What is JavaScript

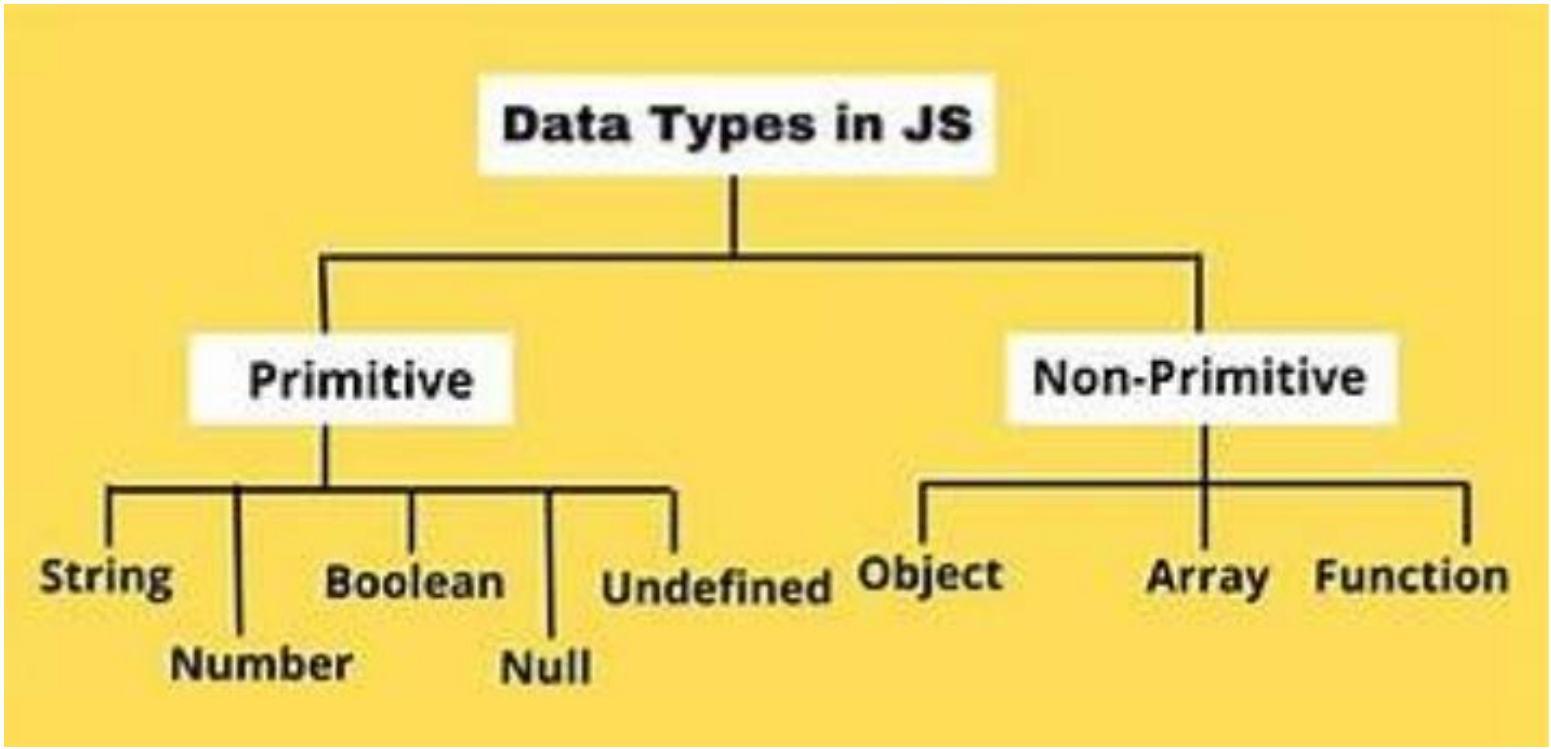
JavaScript is a , high-level programming language commonly used in web development. It was originally developed by Netscape in 1995 as a means to add dynamic and interactive elements to websites.

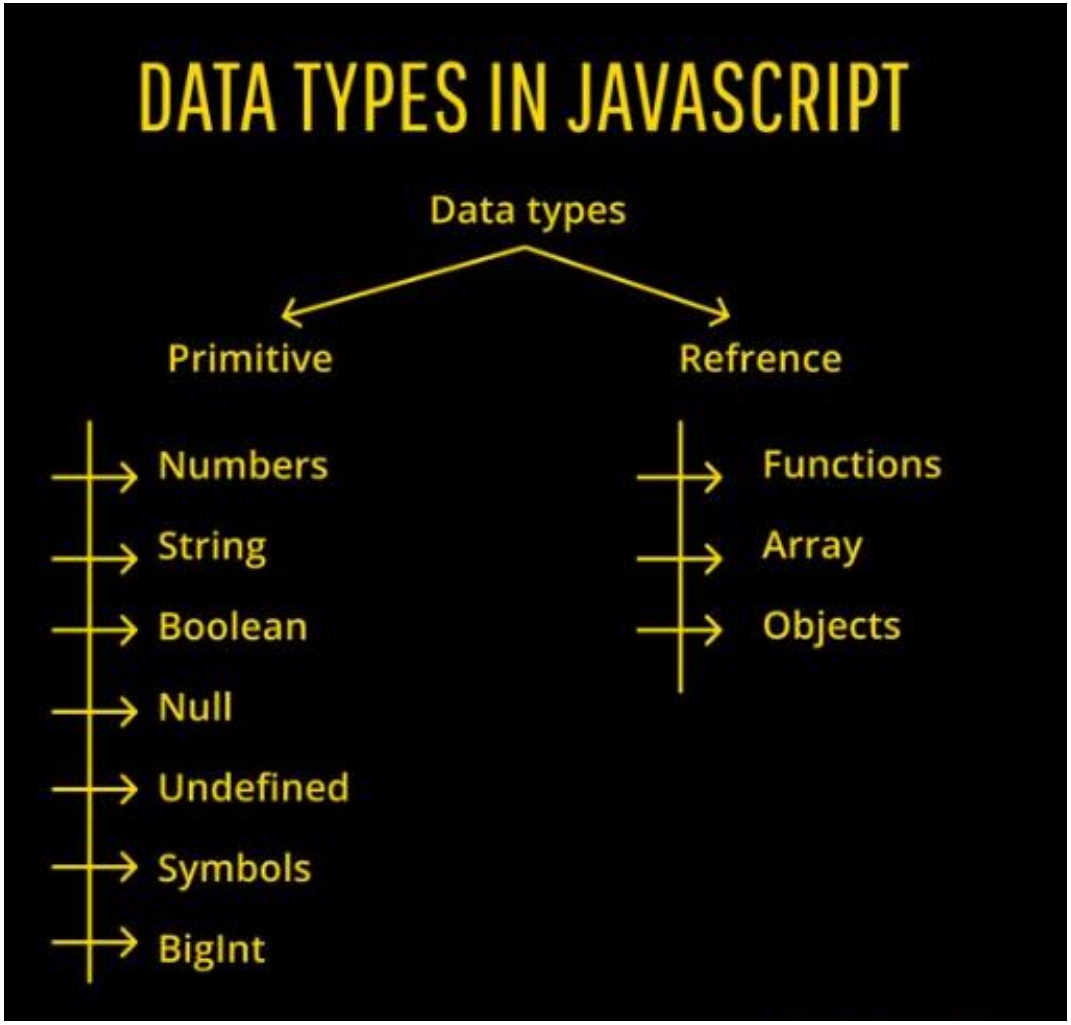
Key Features of JavaScript:

1. **Client-Side Scripting**: JavaScript is predominantly used as a client-side scripting language. This means it runs in the user's web browser without the need for server-side processing, allowing for dynamic content updates, form validations, and interactive UI elements.
2. **Event-Driven Programming**: JavaScript is well-suited for event-driven programming. It can respond to user actions such as clicks, mouse movements, and keyboard inputs, making it ideal for creating responsive and interactive web pages.
3. **Compatibility**: JavaScript can be integrated with other web technologies like HTML and CSS. It can manipulate the DOM (Document Object Model), allowing developers to change the structure, content, and style of web pages on the fly.
4. **Wide Range of Libraries and Frameworks**: There are numerous libraries (like jQuery) and frameworks (such as Angular, React, and Vue.js) built on top of JavaScript that simplify complex tasks and streamline the development process.
5. **Server-Side Development**: With the advent of Node.js, JavaScript has also become popular for server-side development. Node.js allows developers to use JavaScript to write server-side code, making it possible to use the same language for both client and server development.

What is data types

Data types in JavaScript define the data type that a variable can store.
JavaScript includes primitive and non-primitive data types




DATA TYPES IN JAVASCRIPT
Data types
Primitive
Refrence
Numbers
String
Boolean
Null
Undefined
Symbols
BigInt
Functions
Array
Objects

JavaScript has several built-in data types that are used to represent various kinds of data. These data types can be broadly categorized into primitive types and object types.

Primitive Data Types

1. **Number**: Represents both integer and floating-point numbers. Example: `42`, `3.14`.
2. **String**: Represents sequences of characters used to store text. Example: `"hello"`, `'world'`.
3. **Boolean**: Represents logical values, either `true` or `false`.
4. **Undefined**: A variable that has been declared but not assigned a value is of type `undefined`.
5. **Null**: Represents the intentional absence of any object value. It's a special keyword and type in JavaScript.
6. **Symbol**: Represents a unique and immutable value, often used to identify object properties uniquely.
7. **BigInt**: Used to represent integers that are too large to be represented by the `Number` type. Example: `1234567890123456789012345678901234567890n`.

1. Number

The `Number` type represents both integer and floating-point numbers.

javascript

```javascript
let integer = 42;
let float = 3.14;

console.log(typeof integer); // "number"
console.log(typeof float);   // "number"
```

2. String

The `String` type is used to represent textual data.

javascript

```javascript
let singleQuoteString = 'Hello';
let doubleQuoteString = "World";

console.log(typeof singleQuoteString); // "string"
console.log(typeof doubleQuoteString); // "string"
```

3. Boolean

The `Boolean` type has only two values: `true` and `false`.

javascript

```javascript
let isJavaScriptFun = true;
let isRaining = false;

console.log(typeof isJavaScriptFun); // "boolean"
console.log(typeof isRaining);       // "boolean"
```

4. Undefined

A variable that has been declared but not assigned a value is of type `undefined`.

javascript

```javascript
let notAssigned;

console.log(typeof notAssigned); // "undefined"
```

5. Null

The `null` type represents the intentional absence of any object value.

javascript Copy code

```javascript
let emptyValue = null;

console.log(typeof emptyValue); // "object" (this is a known quirk in JavaScript, `null` i
```

6. Symbol

The `Symbol` type is used to create unique identifiers.

javascript

```javascript
let uniqueID = Symbol('id');


console.log(typeof uniqueID); // "symbol"
```

7. BigInt

The `BigInt` type is used for integers that are too large to be represented by the `Number` type.

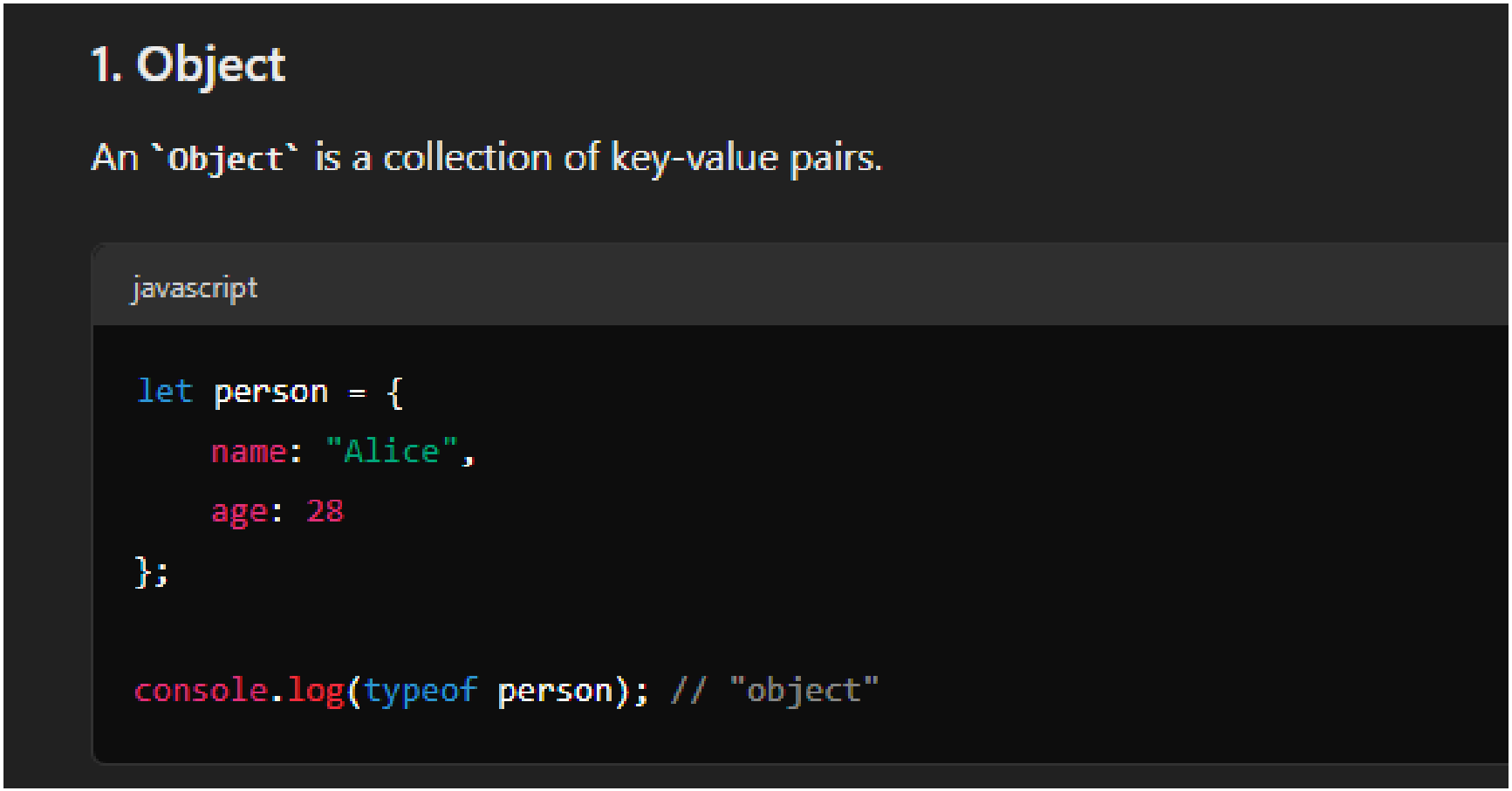
javascript Copy code

```javascript
let bigNumber = 123456789012345678901234567890123456789n;

console.log(typeof bigNumber); // "bigint"
```

Non-primitive data types

In JavaScript, non-primitive data types are objects. These include regular objects, arrays, functions



2. Array

An `Array` is an ordered collection of values.

javascript

```javascript
let colors = ["red", "green", "blue"];

console.log(typeof colors); // "object"
```

3. Function

A `Function` is a block of code designed to perform a particular task.

javascript

```javascript
function greet() {
    console.log("Hello, world!");
}

console.log(typeof greet); // "function"
```