Here are conceptual examples of JavaScript data types, along with short explanations:

---

1. Primitive Data Types

These are immutable and stored by value.

String

- Represents a sequence of characters.

```
let greeting = "Hello, World!";
console.log(typeof greeting); // Output: "string"
console.log(greeting.length); // Output: 13
```

Number

- Represents both integers and floating-point numbers.

```
let age = 25;
let price = 19.99;
console.log(typeof age); // Output: "number"
console.log(typeof price); // Output: "number"
```

BigInt

- Represents large integers beyond the Number type limit.

```
let bigNumber = 1234567890123456789012345678901234567890n;
console.log(typeof bigNumber); // Output: "bigint"
```

Boolean

- Represents a logical entity (true or false).

```
let isLoggedIn = true;
console.log(typeof isLoggedIn); // Output: "boolean"
```

Undefined

- A variable that has been declared but not assigned a value.

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

Null

- Represents the intentional absence of any object value.

```
let emptyValue = null;
console.log(typeof emptyValue); // Output: "object" (this is a known quirk of JavaScript)
```

Symbol

- A unique and immutable value often used as object property keys.

```
let sym = Symbol("id");
console.log(typeof sym); // Output: "symbol"
```

---

2. Non-Primitive Data Types (Objects)

These are mutable and stored by reference.

Object

- A collection of key-value pairs.

```
let person = { name: "John", age: 30 };
console.log(typeof person); // Output: "object"
console.log(person.name); // Output: "John"
```

Array

- A special type of object used for storing ordered collections.

```
let colors = ["red", "green", "blue"];
console.log(typeof colors); // Output: "object"
console.log(colors[0]); // Output: "red"
```

Function

- A callable object that can perform actions.

```
function greet(name) {
return `Hello, ${name}!`;
}
console.log(typeof greet); // Output: "function"
console.log(greet("Alice")); // Output: "Hello, Alice!"
```

Date

- An object for handling dates and times.

```
let today = new Date();
console.log(typeof today); // Output: "object"
console.log(today.toDateString()); // Output: "Thu Jan 23 2025"
```

RegExp

- An object for pattern matching using regular expressions.

```
let pattern = /hello/i;
console.log(typeof pattern); // Output: "object"
console.log(pattern.test("Hello, world!")); // Output: true
```

---

3. Special Cases

Dynamic Typing

- A variable can hold values of different types at different times.

```
let value = 42; // Initially a number
console.log(typeof value); // Output: "number"

value = "forty-two"; // Now a string
console.log(typeof value); // Output: "string"
```

Type Conversion

- Implicit or explicit conversion between types.

```
            let num = "42"; // String
            let convertedNum = Number(num); // Explicit conversion
            console.log(typeof convertedNum); // Output: "number"

            let implicitConversion = "5" * 2; // Implicit conversion to
number
            console.log(typeof implicitConversion); // Output: "number"
```

Infinity and NaN3

- Special Number values.

```
        console.log(1 / 0); // Output: Infinity
        console.log("abc" * 2); // Output: NaN (Not a Number)
```

Would you like practical tasks, code challenges, or deeper explanations for any of these?