

# Functions



bind  
bindAll  
partial  
memoize  
delay  
defer

throttle  
debounce  
once  
after  
wrap  
compose

# JavaScript this



Mozilla Developer Network: this

<https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Operators/this>

# JavaScript this

**window** ←

```
<script>
```

```
function hello(){
```

```
  this. _____
```

```
}
```

```
hello();
```

```
</script>
```



# JavaScript this

## window

```
<script>
```

```
function hello(){  
  this.  
}
```

```
$('#button-save').click(hello);
```

```
<script>
```

```
...
```

```
<button id='#button-save'>Save</button>
```

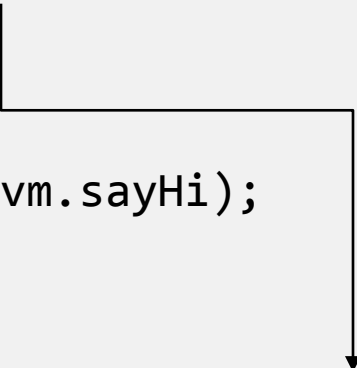


# Without Binding

## window

```
<script>
  var vm = {
    name: 'Craig',
    sayHi: function(){
      alert('Hello, ' + this.name);
    }
  };

  $('#button-save').click(vm.sayHi);
</script>
...
<button id='#button-save'>Save</button>
```

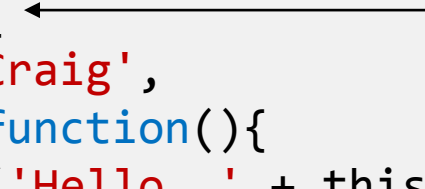
A diagram consisting of a horizontal line that starts from the right side of the 'this.name' property access in the function call 'vm.sayHi()'. This line extends to the right and then turns downwards as an arrow, pointing to the 'id' attribute of the '<button id='#button-save'>' HTML element. This illustrates that in the absence of a binding mechanism, the 'this' context inside the function is the global window object, which is the parent of the button element in the DOM.

# With Binding


## window

```
<script>
  var vm = {
    name: 'Craig',
    sayHi: function(){
      alert('Hello, ' + this.name);
    }
  };

  _.bindAll(vm);
  $('#button-save').click(vm.sayHi);
</script>
...
<button id='#button-save'>Save</button>
```



A diagram consisting of a horizontal line with a downward-pointing arrow at its right end, connecting the 'vm' variable in the first line of the code block to the 'this' keyword in the 'sayHi' function.



An orange arrow pointing to the right, highlighting the `_.bindAll(vm);` line in the code block.

# Memoization



Memoization is derived from the Latin word memorandum (to be remembered), and thus carries the meaning of turning [the results of] a function into something to be remembered.



**Source:** <http://en.wikipedia.org/wiki/Memoization>

# Memoization

First execution





# Memoization

Second execution



# Summary



bind  
bindAll  
partial  
m  
de  
de

throttle  
debounce  
once



Next Up....

Objects

