

JavaScript for Developers

Function

(part 1: basic)

1. function

➤ declare

```
function myFunction(param1, param2) {  
    // function's body  
}
```

- ✓ Không kiểu trả về
- ✓ Param ko có type

1. function

➤ A function can consider as a variable

✓ **Type:** function

```
function myFunction(param1, param2) {  
  console.log("this is myFunction");  
}
```

```
console.log(typeof myFunction); // => function
```

1. function

- A function can consider as a variable
 - ✓ Can be re-assign to another value

```
function myFunction(param1, param2) {  
    console.log("this is myFunction");  
}
```

```
myFunction = 1;
```

```
console.log(typeof myFunction); // => number
```

1. function

➤ A function can consider as a variable

✓ Can be used as parameter

```
function executeSomeFunction(someFunction) {  
    someFunction();  
}
```

```
executeSomeFunction(myFunction); // => this is myFunction
```

1. function

➤ A function can consider as a variable

✓ Can be declare as a variable

```
const myAnotherFunction = function () {  
    console.log("this is myAnotherFunction");  
};
```

```
myAnotherFunction(); // => this is myAnotherFunction
```

1. function

➤ A function can consider as a variable

✓ Nested function

```
function myFunction(a) {  
    function myNestedFunction() {  
        console.log("this is myNestedFunction");  
    }  
    console.log("this is myFunction");  
    myNestedFunction();  
}
```

```
myFunction();  
// Result:  
// this is myFunction  
// this is myNestedFunction
```

2. parameter

- ✓ undefined – parameter

Doesn't pass parameter to function

```
function myFunction(a, b) {  
  console.log("a = " + a); // => a = undefined  
  console.log("b = " + b); // => b = undefined  
}  
  
myFunction();
```


2. parameter

✓ default – parameter

```
function myFunction(a = 1, b = 2) {  
  console.log("a = " + a); // => a = 1  
  console.log("b = " + b); // => b = 2  
}
```

```
myFunction();
```

2. parameter

✓ **rest** – parameter

Passing un-know number of parameters to function

```
function sum(...values) {  
  console.log(Array.isArray(values)); // => true  
}  
  
sum(1, 2);
```

2. parameter

✓ overloading function

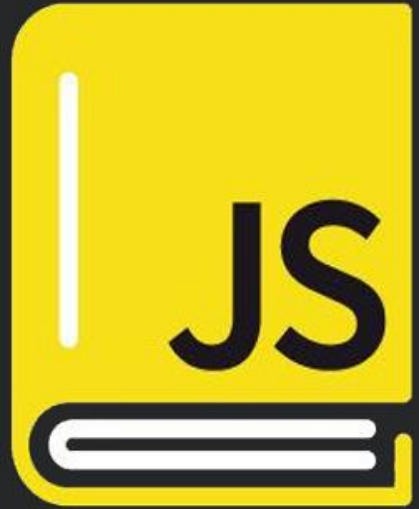
```
function f(a) {  
  console.log("a = " + a);  
}
```

```
function f(a, b) {  
  console.log("a = " + a);  
  console.log("b = " + b);  
}
```

```
const a = 1;  
f(a);  
// Result:  
// a = 1  
// b = undefined
```

3. Refs

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions>
- https://www.w3schools.com/js/js_functions.asp



JavaScript for Developers

Function

(part 1: basic)