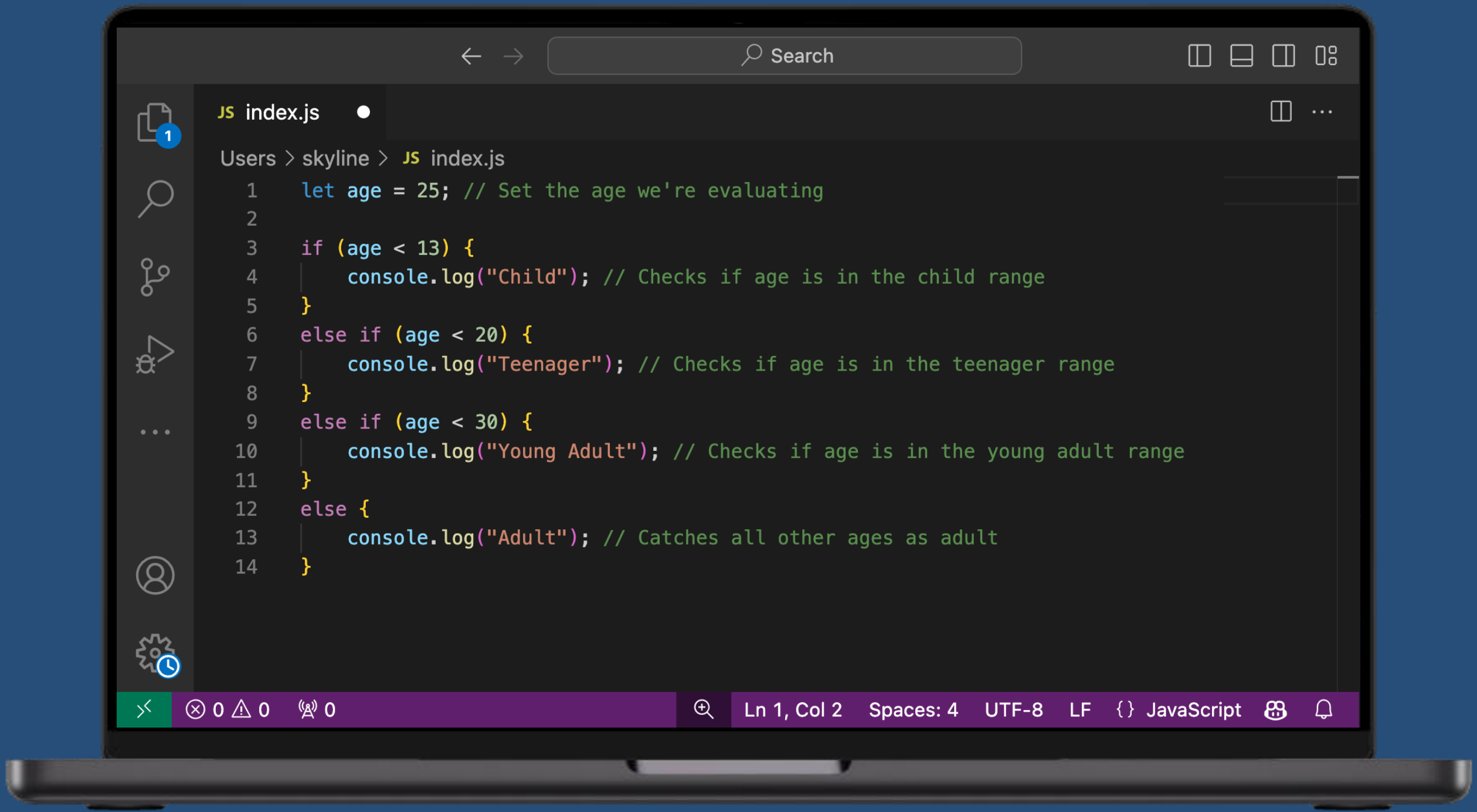
Evaluate Conditions With if-else

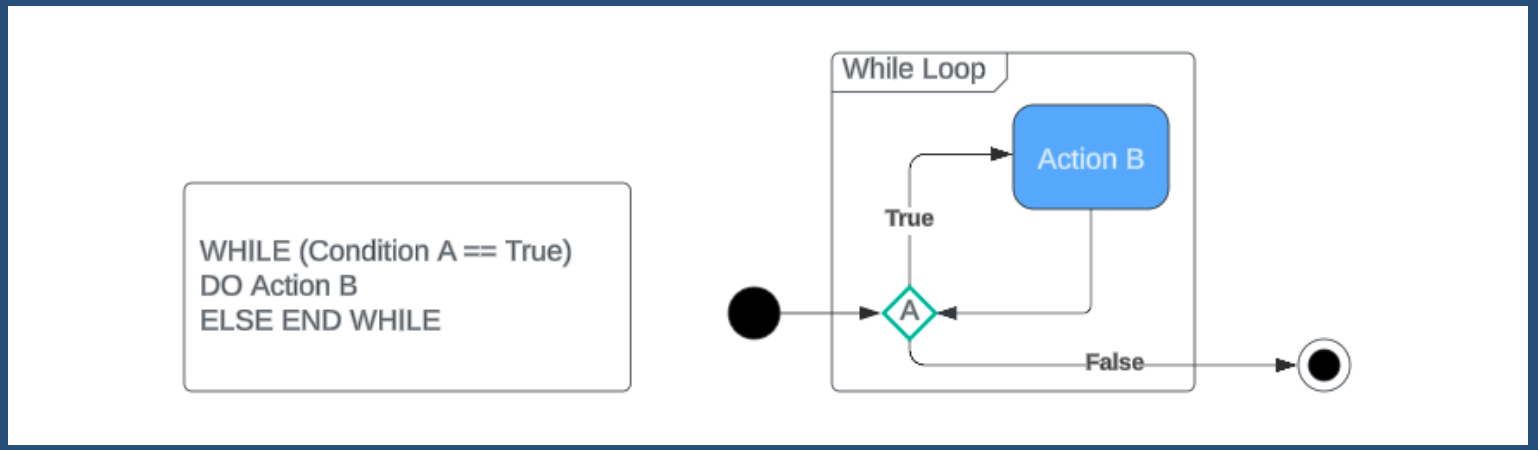


Conditional (if-else) statements allow your code to make decisions by executing different code blocks based on certain conditions.

In the code above:

- The if statement checks whether the score is 70 or higher.
- If true, "You passed!" is logged to the console.
- If false, the code in the else block runs, logging "Try again!".

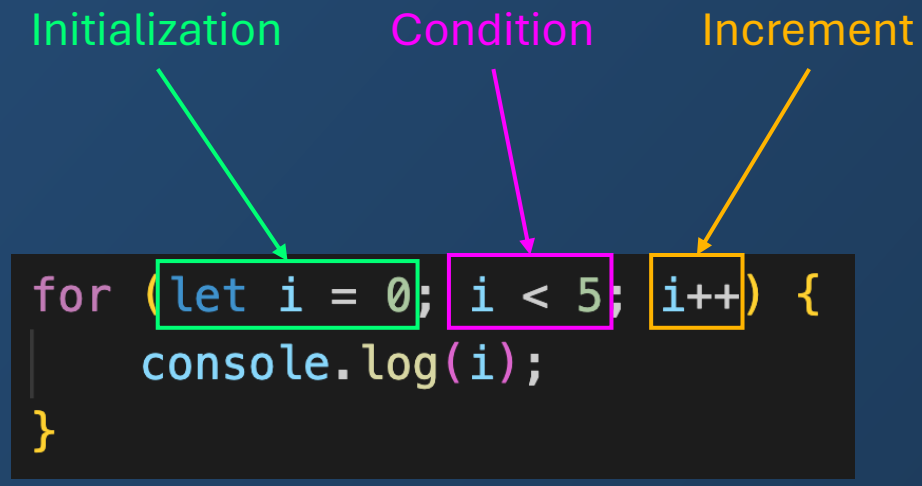

Search
index.js
Users > skyline > index.js
1 let age = 25; // Set the age we're evaluating
2
3 if (age < 13) {
4 console.log("Child"); // Checks if age is in the child range
5 }
6 else if (age < 20) {
7 console.log("Teenager"); // Checks if age is in the teenager range
8 }
9 else if (age < 30) {
10 console.log("Young Adult"); // Checks if age is in the young adult range
11 }
12 else {
13 console.log("Adult"); // Catches all other ages as adult
14 }
0 0 0
Ln 1, Col 2 Spaces: 4 UTF-8 LF JavaScript



Loops

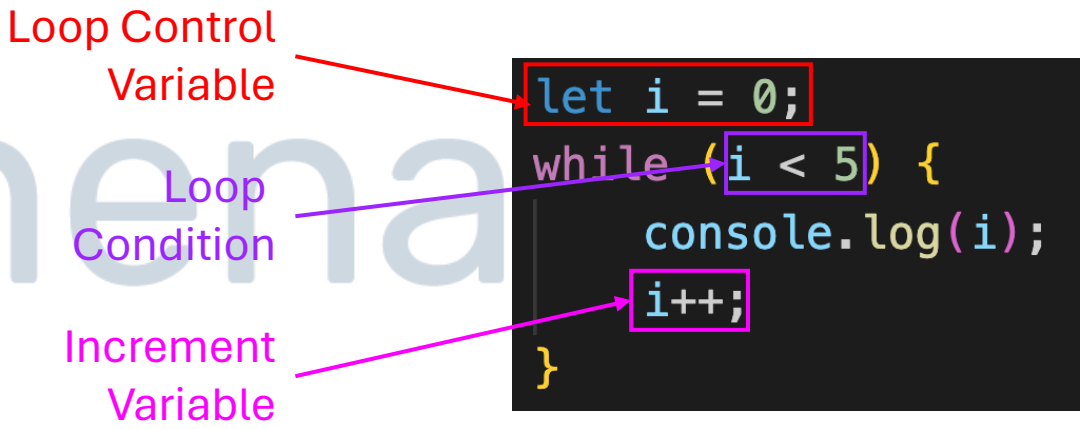
Repeat a block of code multiple times until a specified condition is met or no longer true.

For Loops



The for loop runs a block of code a specific number of times. It's useful for when you know in advance how many times you want to execute the loop.

While Loops



The while loop runs as long as a specified condition is true (i < 5). It's useful when the number of iterations is not known before the loop starts.

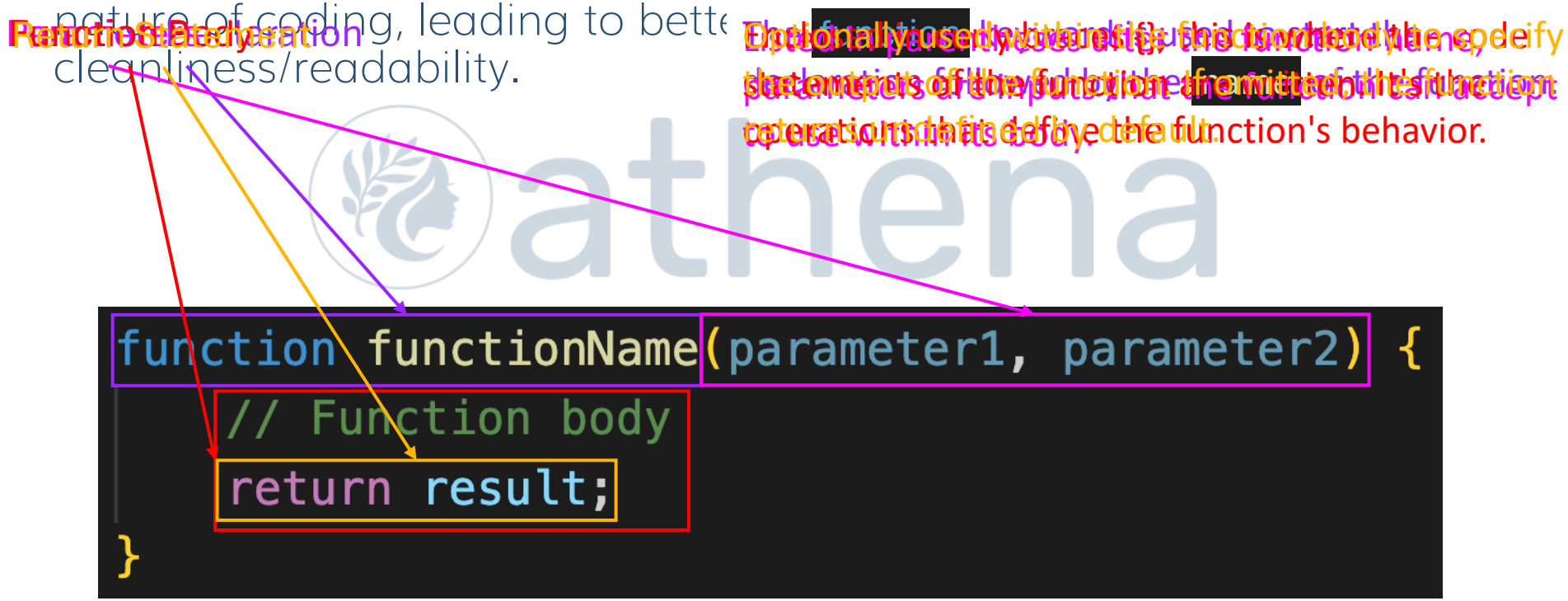
```
const sum = (x, y) => {
    return x + y;
}

sum(2, 3) // 5
```

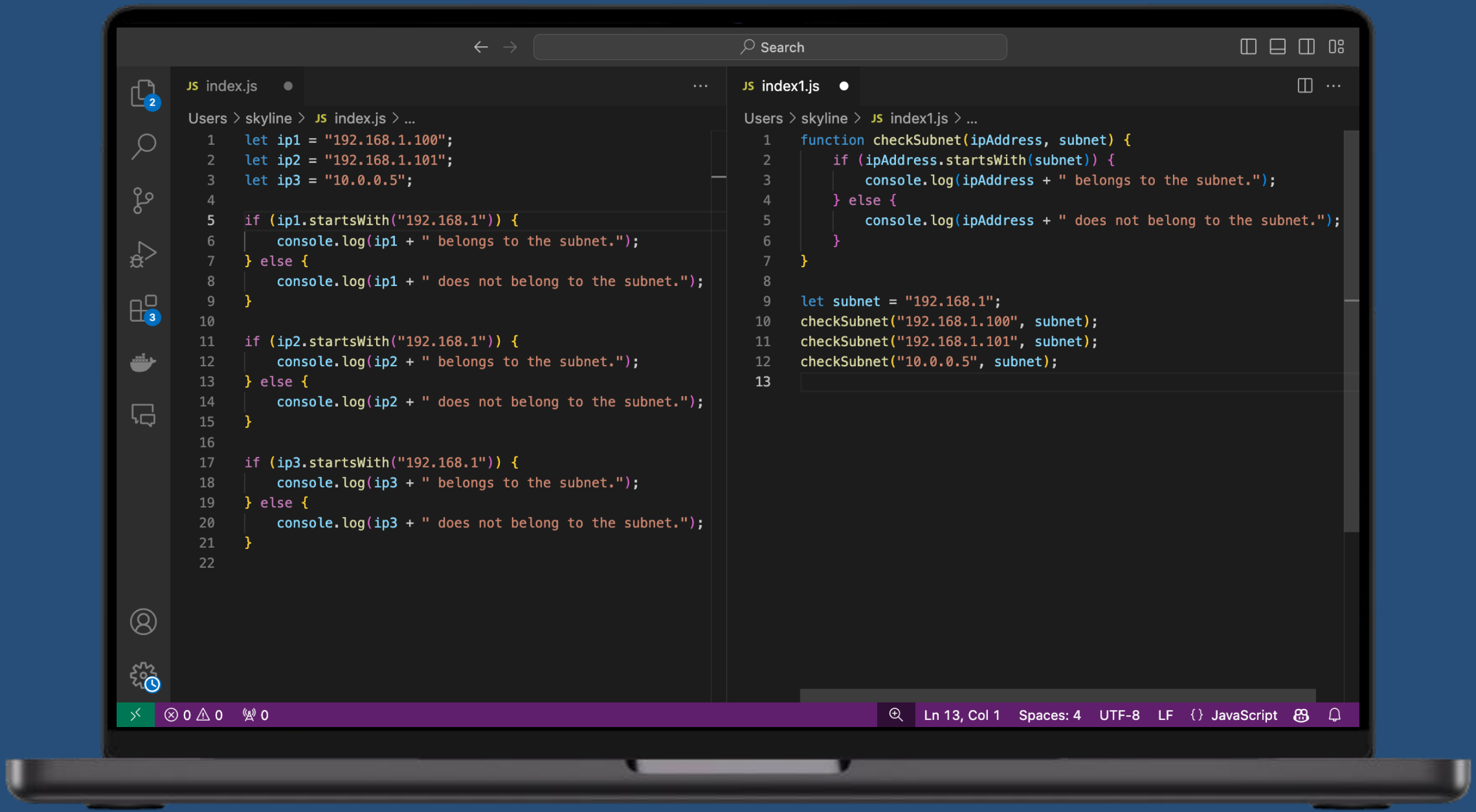
Functions

Name blocks of code that you can reuse throughout your script or codebase.

Elements of a Function

We can use functions and their arguments to reduce the repetitive nature of coding, leading to better code organization and cleanliness/readability.




Search
index.js
Users > skyline > index.js > ...
let ip1 = "192.168.1.100";
let ip2 = "192.168.1.101";
let ip3 = "10.0.0.5";
if (ip1.startsWith("192.168.1")) {
console.log(ip1 + " belongs to the subnet.");
} else {
console.log(ip1 + " does not belong to the subnet.");
}
if (ip2.startsWith("192.168.1")) {
console.log(ip2 + " belongs to the subnet.");
} else {
console.log(ip2 + " does not belong to the subnet.");
}
if (ip3.startsWith("192.168.1")) {
console.log(ip3 + " belongs to the subnet.");
} else {
console.log(ip3 + " does not belong to the subnet.");
}
index1.js
Users > skyline > index1.js > ...
function checkSubnet(ipAddress, subnet) {
if (ipAddress.startsWith(subnet)) {
console.log(ipAddress + " belongs to the subnet.");
} else {
console.log(ipAddress + " does not belong to the subnet.");
}
}
let subnet = "192.168.1";
checkSubnet("192.168.1.100", subnet);
checkSubnet("192.168.1.101", subnet);
checkSubnet("10.0.0.5", subnet);
0 0 0
Ln 13, Col 1
Spaces: 4
UTF-8
LF
JavaScript

Questions

Thank you for attending.

If you have any additional questions, or would like to learn more about our Athena program, please email...

pka@skyline-ats.com