

## ◆ Function কী? (সহজ ভাষায়)

👉 **Function** হলো একটা কাজের প্যাকেট (**machine**) যেখানে—

- তুমি কাজটা একবার লিখবে
- তারপর যতবার দরকার, ততবার ব্যবহার করবে

📌 যেমন:

“চা বানানো”

তুমি প্রতিবার নতুন করে রেসিপি লেখো না—একই নিয়ম বারবার ব্যবহার করো

👉 এটাকেই প্রোগ্রামিং-এ বলে **function**

---

## ◆ Function কেন দরকার?

- ❶ একই কোড বারবার লিখতে হয় না
  - ❷ কোড ছোট ও পরিষ্কার হয়
  - ❸ বোঝা ও ঠিক করা সহজ হয়
- 

## ◆ Function লেখার গঠন (Structure)

```
function functionName(){  
    // এখানে কাজ লেখা হবে  
}
```

👉 এখানে:

- **function** → keyword
- **functionName** → function এর নাম
- **{ }** → function এর শরীর (body)

---

### ◆ Example 1: Simple Function (সবচেয়ে সহজ)

```
function sayHello(){  
  console.log("Hello World");  
}
```

? এখন প্রশ্ন:  
এটা কি নিজে নিজে চলবে?  
👉 না ❌

চালাতে হলে ডাকতে হবে 📌

```
sayHello();
```

📌 Output:

```
Hello World
```

---

### ◆ Function Call মানে কী?

👉 **Function** লেখা = রেসিপি লেখা  
👉 **Function call** = রান্না করা

```
sayHello(); // function call
```

---

### ◆ Example 2: Function বারবার ব্যবহার

```
function greet(){  
  console.log("Good Morning");  
}
```

```
greet();  
greet();  
greet();
```

📌 Output:

Good Morning  
Good Morning  
Good Morning

👉 একবার লিখে, বারবার ব্যবহার 👍

---

## ◆ Function with Parameter (ডাটা পাঠানো)

এখন একটু গুরুত্বপূর্ণ 👁👁

```
function greet(name){  
  console.log("Hello " + name);  
}
```

এখানে:

- **name** → **parameter** (function এর ভেতরের variable)

Call করার সময় value পাঠাবো:

```
greet("Nishat");  
greet("Rahim");
```

📌 Output:

Hello Nishat  
Hello Rahim

👉 Function যেন একটা মেশিন

👉 তুমি নাম ঢোকাও, সে greeting বের করে দেয়

---

## ◆ Parameter vs Argument (খুব সহজ)

```
function greet(name){    // parameter
    console.log(name);
}
```

```
greet("Nishat");        // argument
```

📌 মনে রাখার ট্রিক:

- **Parameter** → **function** লেখার সময়
  - **Argument** → **function** চালানোর সময়
- 

### ♦ Example 3: Function দিয়ে যোগ (Addition)

```
function add(a, b){
    console.log(a + b);
}
```

```
add(5, 3);
```

📌 Output:

8

---

### ♦ Return কী? (খুব গুরুত্বপূর্ণ)

👉 `console.log` শুধু দেখায়

👉 `return` মান ফেরত দেয়

```
function add(a, b){
    return a + b;
}
```

```
let result = add(4, 6);
console.log(result);
```

📌 Output:

---

### ◆ console.log vs return (difference)

console.log	return
শুধু দেখায়	value ফেরত দেয়
ব্যবহার করা যায় না	পরে ব্যবহার করা যায়

---

### ◆ Example 4: Return ছাড়া সমস্যা

```
function add(a, b){  
  console.log(a + b);  
}
```

```
let x = add(2, 3);  
console.log(x);
```

📌 Output:

```
5  
undefined
```

👉 কারণ `return` নাই ❌

---

### ◆ Function মনে রাখার 1 লাইনের নিয়ম

**Function** = কাজের নাম + কাজের কোড + দরকার হলে ডাটা নেয় + দরকার হলে ফলাফল দেয়

---

## 🟢 PRACTICE SET 1 — Function (Absolute Beginner)

---

## ✓ Practice 1: নিজের নাম দেখাও

👉 একটি function বানাও যেটা তোমার নাম print করবে।

### ◆ Code:

```
function showName(){  
  console.log("Nishat Chowdhury");  
}
```

```
showName();
```

📌 এখানে কি হলো?

- `showName` → function এর নাম
- কোনো parameter নাই
- call করলে নাম দেখাচ্ছে

---

## ✓ Practice 2: Function দিয়ে যোগ করো

👉 দুইটা সংখ্যা নিয়ে যোগ করবে।

### ◆ Code:

```
function add(a, b){  
  console.log(a + b);  
}
```

```
add(10, 5);
```

📌 `a` ও `b` হলো parameter

📌 `10` ও `5` হলো argument

---

## ✓ Practice 3: Return ব্যবহার করে যোগ

👉 এবার result পরে ব্যবহার করবো।

♦ **Code:**

```
function add(a, b){  
  return a + b;  
}
```

```
let sum = add(20, 30);  
console.log(sum);
```

📌 **return** result ফিরিয়ে দিয়েছে

📌 তাই **sum** এ value রাখা গেছে

---

## ✅ **Practice 4:** বয়স **Adult** কিনা চেক

👉 বয়স 18 বা বেশি হলে "Adult" দেখাবে।

♦ **Code:**

```
function checkAge(age){  
  if(age >= 18){  
    return "Adult";  
  } else {  
    return "Not Adult";  
  }  
}
```

```
console.log(checkAge(20));  
console.log(checkAge(15));
```

📌 **Output:**

Adult

Not Adult

---

## 🟡 **Arrow Function (Modern JavaScript)**

👉 ছোট করে function লেখার উপায়

♦ **Normal Function:**

```
function multiply(a, b){  
  return a * b;  
}
```

♦ **Arrow Function:**

```
const multiply = (a, b) => {  
  return a * b;  
}
```

📌 দুইটাই একই কাজ করে

📌 Arrow function বেশি ব্যবহার হয় modern JS এ

---

♦ আরও ছোট **arrow function**

```
const multiply = (a, b) => a * b;  
  
console.log(multiply(4, 5));
```

---

## Real-Life Example

👉 দোকানে মোট দাম বের করা

```
function totalPrice(price, quantity){  
  return price * quantity;  
}
```

```
let bill = totalPrice(50, 4);  
console.log(bill);
```

📌 Output:

200

---



## মনে রাখার **Shortcut**

- `function` = কাজের মেশিন
- `parameter` = মেশিনের ইনপুট
- `return` = মেশিনের আউটপুট