

This keyword in Javascript

this is a keyword

Usage of this in global space

What is global object?

JS can run in browsers, smart watches, node JS, etc in many applications. Each application has a JRE (Javascript run time environment) which has defined global objects

Eg. In browser, global object is window, in nodeJS it is global.

* So when this is used in global space it uses global object value

Usage of this in function?

```
function x() {
```

```
  console.log(this);
```

```
}
```

```
x();
```

* Generally the value of this should be undefined, but this value is substituted with global object when the js code is using non-strict mode, if js is using strict mode, then it is undefined

The value of this also depends on how function is called

But it give reference like by calling window.x(), then this is defined as window.

Usage of this inside object

```
const obj = {
```

```
  a: 10,
```

```
  x: function() {  
    console.log(this);  
  }  
};
```

```
obj.x();
```

This refers to { a: 10, x: () }

Call, apply, bind methods:

call, apply, bind methods we can change the context of this

Eg:


```
student2 = {  
  name: 'Aju',  
  printName: function() {  
    console.log(this.name);  
  }  
}
```

```
student2 = {  
  name: 'Varghese'  
}
```

* What to do if we want to call printName function for student2 object.

student2.printName() X not possible

Through call we will share student1 function

student1.printName.call(student2)  Takes new this value
of
Varghese

* Check more on call, apply, bind methods

Using This inside arrow functions

Arrow functions can't provide their own this binding. (it retains this value of the enclosing lexical context).

Eg:

```
obj1 = {  
  a: 10  
  x: () => {  
    console.log(this);  
  }  
}
```

o/p
window b
because x is enclosed in
global space

It actually behaves as `console.log(this)` in global space

Eg2

```
const obj2 = {
```

```
  a: 10,
```

```
  x: function () {
```

```
    const y = () => {
```

```
      console.log(this)
```

```
    } y();
```

```
  }
```

```
obj2.x();
```

q0

```
{a: 10, x: fn() }
```

As, B Arrow function doesn't have its own value, it retains this value of its enclosed lexical context

This inside DOM

```
<button onClick = "alert(this)" ></button>
```

This represents the object HTML Button element.

```
<button onClick = "alert(this.tagName)" ></button>
```

O/p

BUTTON.