

# CLASS ASSIGNMENT SEPTEMBER 19

## CODE SNIPPET

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Understanding "this" in JavaScript</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      margin: 20px;

      display: flex;

      flex-direction: row;

    }

    button {

      margin: 10px;

    }

  </style>

</head>

<body>

  <div>

    <h1>Understanding "this" in JavaScript</h1>

    <h3>1. "this" inside an object method</h3>

    <button onclick="obj.showThis()">Click to see 'this' in an object method</button>

    <h3>2. "this" by itself</h3>

    <button onclick="showGlobalThis()">Click to see 'this' by itself</button>
```

<h3>3. "this" inside a regular function</h3>

<button onclick="regularFunction()">Click to see 'this' in a function</button>

<h3>4. "this" in a function under strict mode</h3>

<button onclick="strictModeFunction()">Click to see 'this' in strict mode</button>

<h3>5. "this" in an event handler</h3>

<button id="eventButton">Click me to see 'this' in an event handler</button>

<h3>6. Reassigning "this" using call(), apply(), and bind()</h3>

<button onclick="useCall()">Click to see 'this' with call()</button>

<button onclick="useApply()">Click to see 'this' with apply()</button>

<button onclick="useBind()">Click to see 'this' with bind()</button>

</div>

<div>



</div>

<script>

var globalObject = 'This is the global object';

// 1. this inside an object method

```
const obj = {  
  name: 'Inside of an object method',  
  showThis: function () {  
    console.log(this); // 'this' refers to the obj itself  
    alert(this.name);  
  }  
};
```

// 2. this by itself

```
function showGlobalThis() {  
  var globalObject = 'This is the local object';  
  
  alert(`This by itself refers to the global object which is: ${this.globalObject} but the local object is:  
  ${globalObject}`);
```

```
}
```

```
// 3. this inside a regular function
```

```
function regularFunction() {  
    var globalObject = 'This is the local object inside function';  
  
    function innerFunction() {  
        alert(`This inside a function refers to the global object which is: ${this.globalObject} but the local object is: ${globalObject}`);  
    }  
  
    innerFunction();  
}
```

```
// 4. this inside a function in strict mode
```

```
function strictModeFunction() {  
    'use strict';  
  
    function innerFunction() {  
        alert(`In strict mode, "this" is: ${this}`);  
    }  
  
    innerFunction();  
}
```

```
// 5. this inside an event handler
```

```
document.getElementById('eventButton').addEventListener('click', showEventThis);
```

```
function showEventThis() {  
    alert(`In an event, "this" refers to the element: ${this.id}`);  
}
```

```
// 6. Reassigning this using call(), apply(), and bind()
```

```
const usingCall = {  
    name: 'Object used for call()'   
};  
  
const usingApply = {  
    name: 'Object used for apply()'   
};  
  
const usingBind = {  
    name: 'Object used for bind()'   
};
```

```
};

function showThisReassigned() {
  alert(`"this" is reassigned to: ${this.name}`);
}

function useCall() {
  showThisReassigned.call(usingCall);
}

function useApply() {
  showThisReassigned.apply(usingApply);
}

function useBind() {
  const boundFunction = showThisReassigned.bind(usingBind);
  boundFunction();
}
</script>
</body>

</html>
```

**GitHub:** <https://github.com/DMegatron/Full-Stack-Assignment-September-19>