Name: Mohamed Hamdi Elsayed

Task(9)

1. Unary Plus (+)

The unary plus operator is a **unary operator** used to convert values into numbers. When applied to a value, it attempts to convert it into the number data type. This operator is a quick and efficient way to perform type conversions in JavaScript.

Example

```
console.log(+"42");          // 42 (String converted to number)
console.log(+true);          // 1 (Boolean true converted to number)

console.log(+false);         // 0 (Boolean false converted to number)

console.log(+"");            // 0 (Empty string converted to number)

console.log(+"abc");         // NaN (String "abc" cannot be converted to a
number)
```

Notes:

- If the value cannot be converted to a number, the result will be NaN.
- It is often preferred over the Number() function due to its conciseness and speed.

2. NaN Type (Not-a-Number)

NaN is a special value in JavaScript that indicates that the result is not a valid number. This value is used to represent computational errors or failed type conversions that do not produce valid numbers.
Properties of NaN:

The type of NaN is number:

```
console.log(typeof NaN); // "number"
```

NaN is not equal to itself :

```
console.log(NaN === NaN); // false
```

Checking for NaN:

Since NaN does not equal itself, traditional comparison operators cannot be used to check for it. Instead, use the following functions:

isNaN():

```
console.log(isNaN(NaN));        // true
console.log(isNaN("hello"));    // true (Internally converts "hello" to
NaN)
```

Notes:

The isNaN() function converts inputs to numbers before checking, which can lead to unexpected results.

Number.isNaN():

```
console.log(Number.isNaN(NaN));     // true
console.log(Number.isNaN("hello")); // false
```

Notes:

Number.isNaN() does not convert inputs and strictly checks if the value is NaN.

Examples of NaN results:

```
console.log(Math.sqrt(-1));    // NaN (Square root of a negative number)

console.log(0 / 0);            // NaN (Division by zero)
console.log(parseInt("abc"));  // NaN (Parsing a non-numeric string)
```

---

3. Null Type (null)

- null represents intentional absence of a value in JavaScript.
- It is a primitive type with only one value: null.
- typeof null returns "object" (a historical JavaScript bug).
- It differs from undefined, which means a variable has been declared but not assigned a value.
- null == undefined → true (same value, different type).
- null === undefined → false (different types).

Example

```
let data = null;
console.log(data); // null
```


Use null when explicitly clearing a variable's value. 🚀
