

JS

TYPES OF ARROW FUNCTIONS IN JAVASCRIPT PART 1/2

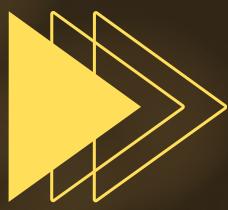


[Abuzar95](#)



[Muhammad Abuzar](#)





ARROW FUNCTIONS TYPES

In JavaScript, **arrow functions** are a more **concise** way to write function **expressions** and have a few **distinctive** features compared to **traditional** functions. They were introduced in **ECMAScript 6 (ES6)**.

We will **discuss** the **types** of **arrow functions** have.

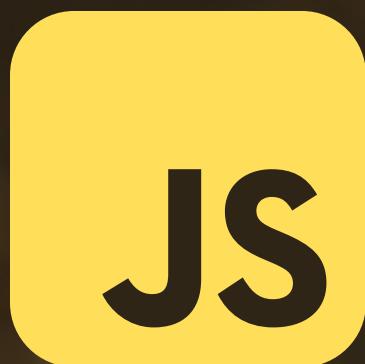
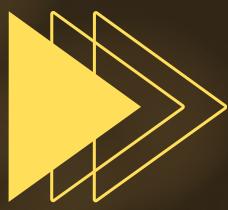


[Abuzar95](#)



[Muhammad Abuzar](#)





BASIC ARROW FUNCTION

The most **straightforward** form of an arrow function, which can replace simple function expressions.

Syntax:

```
const add = (a, b) => {  
    return a + b;  
};
```

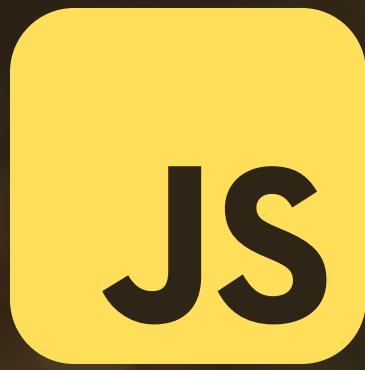
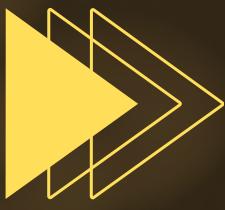


[Abuzar95](#)



[Muhammad Abuzar](#)





SINGLE EXPRESSION ARROW FUNCTION

If the **function body** has only **one expression**, you can **omit** the **curly braces** and the **return** keyword.

Syntax:

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

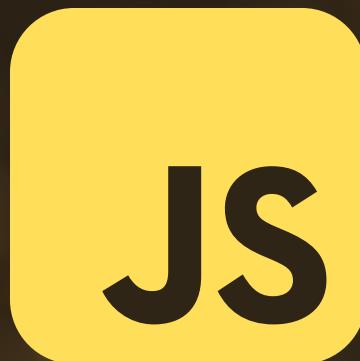
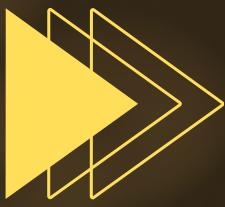


[Abuzar95](#)



[Muhammad Abuzar](#)





SINGLE PARAMETER ARROW FUNCTION

When there is only **one parameter**, you can omit the parentheses around the parameter.

Syntax:

```
const square = x => x * x;
```

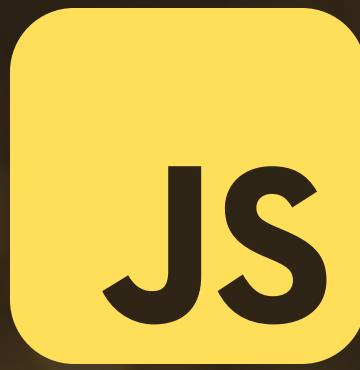
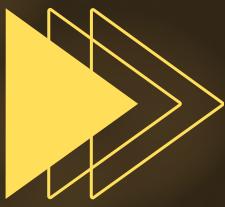


[Abuzar95](#)



[Muhammad Abuzar](#)





RETURNING AN OBJECT LITERAL

When you need to **return** an **object literal**,
you must **wrap** the **object** in **parentheses** to
avoid **ambiguity** with the block body.

Syntax:

```
const getPerson = (name, age) => ({  
    name: name,  
    age: age  
});
```

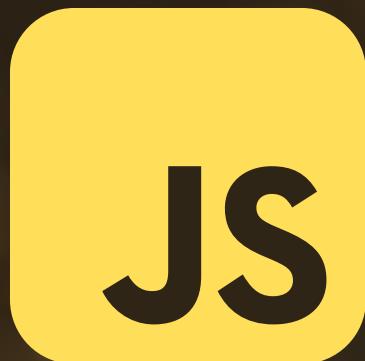
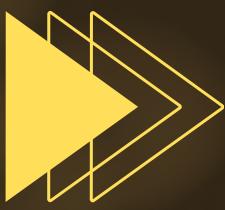


[Abuzar95](#)



[Muhammad Abuzar](#)





NO PARAMETERS ARROW FUNCTION

When the **arrow function** has **no parameters**, you use **empty parentheses**.

Syntax:

```
const sayHello = () => 'Hello, world!';
```

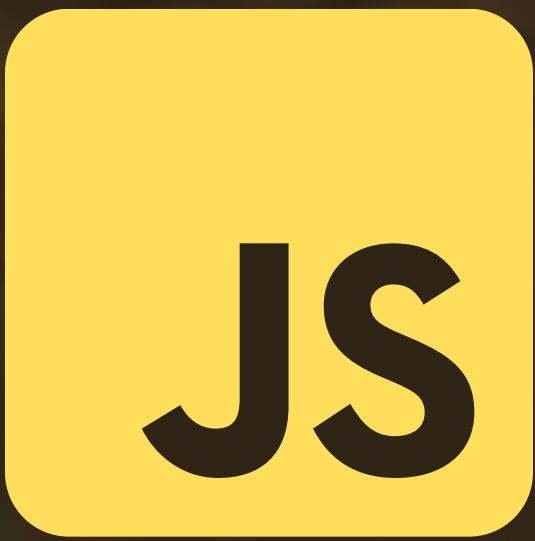
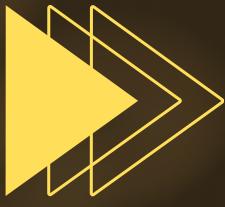


[Abuzar95](#)



[Muhammad Abuzar](#)





JS

FIND HELPFUL

Save 
Like 

Share 
Comment 



[Abuzar95](#)



[Muhammad Abuzar](#)