DATA TYPES

PRIMITIVE VALUES

The predefined data types provided by JavaScript language are known as primitive data types. Primitive data types are also known as in-built data types.

- Null
- Undefined
- BigInt
- Boolean
- Number
- String
- Symbol

OBJECTS

The data types that are derived from primitive data types of the JavaScript language are known as non-primitive data types. It is also known as derived data types or reference data types.

- Object
- Array

NULL

The Null type is inhabited by exactly one value.

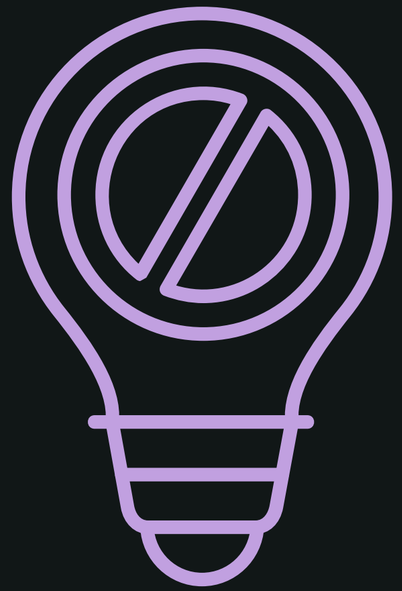
```
let z = null;
console.log(z); // Output: null
```

UNDEFINED

The Undefined type is inhabited by exactly one value.

```
let x;
console.log(x); // Output: undefined
```

In summary, undefined is often a default value for uninitialized variables or function parameters, while null is a value that can be assigned to a variable to explicitly represent the absence of a value. Both are falsy values in JavaScript, meaning they evaluate to false in boolean contexts.



NUMBERS Data type used for decimals and integers

```
let integerNumber = 42;
let floatNumber = 3.14;
```

STRINGS Strings are a fundamental data type in JavaScript used to represent text

```
let greeting = 'Hello, World!';
```

STRING - TEMPLATE LITERAL'S

- Template literals are a new type of string literal that allow embedded expressions and multiline strings.

```
const name = 'John';
const greeting = `Hello, ${name}!`;
```

```
const multilineString = `This is
a multiline
string.`;
```

BOOLEAN

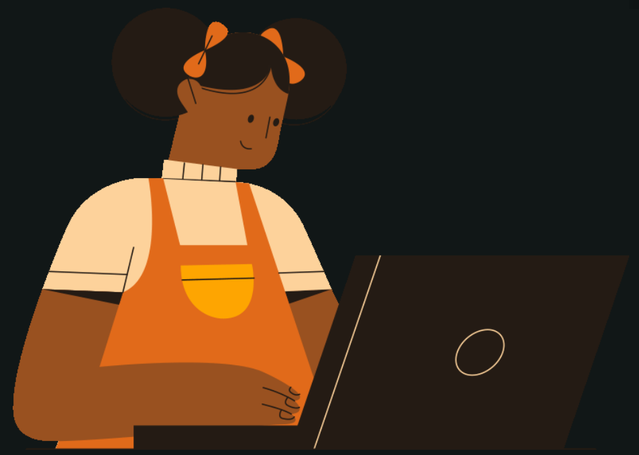
- **Boolean:** A data type representing either **true** or **false**.
- Booleans are essential for logical operations and decision-making in programming.

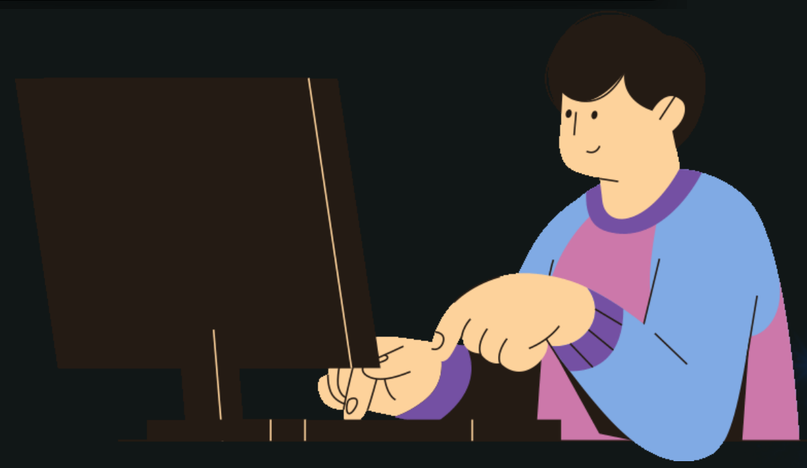
```
let x = 10;
let y = 5;
let isGreater = x > y; // true
```

TYPEOF

typeof is a built-in operator in JavaScript used to determine the data type of a given variable or expression

```
let a = 10;
console.log(typeof a) // number
```





BIGINT

```
const bigInteger =
1234567890123456789012345678901234567890n;
console.log(typeof bigInteger) // bigint

// BigInt literals are created by
appending the letter "n" to the end of an
integer literal.
```

- BigInt is a new primitive data type introduced in ECMAScript 2020 (ES11).
- It addresses the limitation of traditional numbers in JavaScript, allowing representation of integers beyond the 2^53-1 limit.

BIGINT

```
const bigInteger =
1234567890123456789012345678901234567890n;
console.log(typeof bigInteger) // bigint

// BigInt literals are created by
appending the letter "n" to the end of an
integer literal.
```

- BigInt is a new primitive data type introduced in ECMAScript 2020 (ES11).
- It addresses the limitation of traditional numbers in JavaScript, allowing representation of integers beyond the 2^53-1 limit.