

Xtern

# BOOTCAMP

**Week 1 | Day 4**

**June 8, 2017**

Review

Objects

Creating an object

```
const things = {}
```

# Adding properties

```
const things = {  
  country: 'USA'  
}
```

```
things.animals = ['dog', 'rhino']
```

# Functions as Properties

```
things.explode = function() {  
    alert('KABOOM')  
}
```

# Accessing Properties

```
things.animals  
things['KABOOM']
```

# Functions



# Function expressions

```
function sum(a, b) {  
    return a + b  
}
```

# Functions as variables

```
const sum = function(a, b) {  
  return a + b  
}
```

```
const sum = (a, b) => {  
  return a + b  
}
```

# Function invocation

```
const sum = function(a, b) {  
  // this === ?  
  return a + b;  
};  
sum(2, 3);
```

# Function invocation

```
const sum = function(a, b) {  
  // this === window  
  return a + b;  
};  
sum(2, 3);
```

# Function invocation

```
const sum = (a, b) => {  
  // this === ?  
  
  return a + b;  
};  
sum(2, 3);
```

# Function invocation

```
const sum = (a, b) => {  
  // this === whatever `this` is in  
  // the surrounding function  
  return a + b;  
};  
sum(2, 3);
```

# Event handler invocation

```
function handleClick(ev) {  
    // this === ev.target  
}
```

```
btn.addEventListener('click', handleClick)
```

# Event handler invocation

```
function handleClick(ev) {  
    // this === ?  
}
```

```
btn.addEventListener('click', handleClick)
```



Event handler invocation  
(with an anonymous function)

```
btn.addEventListener('click', function(ev) {  
    // this === ?  
})
```

# Event handler invocation

(anonymous function makes no difference)

```
btn.addEventListener('click', function(ev) {  
    // this === ev.target  
})
```

Event handler invocation  
(with an arrow function)

```
btn.addEventListener('click', (ev) => {  
  // this === ?  
})
```

Event handler invocation  
(with an arrow function)

```
btn.addEventListener('click', (ev) => {  
  // this === whatever `this` is in the  
  // surrounding function  
})
```

# Methods

(functions as object properties)

```
const thing = {  
  explode() {  
    alert('KABOOM')  
  }  
}
```

# Method Invocation

```
const thing = {  
  explode() {  
    // this === ?  
    alert('KABOOM')  
  }  
};  
thing.explode()
```

# Method Invocation

```
const thing = {  
  explode() {  
    // this === thing  
    alert('KABOOM')  
  }  
};  
thing.explode()
```