Types Of **Functions**

6 Ways

Mallikarjun | @CodeBustler

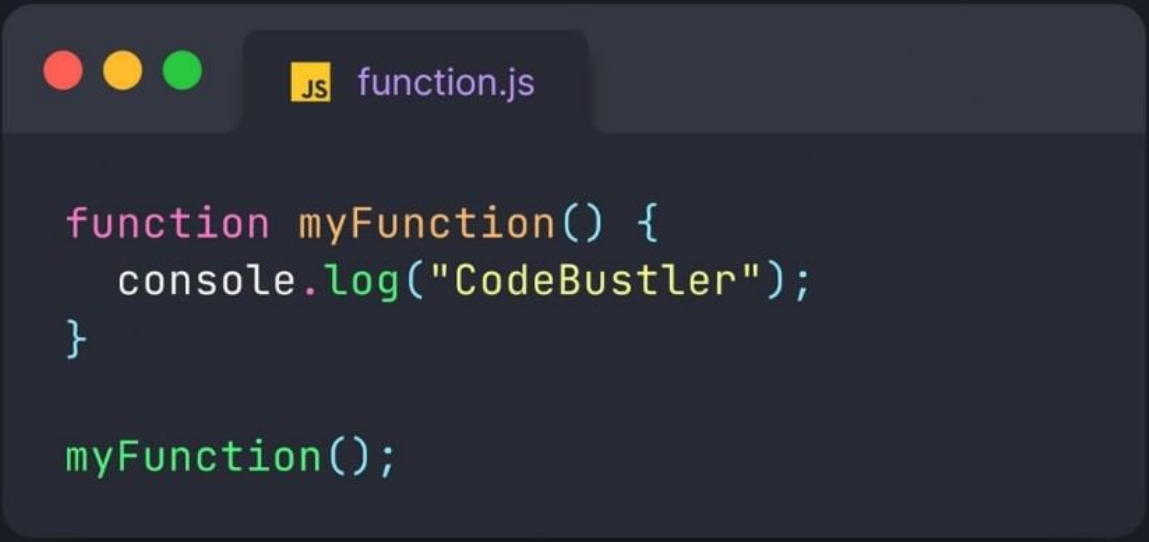
JS Functions | 6 Ways

- Named Functions
- Anonymous Functions
- Arrow Functions
- Immediately Invoked Function Expressions (IIFE)
- Higher Order Functions
- Constructor Functions

@CodeBustler

1. Named Functions

Lacacy / Traditional way



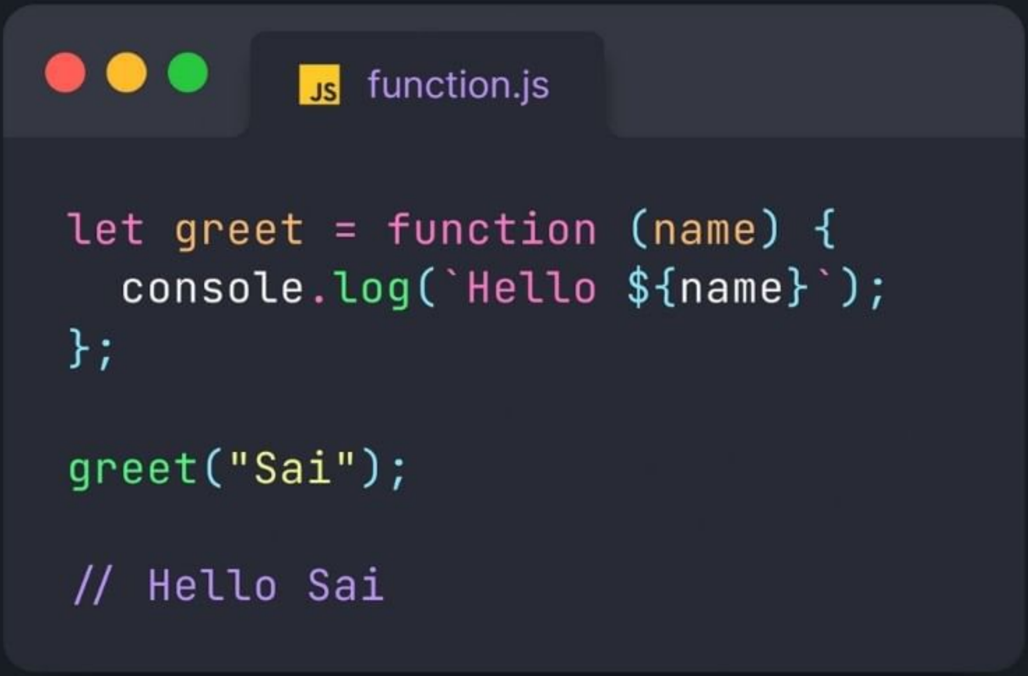
function.js

```
function myFunction() {
  console.log("CodeBustler");
}

myFunction();
```

@CodeBustler

2. Anonymous Functions

Anonymous Functions | without name, used as function expression or a arguments

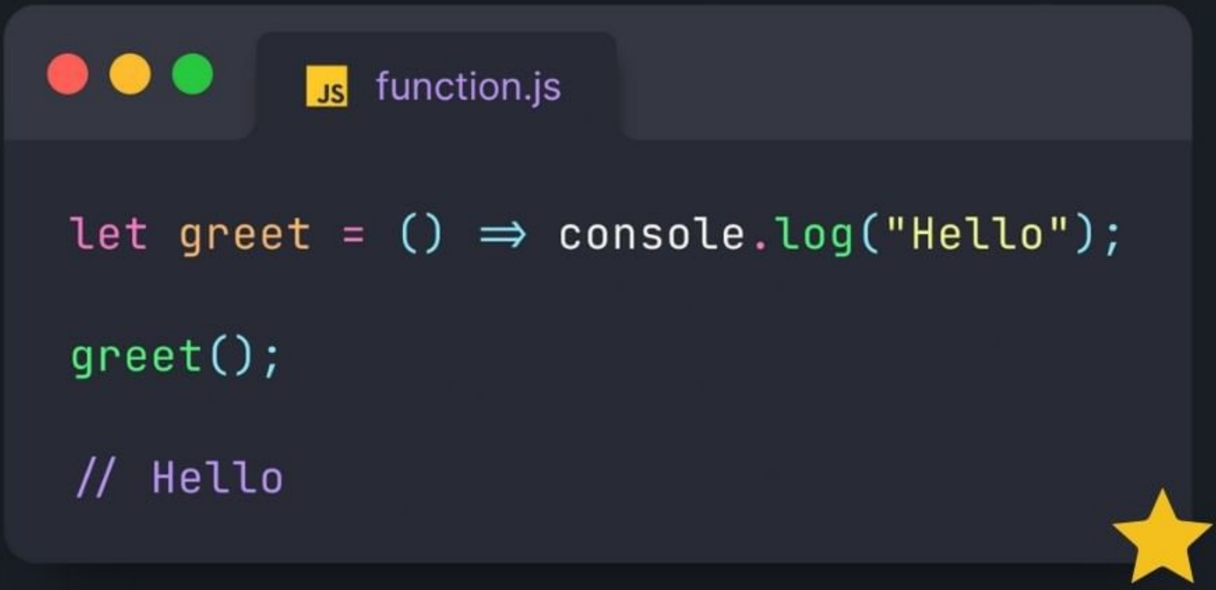


```javascript
let greet = function (name) {
  console.log(`Hello ${name}`);
};

greet("Sai");

// Hello Sai
```

@CodeBustler

3. Arrow Functions

Arrow Functions Introduced in **ES6,** Shorter Syntax & one-lined functions

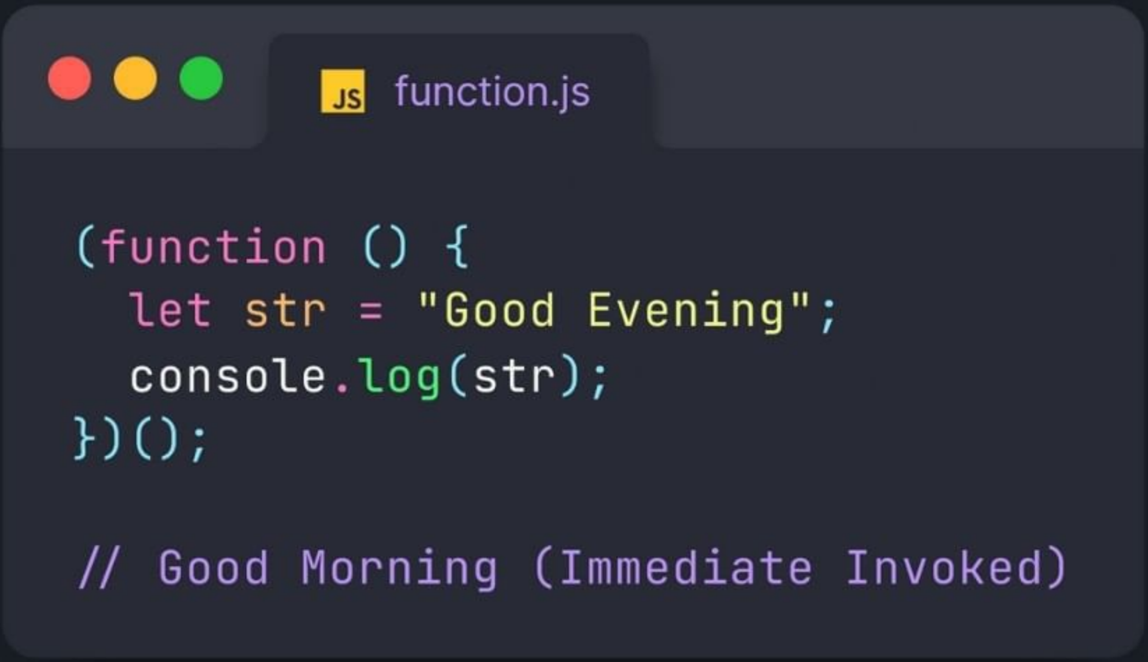


```javascript
let greet = () => console.log("Hello");

greet();

// Hello
```

@CodeBustler

4. IIFE Functions

Immediately Invoked Function Expressions (IIFE) Executed immediately after their creation. Used to create private scopes and avoid polluting the global namespace.



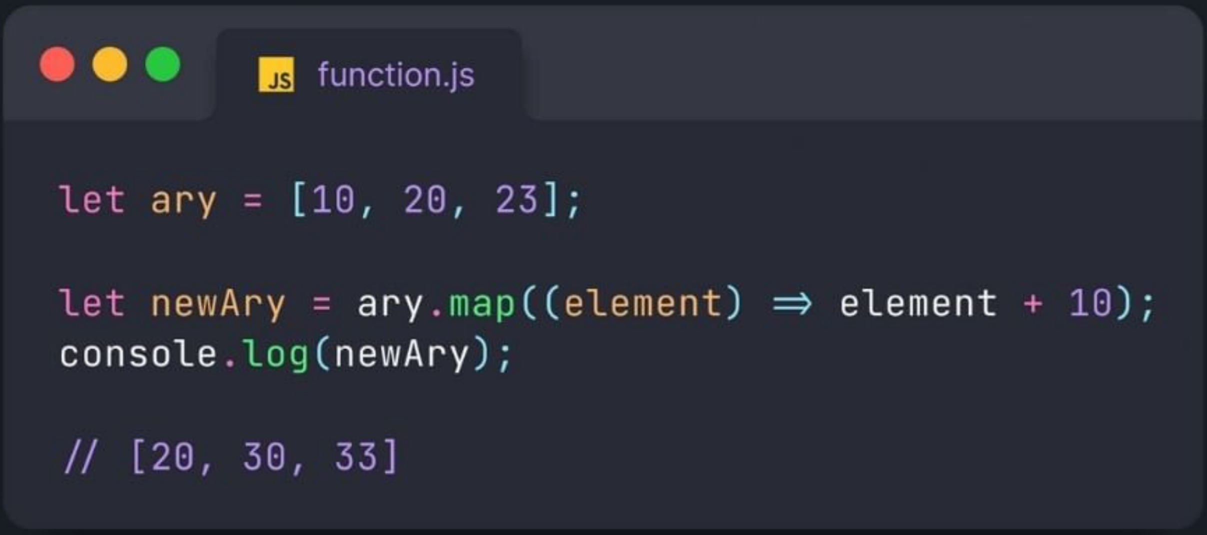
```
(function () {
  let str = "Good Evening";
  console.log(str);
})();

// Good Morning (Immediate Invoked)
```

@CodeBustler

5. Higher Order Func

Functions that take one or more **functions as arguments** or return a function.

eg : map(), filter(), reduce()



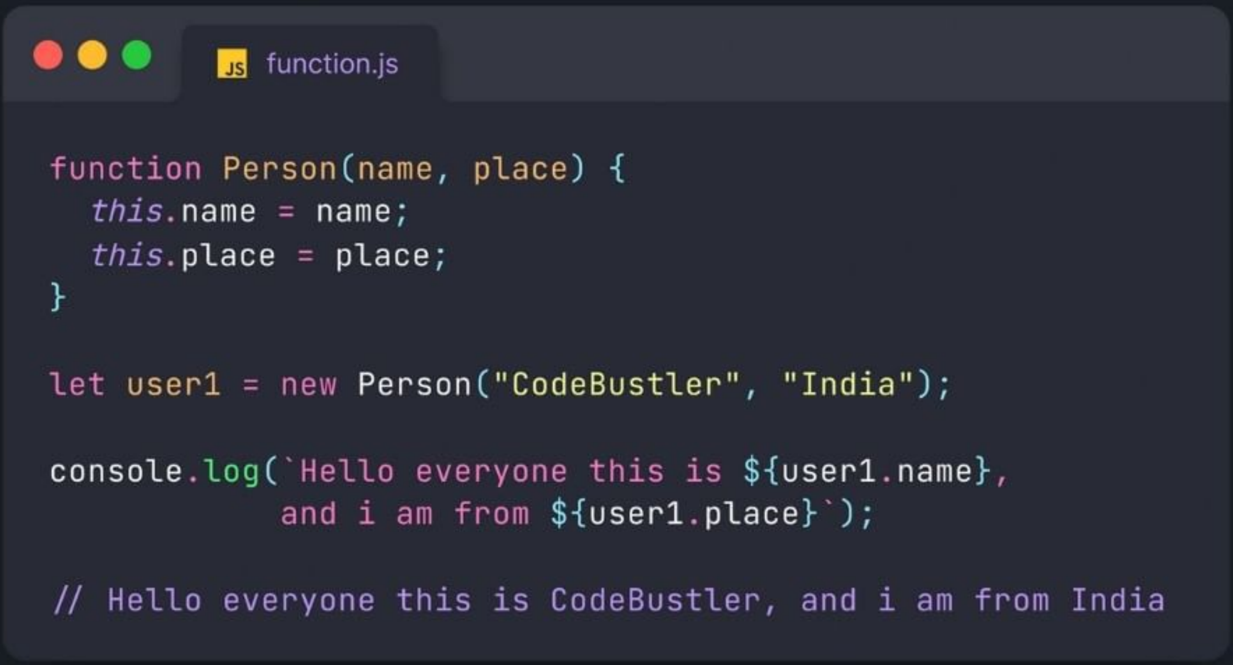
function.js

```javascript
let ary = [10, 20, 23];

let newAry = ary.map((element) => element + 10);
console.log(newAry);

// [20, 30, 33]
```

@CodeBustler

6. Constructor Function

Used as blueprints for **creating objects with similar properties** and methods. They are invoked using the **new** keyword to create instances of objects.



```
function Person(name, place) {
  this.name = name;
  this.place = place;
}

let user1 = new Person("CodeBustler", "India");

console.log(`Hello everyone this is ${user1.name},
            and i am from ${user1.place}`);

// Hello everyone this is CodeBustler, and i am from India
```

@CodeBustler