



# Different ways of Function Declaration in JavaScript





# Basic Function

This is the traditional way of declaring a function using the **function keyword**.

Recommended for named functions or when you need **hoisting**.



```
function basicFunction() {  
  console.log("Basic Function");  
}
```

[ ]

;

{ }



( )



Swipe →



# Function Expressions

Creating a function as part of an expression, often assigned to a variable.

Useful for **anonymous functions** or when you want to control where the function is available.



```
var functionExpression = function() {  
  console.log("Function Expression");  
};
```



Swipe →



# Arrow Functions

Introduced in ES6, a concise way to write functions.

Especially useful for short, **one-liner functions**.



```
const arrowFunction = () =>  
  console.log("Arrow Function");
```



Swipe →



# Function Constructor

Using the Function constructor to create a new function.

Less common and has **security implications**, so use with caution.



```
var constructorFunction = new  
Function('console.log("Function Constructor")');
```





# Function Hoisting

Functions declared using function keyword are **hoisted to the top** of their **containing scope**.

Allows you to call the function before it's declared.



```
hoistedFunction();  
  
function hoistedFunction() {  
    console.log("Function Hoisting");  
}
```



Swipe →



# Variable Hoisting

Variables declared with var are **hoisted**, but **only the declaration**, not the initialization.

Function expressions are also hoisted.



```
hoistedFunction();  
  
function hoistedFunction() {  
    console.log("Function Hoisting");  
}
```







codewith**sloba**.com

Get a weekly digest of my tips and  
tutorials by subscribing now.