জাভাস্ক্রিপ্টে **ফাংশন** হলো কোডের একটি ব্লক যা নির্দিষ্ট কাজ সম্পাদন করে। ফাংশন ব্যবহার করে আপনি কোডকে পুনরায় ব্যবহারযোগ্য (reusable) এবং মডুলার করতে পারেন। ফাংশন একবার ডিফাইন করার পর আপনি একাধিক বার কল করে ব্যবহার করতে পারেন। নিচে জাভাস্ক্রিপ্টে ফাংশনের বিস্তারিত ব্যাখ্যা দেওয়া হলো:

**১. ফাংশন ডিক্লেয়ারেশন:**

জাভাস্ক্রিপ্টে ফাংশন ডিক্লেয়ার করতে function কিওয়ার্ড ব্যবহার করা হয়।

**সিনট্যাক্স:**

javascript

Copy

function functionName(parameter1, parameter2) {

// Code to execute

return result; // Optional

}

**উদাহরণ:**

javascript

Copy

function greet(name) {

return "Hello, " + name;

}

console.log(greet("John")); // আউটপুট: Hello, John

**২. ফাংশন কল:**

ফাংশন ডিক্লেয়ার করার পর আপনি ফাংশনের নাম এবং প্যারামিটার পাস করে কল করতে পারেন।

**উদাহরণ:**

javascript

Copy

function add(a, b) {

return a + b;

}

let result = add(2, 3); // ফাংশন কল

console.log(result); // আউটপুট: 5

**৩. ফাংশনের ধরন:**

জাভাস্ক্রিপ্টে ফাংশন বিভিন্ন ধরনের হতে পারে। নিচে প্রধান ধরনগুলি আলোচনা করা হলো:

**ক. নেমড ফাংশন (Named Function):**

* এটি একটি সাধারণ ফাংশন যা একটি নাম দিয়ে ডিক্লেয়ার করা হয়।
* উদাহরণ:

javascript

Copy

function multiply(a, b) {

return a \* b;

}

console.log(multiply(2, 3)); // আউটপুট: 6

**খ. অ্যানোনিমাস ফাংশন (Anonymous Function):**

* এটি একটি নামবিহীন ফাংশন। সাধারণত ভেরিয়েবলে অ্যাসাইন করা হয় বা অন্য ফাংশনের আর্গুমেন্ট হিসেবে পাস করা হয়।
* উদাহরণ:

javascript

Copy

let greet = function(name) {

return "Hello, " + name;

};

console.log(greet("Alice")); // আউটপুট: Hello, Alice

**গ. অ্যারো ফাংশন (Arrow Function):**

* ES6 এ অ্যারো ফাংশন চালু হয়। এটি একটি সংক্ষিপ্ত সিনট্যাক্স এবং this কিওয়ার্ডের আচরণ আলাদা।
* উদাহরণ:

javascript

Copy

let add = (a, b) => a + b;

console.log(add(2, 3)); // আউটপুট: 5

**ঘ. IIFE (Immediately Invoked Function Expression):**

* এটি একটি ফাংশন যা ডিক্লেয়ার করার পরপরই কল হয়।
* উদাহরণ:

javascript

Copy

(function() {

console.log("IIFE executed!");

})();

**ঙ. হাইয়ার-অর্ডার ফাংশন (Higher-Order Function):**

* এটি একটি ফাংশন যা অন্য ফাংশনকে আর্গুমেন্ট হিসেবে নেয় বা রিটার্ন করে।
* উদাহরণ:

javascript

Copy

function operate(a, b, operation) {

return operation(a, b);

}

let result = operate(2, 3, (x, y) => x + y);

console.log(result); // আউটপুট: 5

**৪. ফাংশনের প্যারামিটার এবং আর্গুমেন্ট:**

* **প্যারামিটার**: ফাংশন ডিক্লেয়ারেশনে ব্যবহৃত ভেরিয়েবল।
* **আর্গুমেন্ট**: ফাংশন কল করার সময় পাস করা ভ্যালু।

**উদাহরণ:**

javascript

Copy

function greet(name) { // name হলো প্যারামিটার

return "Hello, " + name;

}

console.log(greet("John")); // "John" হলো আর্গুমেন্ট

**৫. ফাংশনের রিটার্ন ভ্যালু:**

ফাংশন return স্টেটমেন্ট ব্যবহার করে একটি ভ্যালু রিটার্ন করতে পারে। যদি return স্টেটমেন্ট না থাকে, তাহলে ফাংশন undefined রিটার্ন করে।

**উদাহরণ:**

javascript

Copy

function add(a, b) {

return a + b;

}

let result = add(2, 3);

console.log(result); // আউটপুট: 5

**৬. ফাংশনের স্কোপ:**

ফাংশনের ভিতরে ডিক্লেয়ার করা ভেরিয়েবল শুধুমাত্র সেই ফাংশনের মধ্যে অ্যাক্সেস করা যায় (লোকাল স্কোপ)। ফাংশনের বাইরে ডিক্লেয়ার করা ভেরিয়েবল গ্লোবাল স্কোপে থাকে।

**উদাহরণ:**

javascript

Copy

let globalVar = "I am global"; // গ্লোবাল ভেরিয়েবল

function myFunction() {

let localVar = "I am local"; // লোকাল ভেরিয়েবল

console.log(globalVar); // গ্লোবাল ভেরিয়েবল অ্যাক্সেস করা যায়

console.log(localVar); // লোকাল ভেরিয়েবল অ্যাক্সেস করা যায়

}

myFunction();

// console.log(localVar); // Error: লোকাল ভেরিয়েবল বাইরে অ্যাক্সেস করা যায় না

**৭. ডিফল্ট প্যারামিটার:**

ES6 এ ডিফল্ট প্যারামিটার চালু হয়। যদি আর্গুমেন্ট পাস না করা হয়, তাহলে ডিফল্ট ভ্যালু ব্যবহার করা হয়।

**উদাহরণ:**

javascript

Copy

function greet(name = "Guest") {

return "Hello, " + name;

}

console.log(greet()); // আউটপুট: Hello, Guest

console.log(greet("John")); // আউটপুট: Hello, John

**৮. রেস্ট প্যারামিটার:**

রেস্ট প্যারামিটার ব্যবহার করে আপনি একাধিক আর্গুমেন্টকে একটি অ্যারেতে সংগ্রহ করতে পারেন।

**উদাহরণ:**

javascript

Copy

function sum(...numbers) {

return numbers.reduce((total, num) => total + num, 0);

}

console.log(sum(1, 2, 3, 4)); // আউটপুট: 10

**৯. রিকার্সিভ ফাংশন:**

একটি ফাংশন নিজেকে কল করলে তাকে রিকার্সিভ ফাংশন বলে।

**উদাহরণ:**

javascript

Copy

function factorial(n) {

if (n === 0) return 1;

return n \* factorial(n - 1);

}

console.log(factorial(5)); // আউটপুট: 120

**সারসংক্ষেপ:**

| **বৈশিষ্ট্য** | **বর্ণনা** | **উদাহরণ** |
| --- | --- | --- |
| **নেমড ফাংশন** | নাম দিয়ে ডিক্লেয়ার করা ফাংশন | function greet() { ... } |
| **অ্যানোনিমাস ফাংশন** | নামবিহীন ফাংশন | let greet = function() { ... } |
| **অ্যারো ফাংশন** | সংক্ষিপ্ত সিনট্যাক্স | let add = (a, b) => a + b; |
| **IIFE** | ডিক্লেয়ার করার পরপরই কল হয় | (function() { ... })(); |
| **হাইয়ার-অর্ডার ফাংশন** | ফাংশনকে আর্গুমেন্ট বা রিটার্ন ভ্যালু হিসেবে নেয় | function operate(a, b, operation) { ... } |
| **ডিফল্ট প্যারামিটার** | আর্গুমেন্ট না পাস করলে ডিফল্ট ভ্যালু ব্যবহার করে | function greet(name = "Guest") { ... } |
| **রেস্ট প্যারামিটার** | একাধিক আর্গুমেন্টকে অ্যারেতে সংগ্রহ করে | function sum(...numbers) { ... } |
| **রিকার্সিভ ফাংশন** | ফাংশন নিজেকে কল করে | function factorial(n) { ... } |

ফাংশন ব্যবহার করে আপনি আপনার কোডকে মডুলার, পুনরায় ব্যবহারযোগ্য এবং সহজে বোঝার মতো করতে পারবেন। 😊