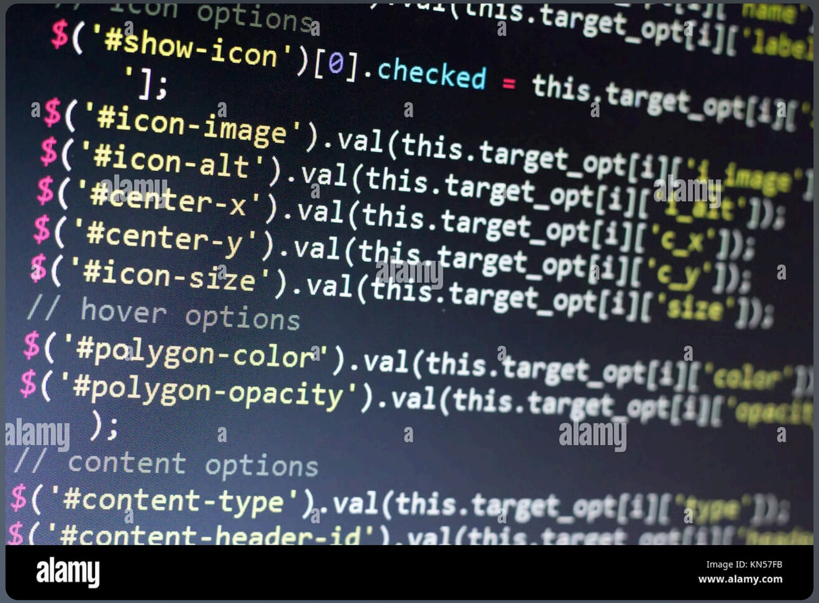
Introduction to JavaScript

JavaScript is a powerful programming language that can be used to build interactive websites and web applications. It is a popular language among developers and is used to create everything from simple animations to complex web games.

K **by Kawaljeet Kaur**



```
// icon options                 .val(this.target_opt[i][ 'name'
$('#show-icon')[0].checked = this.target_opt[i][ 'label
    '];
$('#icon-image').val(this.target_opt[i][ 'i_image']
$('#icon-alt').val(this.target_opt[i][ 'i_alt']);
$('#center-x').val(this.target_opt[i]['c_x']);
$('#center-y').val(this.target_opt[i]['c_y']);
$('#icon-size').val(this.target_opt[i]['size']);
// hover options
$('#polygon-color').val(this.target_opt[i]['color']
$('#polygon-opacity').val(this.target_opt[i]['opacit
    );
// content options
$('#content-type').val(this.target_opt[i]['type']);
$('#content-header-id').val(this.target_opt[i]['header
```

alamy
Image ID: KN57FB
www.alamy.com

Made with Gamma

Syntax and Variables

JavaScript syntax is similar to other programming languages, but with some unique features. Variables are used to store data in JavaScript and they can be declared with the `var`, `let`, or `const` keywords.

1 Variable Types

Variables can be of various data types, including numbers, strings, booleans, and more. Understanding how to use different data types is crucial for effective coding.

2 Keywords

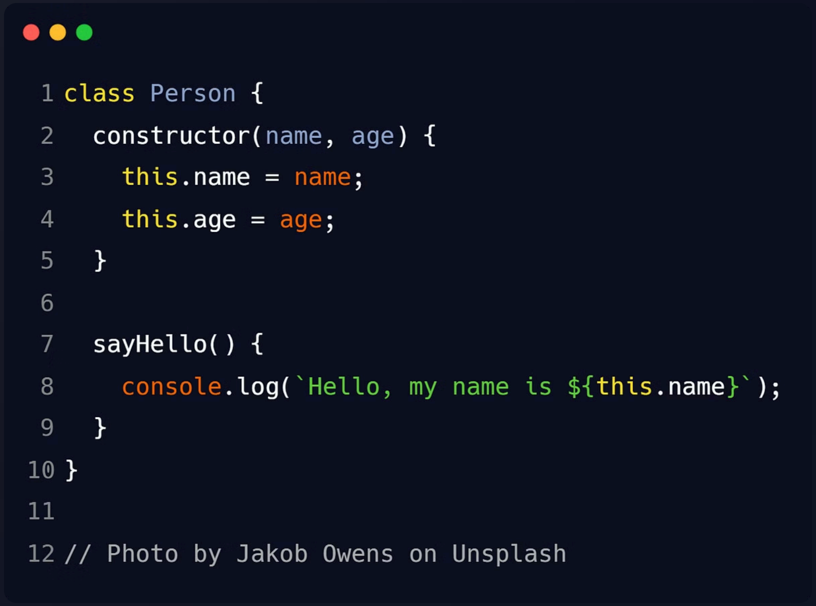
Keywords like `if`, `else`, `for`, and `while` control the flow of your code and how it executes.

3 Operators

JavaScript uses various operators like arithmetic, assignment, comparison, and logical operators to manipulate data.

4 Comments

Comments are a vital part of code readability. They help you document your code and explain your logic for easier understanding later on.



```
class Person {
  constructor(name, age) {
    this.name = name;
    this.age = age;
  }

  sayHello() {
    console.log(`Hello, my name is ${this.name}`);
  }
}

// Photo by Jakob Owens on Unsplash
```

Made with Gamma

Data Types and Operators

JavaScript has built-in data types that represent different kinds of data. Operators are used to perform operations on these data types, like addition, subtraction, comparison, and more.

Numbers

JavaScript uses both integer and floating-point numbers. Arithmetic operators can be used for calculations like addition, subtraction, multiplication, and division.

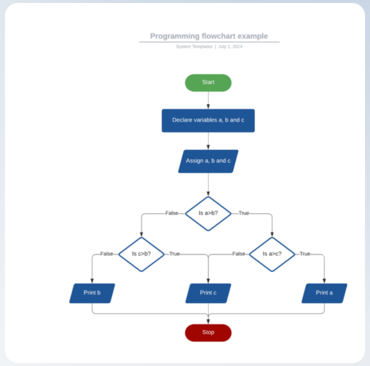
Strings

Strings represent text data. They can be concatenated using the + operator and manipulated with various built-in methods.

Booleans

Booleans are used to represent truth values (true or false) and are often used in conditional statements and logical operations.

Made with Gamma



Control Flow and Conditional Statements

Control flow determines the order in which code is executed. Conditional statements allow you to create different execution paths based on conditions.

1

if Statements

Use `if` statements to execute a block of code only if a certain condition is true.

2

else Statements

Use `else` statements to execute code if the `if` condition is false.

3

else if Statements

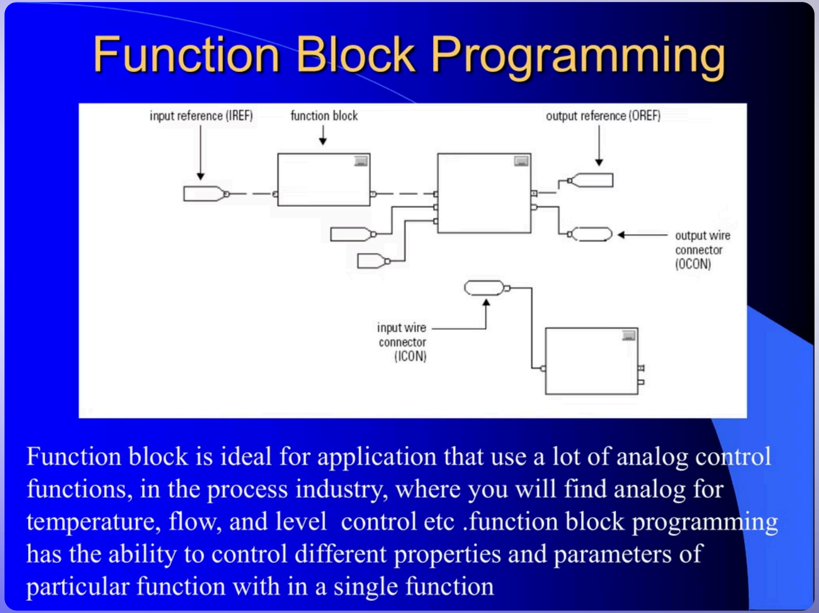
Use `else if` statements to check multiple conditions in sequence.

4

Switch Statements

Use `switch` statements to efficiently check multiple possible values against a single expression.

Made with Gamma


input reference (IREF)
function block
output reference (OREF)
output wire connector (OCON)
input wire connector (ICON)

Function Block Programming

Function block is ideal for application that use a lot of analog control functions, in the process industry, where you will find analog for temperature, flow, and level control etc .function block programming has the ability to control different properties and parameters of particular function with in a single function

Functions and Scope

Functions are blocks of reusable code that perform specific tasks. They help to organize your code and improve readability.

Function Declaration

Use the `function` keyword to declare a function, giving it a name and defining its parameters.

Function Call

To use a function, you call it by its name, passing in any required arguments.

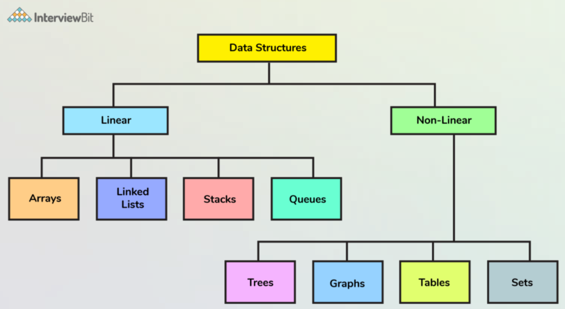
Function Scope

Variables declared inside a function have local scope, meaning they are only accessible within that function.

Return Values

Functions can return a value using the `return` keyword, which can then be used in other parts of your code.

Made with Gamma



Arrays and Objects

Arrays are used to store collections of data, while objects represent key-value pairs. They are fundamental data structures used for organizing and managing information in your code.

1

Arrays

Arrays can contain different data types and elements can be accessed using their index (starting from 0).

2

Objects

Objects have key-value pairs, where keys are strings and values can be any data type. You can access properties using dot notation or bracket notation.

3

Loops

Loops like `for` and `while` are used to iterate through arrays and objects, allowing you to access and manipulate their elements efficiently.

Made with Gamma

DOM Manipulation

The Document Object Model (DOM) represents the structure of an HTML document as a tree-like structure. JavaScript can interact with the DOM to dynamically modify website content and behavior.

Adding Elements

You can create new HTML elements using JavaScript and insert them into the existing DOM.

Removing Elements

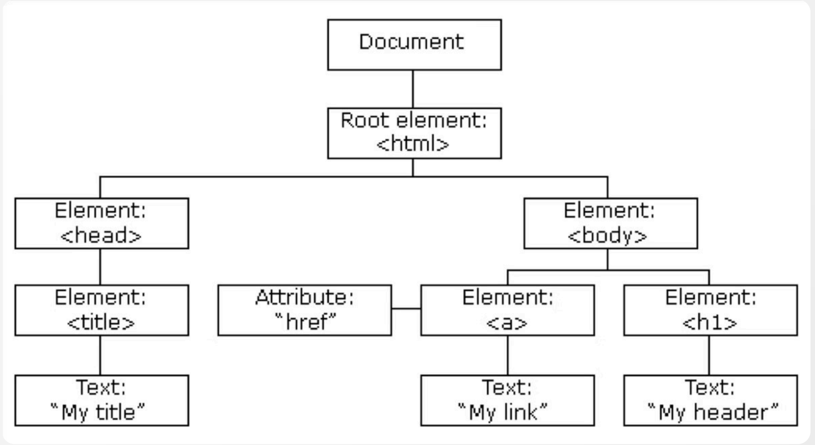
You can remove elements from the DOM, altering the structure of your webpage.

Modifying Elements

You can change the content, style, and attributes of existing elements in the DOM.

Event Handling

You can attach event listeners to elements in the DOM to respond to user interactions like clicks, hovers, or key presses.

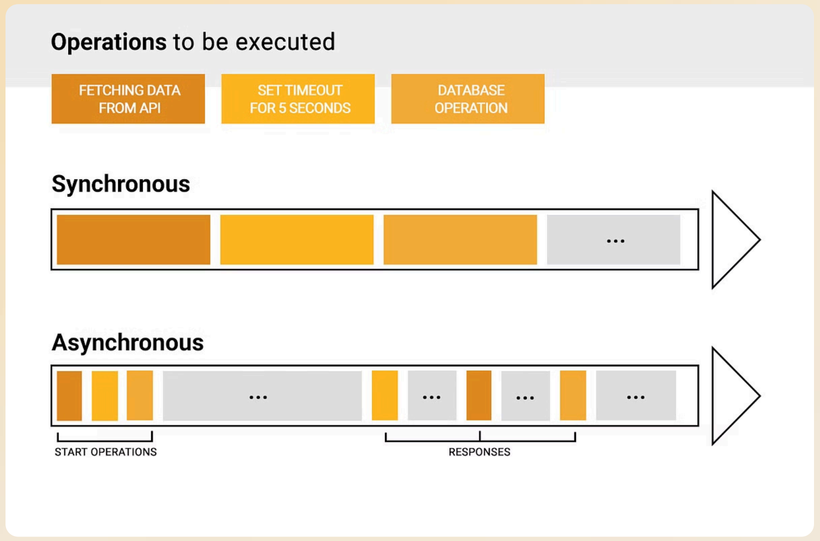


Made with Gamma

Asynchronous JavaScript and Callbacks

Asynchronous JavaScript allows code to execute without blocking the main thread, making it possible to perform tasks like fetching data from a server without interrupting the user's experience.

Callback Functions	A callback function is a function that is passed as an argument to another function and is executed when a certain event occurs.
Promises	Promises represent the eventual result of an asynchronous operation. They offer a cleaner and more structured approach to handling asynchronous code compared to traditional callbacks.
Async/Await	Async/await syntax provides a more readable and synchronous-like way to write asynchronous code, using the `async` keyword for asynchronous functions and `await` to pause execution until a promise resolves.



Made with Gamma