Datatypes

data types define the type of values that can be used and manipulated in a program.

Here are the main data types in JavaScript:

1. Primitive Data Types
2. Composite Data Types / non- Primitive Data Types
3. Special Data Types

**Primitive Data Types:**

    - These are basic data types that store single values and stored in stack.

   - They are immutable, meaning their values cannot be changed.

   - Primitive data types are stored directly in memory.

   - Examples :

   - Number: Represents numeric values, including integers and floating-point numbers.

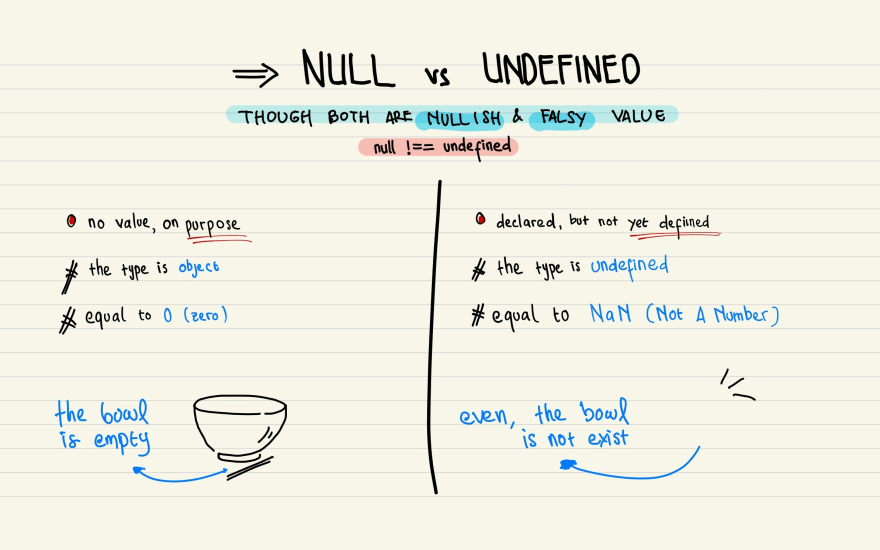
   - String: Represents sequences of characters, enclosed in single (`'`) or double (`"`) quotes.

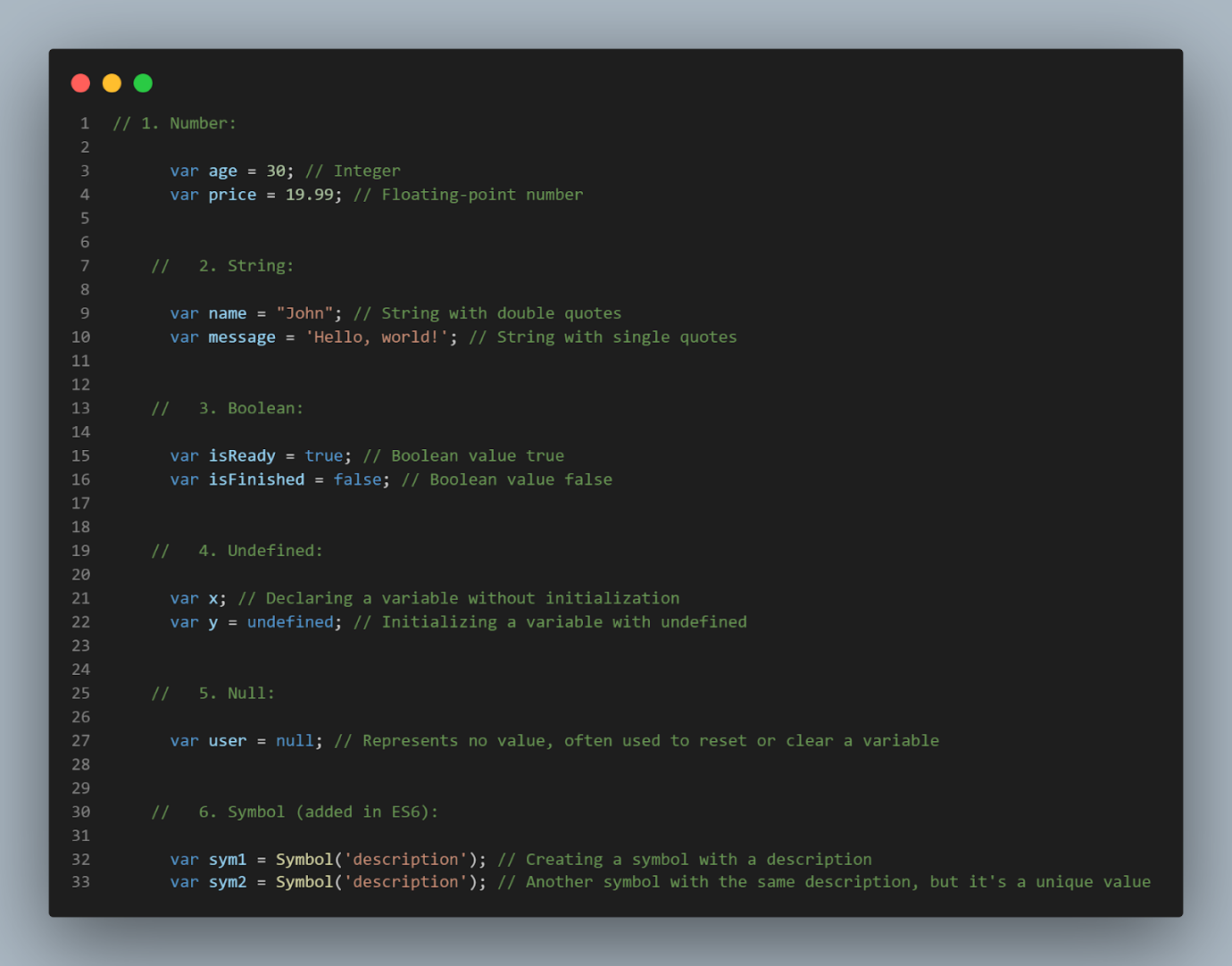
   - Boolean: Represents logical values `true` and `false`.

   - Undefined: Represents a variable that has been declared but not initialized, or a non-existent property of an object.

   - Null: Represents an intentional absence of any value.

   - Symbol (added in ES6): Represents unique identifiers.





Composite Data Types / non- Primitive Data Types

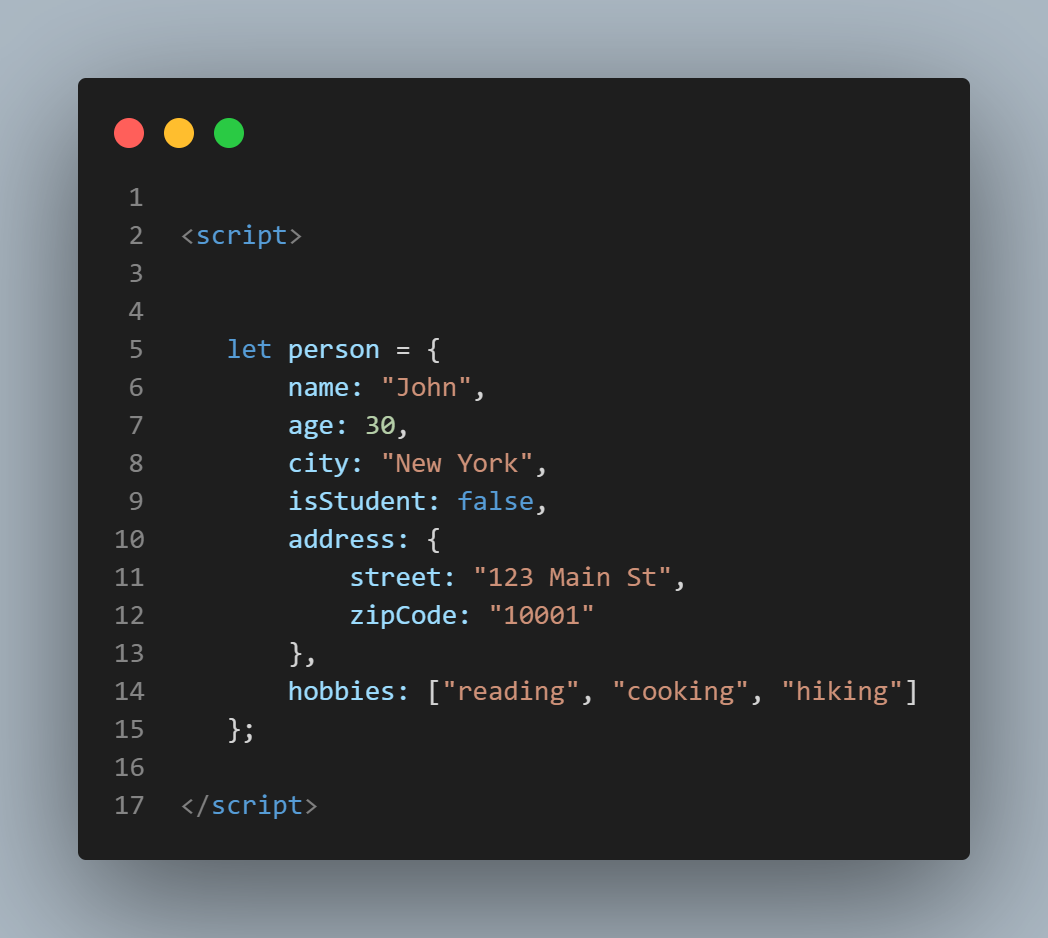
   - These are more complex data types that can store collections of values or references to other values and are stored in heap.

   - They are mutable, meaning their values can be changed.

   - Non-primitive data types are stored as references to memory locations.

   - Examples :

Objects: Objects are like containers that hold many different pieces of information, each identified by a name. They can hold numbers, strings, other objects, and more.



 Arrays: Arrays are like lists in JavaScript. They can hold multiple values, and each value has an index number that identifies its position in the list



Special Data Types:

   - Function: Functions in JavaScript are a type of object that can be invoked.

   - Date: Represents dates and times.

   - RegExp: Represents regular expressions for pattern matching.

Function:

   - Functions in JavaScript are objects that can be invoked or called.

   - They can be defined using function declarations, function expressions, arrow functions, or methods on objects.

   - Functions can accept parameters and return values.

Date:

   - The `Date` object in JavaScript represents dates and times.

   - It can be used to create new Date objects that represent specific dates or times.

   - It provides methods for manipulating dates, such as getting the current date and time, setting specific date components, and calculating differences between dates.

RegExp:

   - The `RegExp` object in JavaScript represents regular expressions, which are patterns used for pattern matching within strings.

   - Regular expressions can be used to search, match, and replace substrings within strings based on specific patterns.

   - They provide a powerful tool for text manipulation and validation.

We will deal deep in functions, date ,regexp in their particular chapter .

JavaScript also has some additional types and concepts related to handling data, such as:

- Primitive Wrapper Objects: JavaScript provides objects that wrap primitive data types (`Number`, `String`, `Boolean`, `Symbol`) to provide additional functionality.

- Type Conversion: JavaScript automatically converts values between different data types as needed, known as type coercion.

- NaN (Not-a-Number): Represents a value that is not a valid number.

- Infinity and -Infinity: Represent positive and negative infinity, respectively.

- Typeof Operator: Used to determine the data type of a variable or expression.