

Regular Functions

```
function f() {  
    // returns undefined  
}
```

Function regular sin retorno

Regular function with return keyword.

```
function f() {  
    return true;  
}
```

Single parameter and return value

```
function f(name){  
    return name.toUpperCase();  
}
```

Multiple parameters and default values.

```
function f(isAlive=true, department=42){  
    return true;  
}
```

IIFE that accepts parameters

```
(function f(name, message){  
    console.log(`  
        ${name} says  
        ${message}`);  
})('Sheldon', 'Bazinga');
```

function returning an object

```
function f(){  
    return {  
        id: 123,  
        name: 'Steve'  
    }  
}  
  
function f(){  
    return (  
        {  
            id: 123,  
            name: 'Steve'  
        }  
    )  
}
```

Destructuring an object as a parameter

```
function f({id, name}){  
    // id and name are now local  
    // variables inside this function  
}  
  
let myObj = {  
    id: 123,  
    name: 'Steve'  
}  
f(myObj);
```



By Mon. (Monz gomz)
cheatography.com/monz-gomz/

Published 11th October, 2022.
Last updated 11th October, 2022.
Page 1 of 2.

Sponsored by [Readable.com](#)
Measure your website readability!
<https://readable.com>

Arrow functions

```
const f = () =>{  
    // returns undefined  
}
```

Arrow sin retorno

Arrow function with return keyword

```
const f = () => {  
    return true;  
}  
const f = () => true;
```

Single parameter and return value

```
const f = name => {  
    return name.toUpperCase();  
}  
const f = name => name.toUpperCase();
```

Multiple parameters and default values.

```
const f = (isAlive=true, department=42) => true;
```

Arrow IIFE that accepts parameters

```
((name, message)=>{  
    const $nameAndMessage=  
    ${message};  
    })('Sheldon', 'Bazinga');
```

Arrow function returning an object.

```
const f = () =>{  
    return {  
        id: 123,  
        name: 'Steve'  
    }  
}  
const f = () =>({ id: 123, name: 'Steve' } );
```

Arrow destructuring an object as a parameter

```
const f = ({id, name}) =>{  
    //id and name are now local  
variables inside this function  
}  
let myObj = {  
    id: 123,  
    name: 'Steve'  
}  
f(myObj);
```



By Mon. (Monz gomz)
cheatography.com/monz-gomz/

Published 11th October, 2022.
Last updated 11th October, 2022.
Page 2 of 2.

Sponsored by [Readable.com](https://readable.com)
Measure your website readability!
<https://readable.com>