Understanding 'this' in JavaScript



Paul D. Sheriff
BUSINESS/TECHNOLOGY CONSULTANT
paul.d.sheriff@gmail.com



Module Goals



What is 'this'

Learn how 'this' changes

Global and function scope

Event handlers

Object literal

call()/apply() methods

Constructor functions



Refers to an object

That object in which the current code is running

Sometimes the object can be changed



```
<script>
console.log(this.toString());
// prints [object Window]
<script>
this = global
window object
```



```
person {
  firstName: "John",
  lastName = "Smith"
  fullName: function () {
    return this.firstName + " " + this.lastName;
  }
}
this = person
  object literal
```

```
function Person(first, last) {
 this.firstName = first;
 this.lastName = last;
 this.fullName = function () {
  return this.firstName + " " + this.lastName;
let p1 = new Person("John", "Smith");
let p2 = new Person("Bob", "Small");
```

this = current Person object (p1 or p2)

this Keyword

Different value based on execution context

In a method: owner object

In a function: global object

In an event:
element that
received the event

call()/apply methods refers to object passed in

'use strict' also affects 'this'





Global scope

Function scope

'use strict'





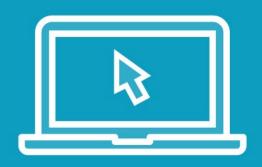
Event handlers





Object literal





call() method
apply() method





Constructor functions



Summary



Scope determines value of 'this'

- Global object
- HTML element
- Method owner

'use strict' makes 'this' undefined in functions

What is passed to call()/apply() methods becomes 'this'

constructor functions owner is 'this'





Coming up in the next module...

Use the spread operator Manipulating arrays
Pass arrays to function

