



# Web Development

## Fundamentals of JavaScript & Client-Side Frameworks

Explore the core concepts of JavaScript and dive into the world of client-side frameworks for building dynamic web applications.





```
11 },
12
13 // Set *default* container specific settings.json values on container create.
14 "settings": {
15     "terminal.integrated.shell.linux": "/bin/bash",
16     "python.pythonPath": "/usr/local/bin/python",
17     "python.linting.enabled": true,
18     "python.linting.pylintEnabled": true,
19     "python.formatting.autopep8Path": "/usr/local/py-utils/bin/autopep8",
20     "python.formatting.blackPath": "/usr/local/py-utils/bin/black",
21     "python.formatting.yapfPath": "/usr/local/py-utils/bin/yapf",
22     "python.linting.banditPath": "/usr/local/py-utils/bin/bandit",
23     "python.linting.flake8Path": "/usr/local/py-utils/bin/flake8",
```

# JavaScript Basics

1

## Variables

Containers for storing values.

2

## Operators

Perform arithmetic, assignment, comparison, and logical operations.

3

## Data Types

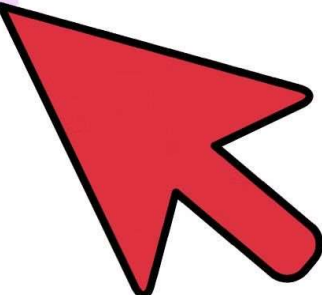
Primitive types like numbers, strings, and booleans, as well as complex objects and arrays.

4

## Control Structures

Conditional statements and loops to control the flow of your code.

```
<script language="javascript">  
var sum, x, y; //  
  
sum = x = y = 0; //  
</script>  
  
</body>  
</html>
```



# Variables

## Definition

A container for a value.

## Example

```
let number = 5; const name = "ABUNESH R P";
```

# Operators

## Arithmetic

+, -, \*, /, %

## Assignment

=, +=, -=, \*=, /=

## Comparison

==, ===, !=, !==, >, <, >=, <=

A diagram illustrating JavaScript data types. At the top, a light pink rounded rectangle contains three dark red boxes labeled 'Null', 'Undefined', and 'Symbol'. Below this, on a dark blue background, is the title 'Data Types' in white. Under the title are two dark blue boxes: 'Primitive' (listing Number, String, Boolean, Null, Undefined, Symbol, and BigInt) and 'Complex' (listing Object (Arrays, Functions, Dates, etc.)).

Null

Undefined

Symbol

## Data Types

### Primitive

Number, String, Boolean, Null, Undefined,  
Symbol, BigInt

### Complex

Object (Arrays, Functions, Dates, etc.)

# Control Structures

1

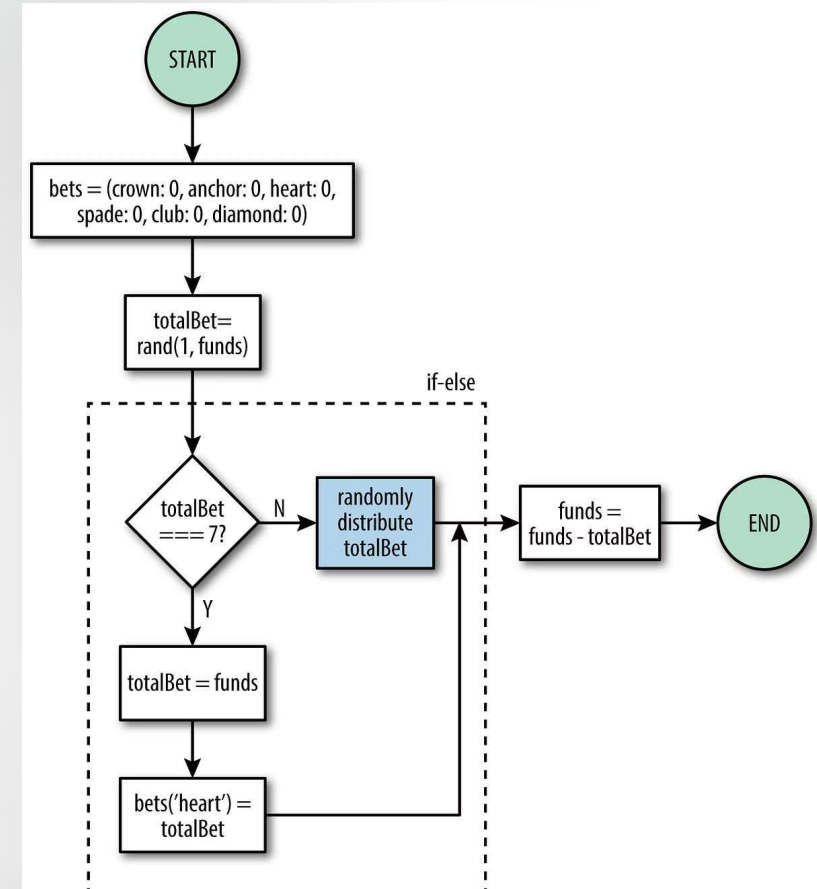
## Conditional Statements

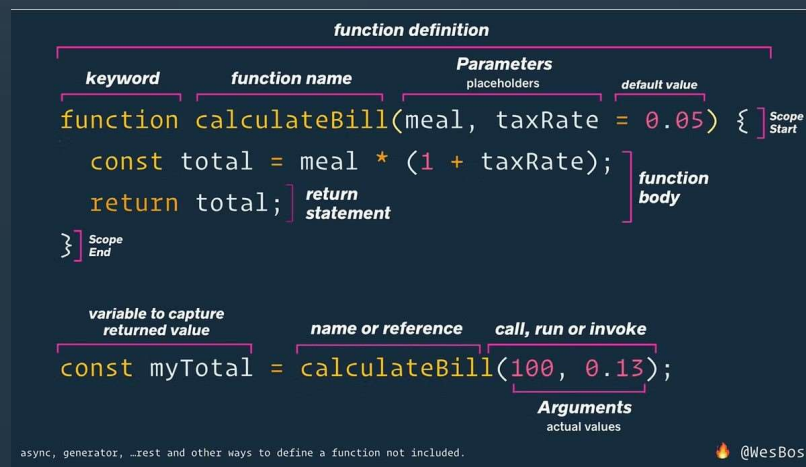
if, else, else if, switch

2

## Loops

for, while, do...while





# Functions

## Definition

Blocks of code designed to perform a task.

## Syntax

```
function functionName(parameters) { /* code */ }
```

## Parameters vs Arguments

Parameters are variables, arguments are the actual values passed to the function.

# Objects and Arrays

## Objects

Key-value pairs, mutable.

## Arrays

Ordered collections, accessed by indices.



## Lexical scope in JavaScript

- In Java, there are only two scopes:
  - **global scope**: global environment for functions, vars, etc.
  - **function scope**: every function gets its own inner scope

```
var x = 10; // foo.js
function main() {
  print(x);
  x = 20;
  if (x > 0) {
    var x = 30;
    print(x);
  }
  var x = 40;
  var f = function(x) { print(x); }
  f(50);
}
```

4

# Scope and Closures

1

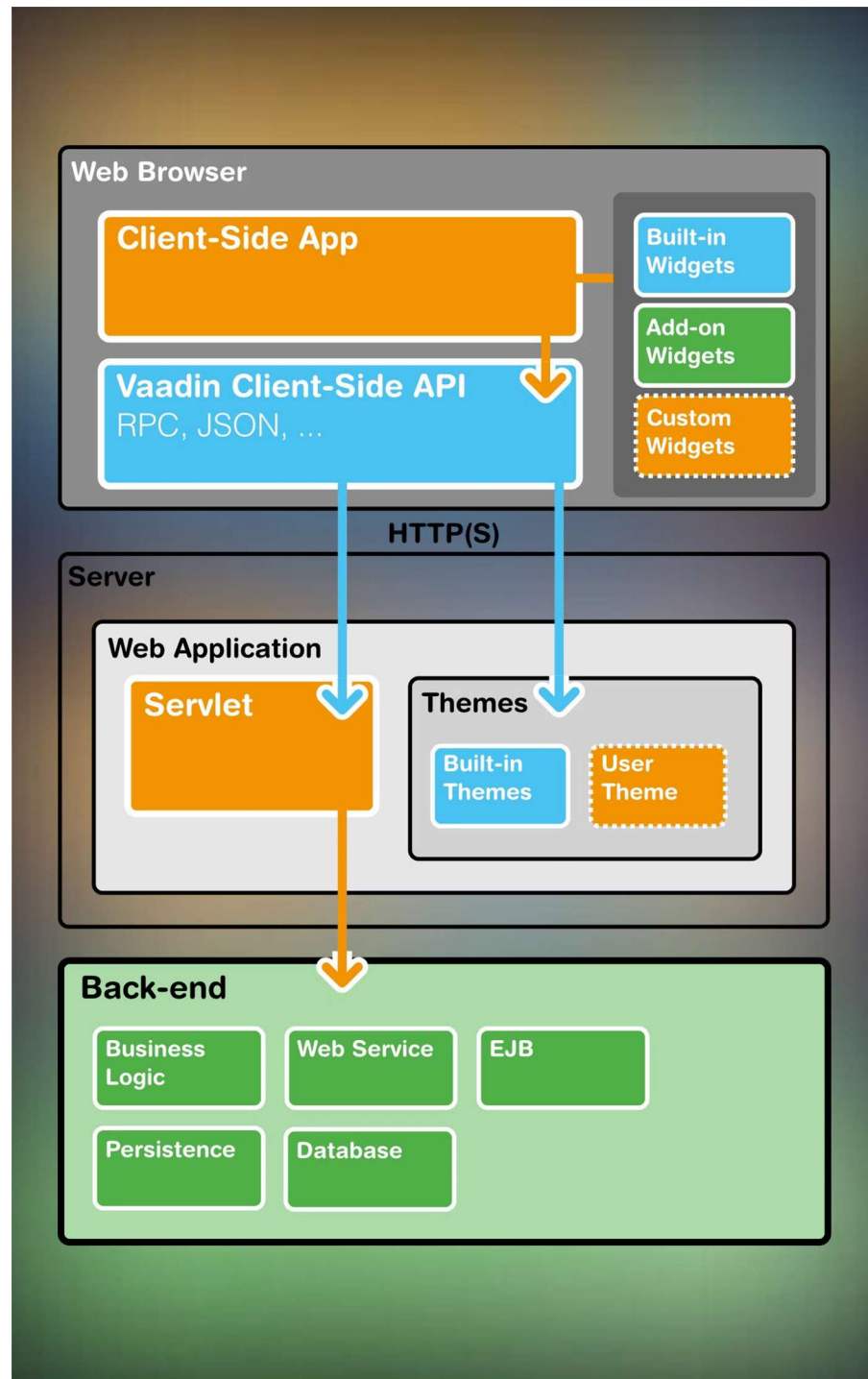
## Scope

Global, Function, Block

2

## Closures

Functions with access to parent scope variables



# Client-Side Frameworks



## React

A JavaScript library for building user interfaces.



## Angular

A comprehensive framework for building complex web applications.



## Vue.js

A progressive framework for building user interfaces.