
this.

Because it ain't that.

this: A special keyword context variable

In JavaScript, there are 4 contexts that determine the value of this

1. global
2. method
3. dynamic
4. constructor

Additionally, there are 2 special cases

1. modules
 2. ES6
-

this: Global Context

```
function logThis() {  
    return this;  
}
```

```
logThis() === global  
true
```

this: Method Context

```
function logThis() {  
    return this;  
}  
var foo = {logThis: logThis};  
  
foo.logThis() === foo  
true
```

this: Dynamic Context

```
function logThis() {  
    return this;  
}  
var foo = {bar: logThis};  
  
logThis.call(foo) === foo  
true
```

this: Constructor Context

```
function logThis() {  
    return this;  
}
```

```
new logThis() instanceof logThis  
true
```

this: Node.js Module Context

```
// inside myModule.js
```

```
this === exports  
  true
```

this: ES6

```
"use strict";  
function logThis() {  
    return this;  
}
```

```
logThis() === undefined  
true
```
