Jitendra Gosavi
@JS_withJeet

JAVASCRIPT FUNCTIONS

A function is a block of code that is designed to perform a specific task and execute when it gets called or invoked.

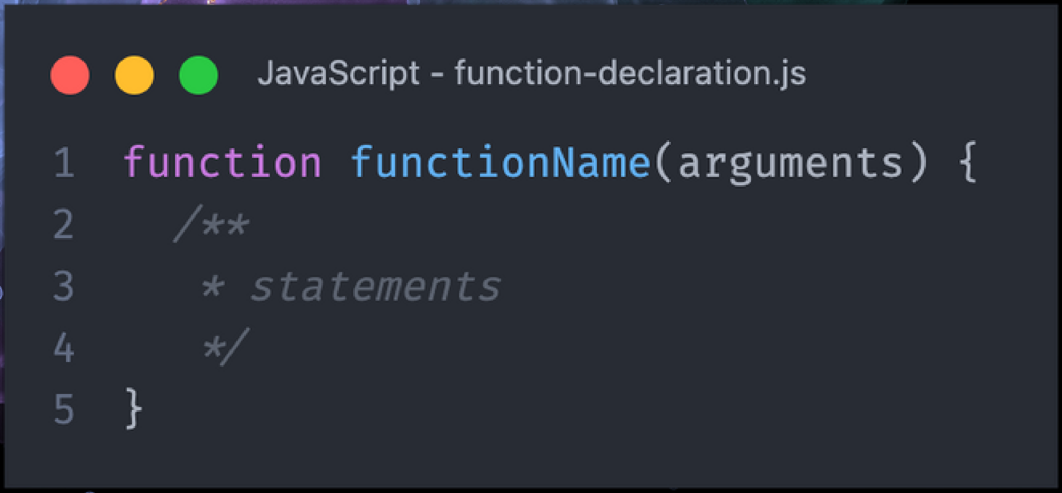
There are four types of writing functions in JavaScript

- Function declaration
- Function expression
- Constructor function
- Arrow function

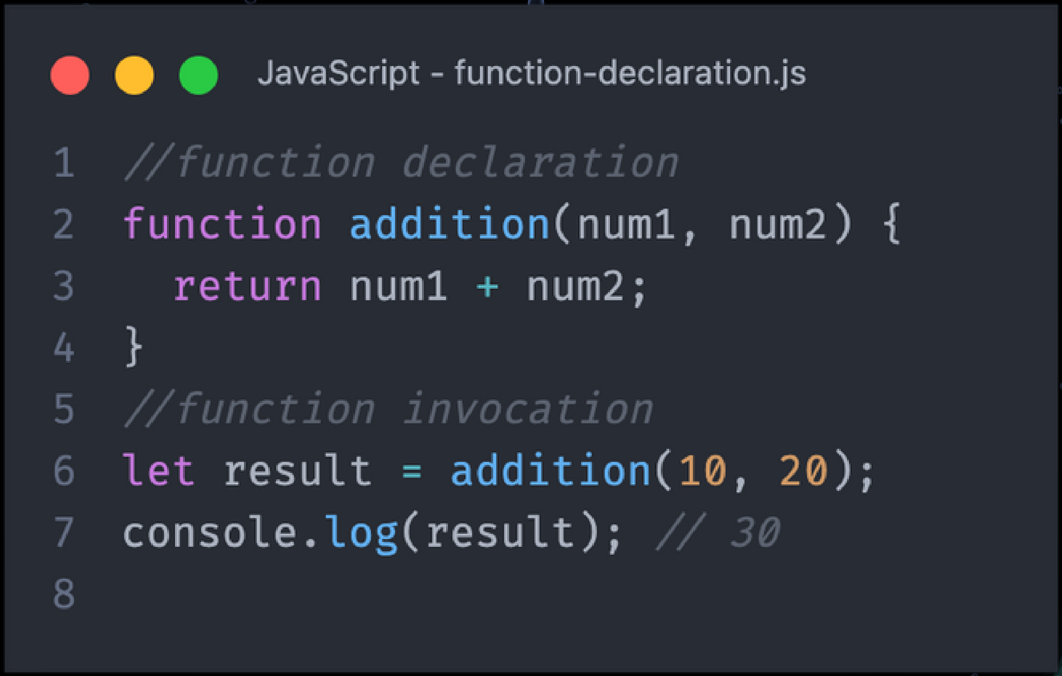
SWIPE >>>

Jitendra Gosavi
@JS_withJeet

Function Declaration



```javascript
function functionName(arguments) {
  /**
   * statements
   */
}
```



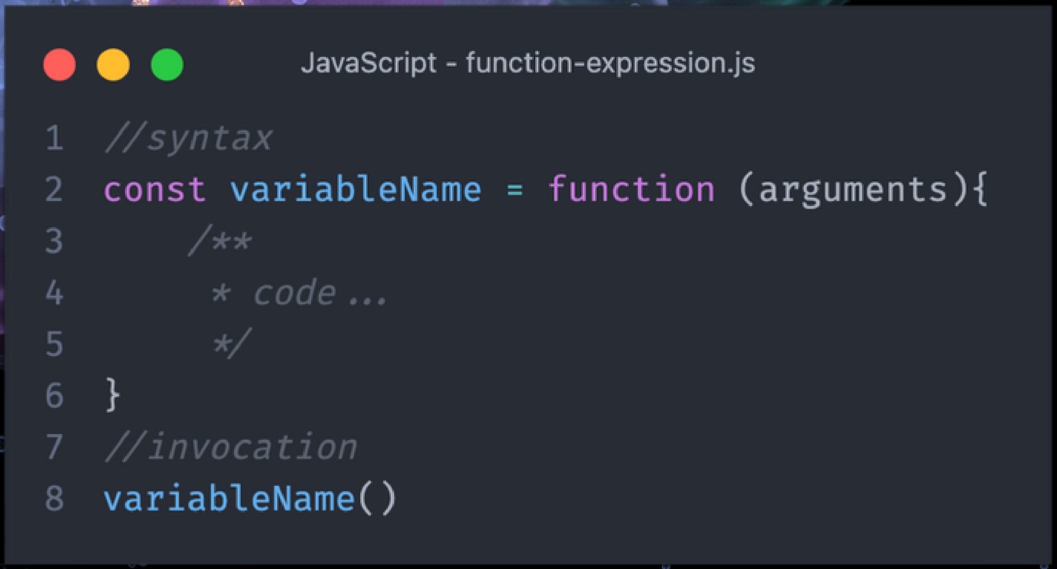
```javascript
//function declaration
function addition(num1, num2) {
  return num1 + num2;
}
//function invocation
let result = addition(10, 20);
console.log(result); // 30

```

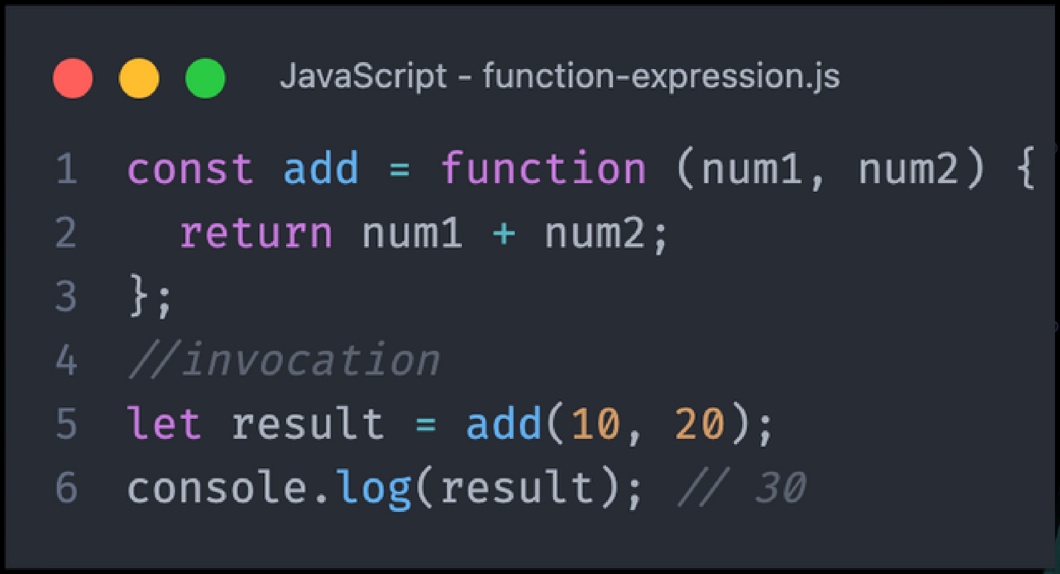
- This is a traditional way to write a function in JavaScript like other programming languages.
- We can start writing Function with function keyword followed by the function name and parentheses () then function body {}.

SWIPE >>>

Function Expression



```javascript
//syntax
const variableName = function (arguments){
    /**
     * code ...
     */
}
//invocation
variableName()
```



```javascript
const add = function (num1, num2) {
  return num1 + num2;
};
//invocation
let result = add(10, 20);
console.log(result); // 30
```

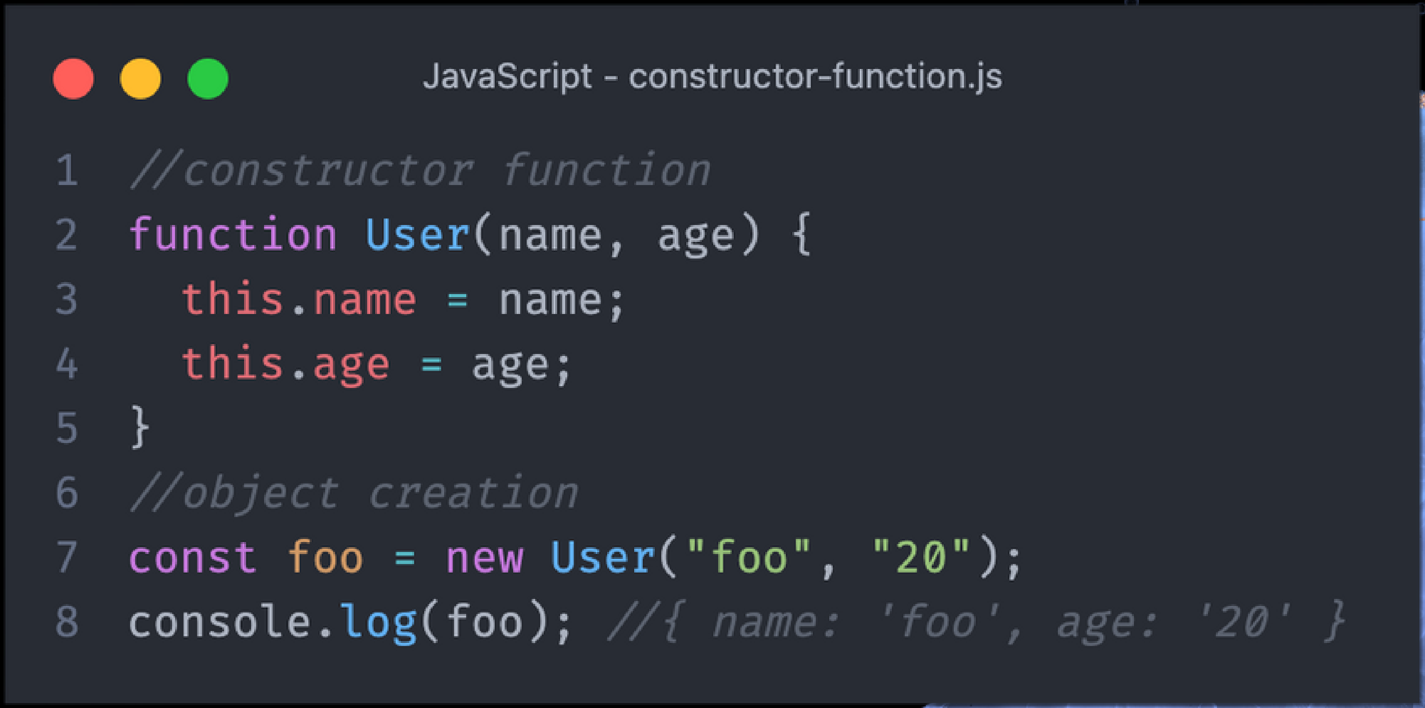
- This is another way to write a function which is not present in other programming languages,
- **we can call it an anonymous function.**
- We can write a function and **store it in a variable.**
- Function Expression **does not support hoisting.**
- We can start writing a function with variable creation and assign an anonymous function to it.

SWIPE >>>

Jitendra Gosavi
@JS_withJeet

Constructor Function

- This is a procedural way to create Objects in JavaScript but is not used much.
- In another programming language, we create a class and its object the same this we can achieve with the Constructor function.
- We can write Constructor function same as function declaration but The naming convention is different in this case we always have to write the first letter in capital letter



JavaScript - constructor-function.js

```javascript
//constructor function
function User(name, age) {
  this.name = name;
  this.age = age;
}
//object creation
const foo = new User("foo", "20");
console.log(foo); //{ name: 'foo', age: '20' }
```

SWIPE >>>

Jitendra Gosavi
@JS_withJeet

Arrow Function

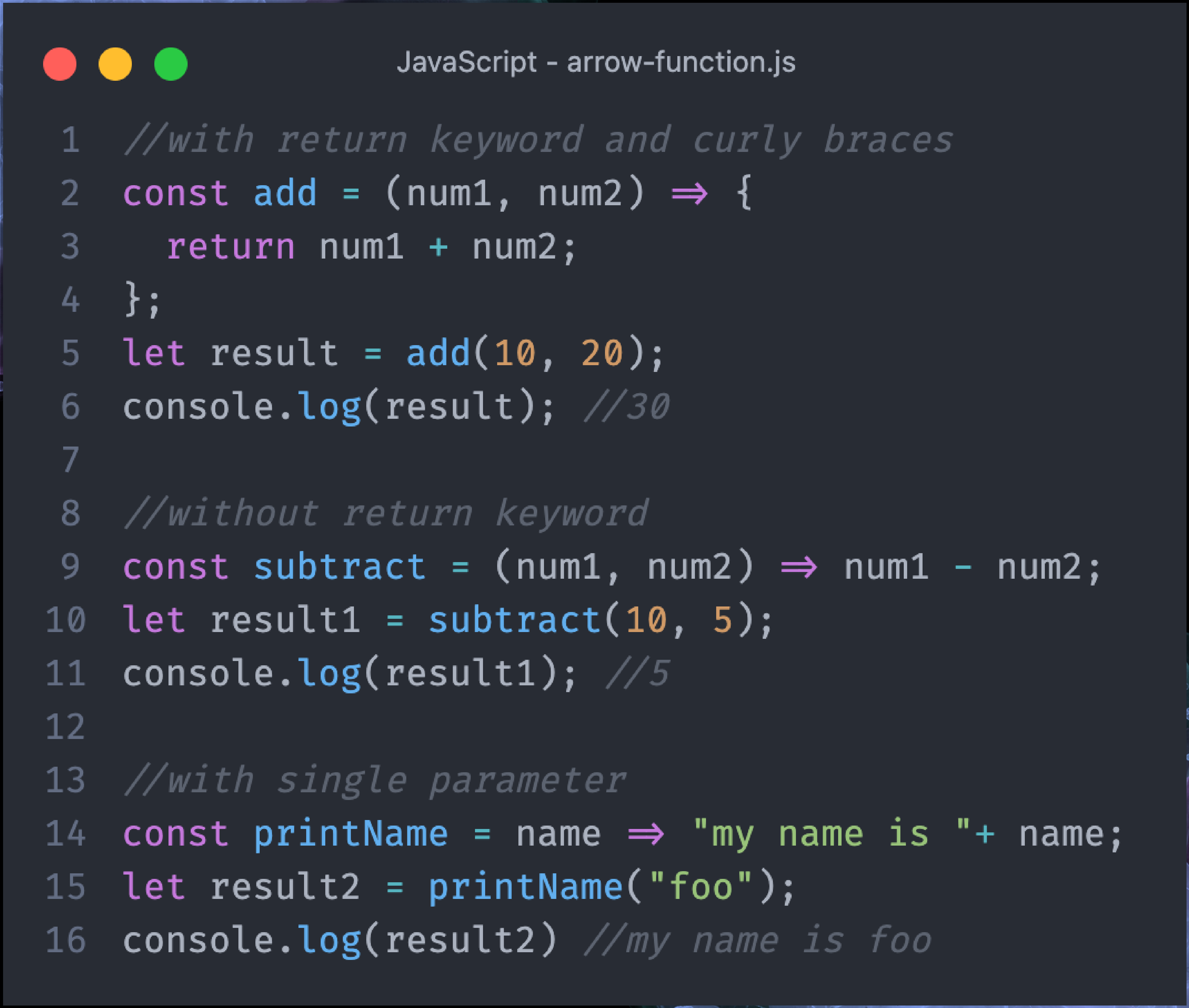
- This is a new way to write functions in JavaScript which is introduced with the release of ES6(ES2015).
- It neither requires a function keyword nor function name to write an arrow function.
- It also can be written as a function expression.
- The most important thing to remember is an Arrow function does not have its own "this" context, it always refers to the lexical environment scope.
- Arrow functions are most of the time used as callbacks in higher-order functions.
- Arrow functions privilege us to write the shortest possible code over tehe traditional functions.

SWIPE >>>

We can write a single line of code to complete the function tasks

- if the function has a single parameter then no need to use parentheses.
- if the function has a single statement and needs to return immediately then the return keyword is not required.
- if the function has multiple statements and still we don't want to use the return keyword then we can wrap it into parentheses

SWIPE >>>



JavaScript - arrow-function.js

```javascript
//with return keyword and curly braces
const add = (num1, num2) => {
  return num1 + num2;
};
let result = add(10, 20);
console.log(result); //30

//without return keyword
const subtract = (num1, num2) => num1 - num2;
let result1 = subtract(10, 5);
console.log(result1); //5

//with single parameter
const printName = name => "my name is "+ name;
let result2 = printName("foo");
console.log(result2) //my name is foo
```

SWIPE >>>

Jitendra Gosavi
@JS_withJeet

Did you find this useful?

Follow for more Amazing Posts

@JS_withJeet

in jitendragosavi

Like

Share

Comment