

★ JavaScript Functions with Parameters - Explained Simply

A **function** in JavaScript is a reusable block of code that performs a task **whenever you call it**.

🔧 **Functions become more powerful when they accept arguments**, allowing them to be more flexible instead of performing only one fixed task.

◆ Example Without Parameters (Fixed Output)

This function always **alerts "Hello, there."** because the message is **hardcoded**:

```
function greetUser() {  
  alert("Hello, there.");  
}
```

```
// Calling the function
```

```
greetUser();
```

✓ **Problem:** What if we want to **change the greeting message** without modifying the function?

◆ Function with Parameters (Flexible Output)

Now, let's modify the function so it can accept **custom messages** as arguments:

```
function greetUser(greeting) {  
  alert(greeting);  
}
```

```
// Calling the function with different messages
```

```
greetUser("Hello, there!"); // Shows: Hello, there!
```

```
greetUser("Good morning!"); // Shows: Good morning!
```

```
greetUser("Welcome to our site!"); // Shows: Welcome to our site!
```

✓ Now, the function **does not have a fixed message**. It displays **whatever message** is passed to it!

◆ How Function Parameters Work

Concept	Explanation
Parameter	A placeholder variable inside the function definition (e.g., <code>greeting</code>).
Argument	The actual value passed to the function (e.g., <code>"Hello, there!"</code>).
Dynamic Behavior	The function can now work with any greeting instead of just <code>"Hello, there."</code> .

Key Rule:

★ The parameter **catches the argument** passed into the function.

Example:

```
greetUser("Hello, World!");
```

Here, `"Hello, World!"` is **passed as an argument**, and the **parameter `greeting` stores this value** inside the function.

◆ Using Variables as Arguments

Instead of passing **direct strings**, we can use **variables**:

```
var message = "Good afternoon!";
```

```
greetUser(message);
```

✓ The variable `message` holds `"Good afternoon!"`, which is passed as an argument to `greetUser`.

◆ Function with Multiple Parameters

Functions can **accept multiple arguments**:

```
function showMessage(name, message) {  
  alert(name + " says: " + message);  
}
```

// Calling the function

```
showMessage("Alice", "Hi, everyone!"); // Alice says: Hi, everyone!  
showMessage("Bob", "How are you?");    // Bob says: How are you?
```

✓ **JavaScript matches arguments to parameters in order** (first argument → first parameter, second argument → second parameter, etc.).

◆ Using Numbers as Arguments

You can also **pass numbers** to functions:

```
function showBigNum(num) {  
  alert("The big number is: " + num);  
}
```

// Calling the function with a number

```
showBigNum(999999);
```

✓ **Works with variables too:**

```
var bigNumber = 1000000;  
showBigNum(bigNumber);
```

◆ Passing Different Data Types as Arguments

A function can accept **a mix of variables, strings, and numbers**:

```
function showWinner(month, message, number) {  
  alert(month + message + number);  
}
```

// Calling the function

```
showWinner("March", "'s winner number is ", 23);
```

// Output: "March's winner number is 23"

✓ Here, the arguments are:

1. "March" (String)
2. "'s winner number is " (String)
3. 23 (Number)

They are passed to the function **in the same order** as the parameters.

🔑 Key Takeaways

✓ Functions **save time** and **avoid repetition**.

✓ **Parameters** allow functions to work **dynamically** instead of being fixed.

✓ **Arguments** are values passed when calling a function, and **parameters** catch those values.

✓ You can pass **multiple arguments** (Strings, Numbers, Variables, etc.).