

18 'this' keyword

The **this** keyword is used inside a function. The **this** keyword is merely a reference to another object.

What the **this** keyword refers to depends on the scenario or the way the function is implemented. Here are the 3 scenarios to remember:

Scenario #1: **this** inside a function

The **this** keyword points to global object.

Scenario #2: **this** inside a method

The **this** keyword points to the object the method is in.

Scenario #3: When function is run with **call**, **bind** or **apply**

When a function is called using the **.call(param)**, **.bind(param)** or **.apply(param)** method, the first **param** become the object that the **this** keyword refers to.

Important Note:

In the browser, global is the **window** object.
In Node.js, global is the **global** object.

```
var name = "Fatema";
```

```
function fun(){  
  // some code here  
  console.log(this.name);  
}
```

```
const user = {  
  name: "Marium",  
  yearOfBirth: 1999,  
  calcAge: function(){  
    const currentYear = (new Date()).getFullYear();  
    return currentYear - this.yearOfBirth;  
  }  
}
```

```
fun(); // 'this' is global. Logs "Fatema"  
user.calcAge(); // 'this' is the user object  
fun.call(user); // 'this' is the user object. Logs "Marium"
```