



Most common **JavaScript**  
Interview Questions

# **This keyword** in **JavaScript**



One of JavaScript's most difficult topics, the **this** keyword, allows us to refer to the **object** that executes a **method**.

swipe for detailed examples



Swipe →



```
class Client {  
  constructor() {  
    this.clientId = '';  
  }  
  
  changeClientId(clientId) {  
    this.clientId = clientId;  
  }  
  
  getClientId() {  
    return this.clientId;  
  }  
  
  sayWelcome() {  
    console.log(`Welcome, ${this.getClientId()}`);  
  }  
}  
  
const newContract = new Client();  
newContract.changeClientId('577');  
newContract.sayWelcome();
```

reference the  
object itself

allows us to access data  
and methods on the object.





Its **value** is determined by **where the function is called** that uses this

Even after **defining** 'this' to work a certain way it can still **change** at any point in your program.

swipe for rules



## Rules to help determine the value of 'this' :

1. When you create an object using the new keyword with a constructor function/class, **this** will refer to the **new object** inside the function.
2. Using bind, call, or apply will **override** the value inside a function, and you can **hardcode** its value for **this**.
3. If a function is called on an object as a method, **this** will refer to the object that is calling it. For example, myObject.method() would have a **value** of **this** that **refers to myObject**.



Swipe →

4. If a function is executed without any of the three previous criteria being applied, **this** will refer to the **global object**, which is window in the browser or global in Node. If you are using strict mode, **this** will be **undefined** instead of the global object.
5. If **multiple** rules from above apply, it will use the rule that comes **first in this list**.
6. Arrow functions ignore all the above rules, and the value of **this** is determined by the **scope** enclosed by the arrow function.



# Do you find it helpful?

let me know down in the  
comments !



## Slobodan Gajić

Content Creator



FOLLOW FOR MORE