FEWD

FINAL PROJECTS

FINISH UP RESPONSIVENESS

AGENDA

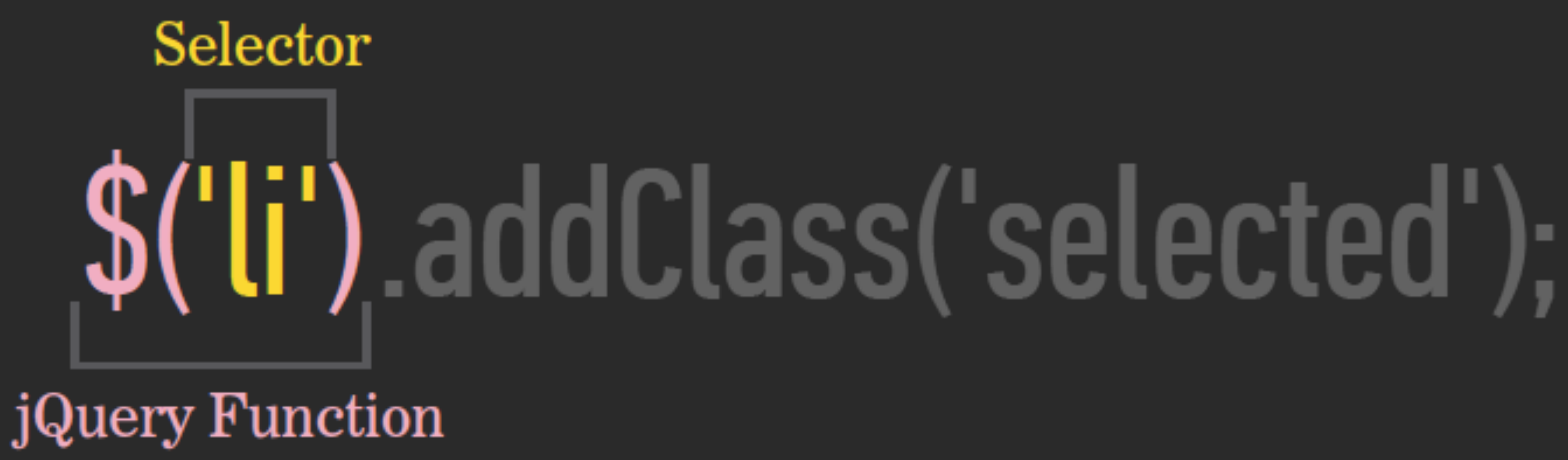
- Review
- Objects
- Lab —Work on final project

LEARNING OBJECTIVES

- Describe the concept of "this" as it applies within jQuery functions.
- Know how to make javascript objects and then to use them

QUICK JS REVIEW

JQUERY — SELECTING ELEMENTS



jQuery Function:

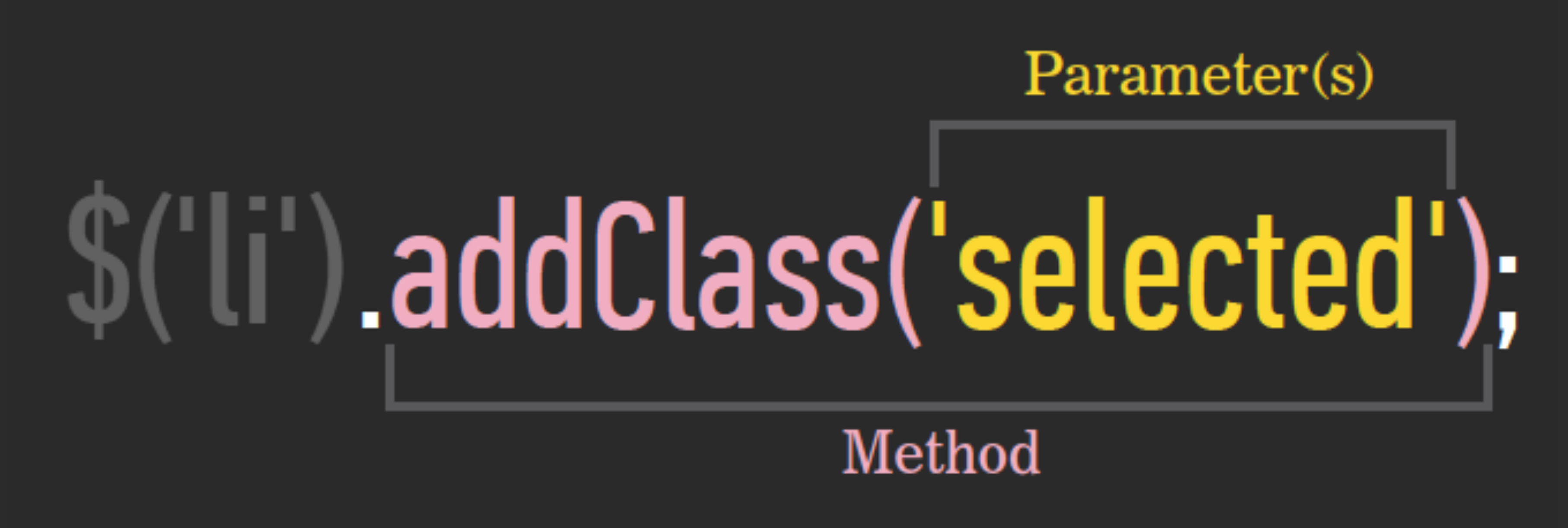
- Lets us find one or more elements in the page
- Creates a jQuery object which holds references to those elements

JQUERY OBJECTS — FINDING ELEMENTS: SOME EXAMPLES

- You can use your CSS-style selectors!!!

SELECTOR:	CSS:	JQUERY:
CLASS	.className	$('.className')
ID	#idName	$('#idName')
MULTIPLE SELECTORS	h1, h2, h3	$('h1, h2, h3')
DESCENDANT	li a	$('li a')

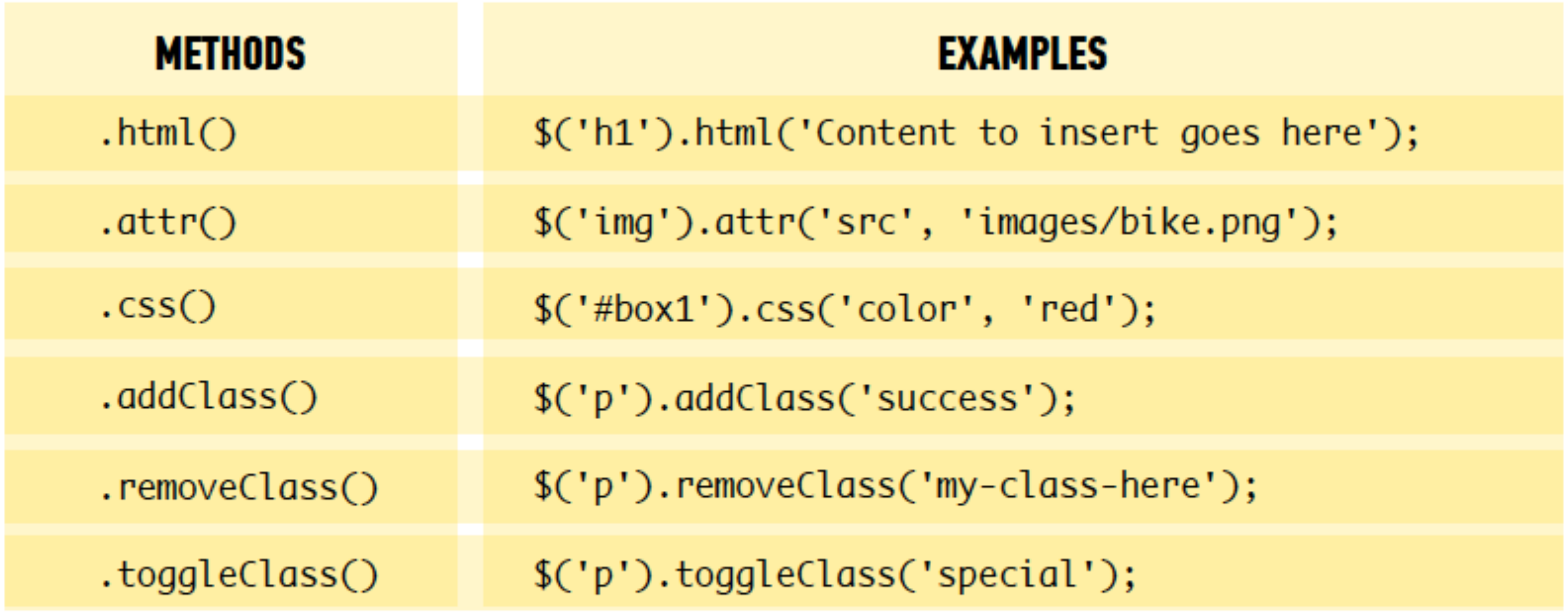
JQUERY — WORKING WITH THOSE ELEMENTS



```
$('li').addClass('selected');
```

JQUERY METHODS — GETTING/SETTING CONTENT

Get/change content of elements, attributes, text nodes



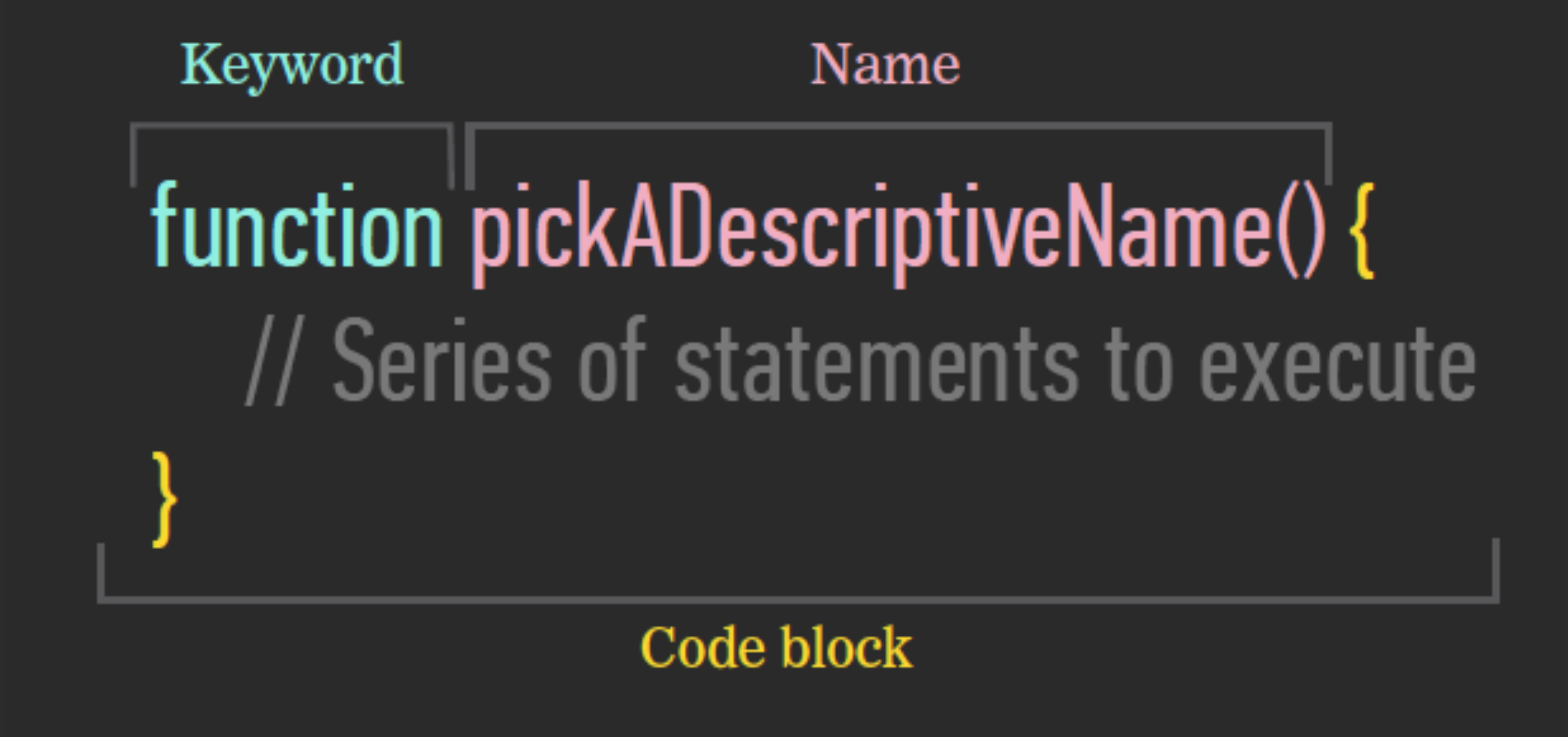
METHODS	EXAMPLES
`.html()`	`$('h1').html('Content to insert goes here');`
`.attr()`	`$('img').attr('src', 'images/bike.png');`
`.css()`	`$('#box1').css('color', 'red');`
`.addClass()`	`$('p').addClass('success');`
`.removeClass()`	`$('p').removeClass('my-class-here');`
`.toggleClass()`	`$('p').toggleClass('special');`

JQUERY METHODS — EFFECTS/ANIMATION

Add effects and animation to parts of the page

METHODS	EXAMPLES
`.show()`	`$('h1').show();`
`.hide()`	`$('ul').hide();`
`.fadeIn()`	`$('h1').fadeIn(300);`
`.fadeOut()`	`$('.special').fadeOut('fast');`
`.slideUp()`	`$('div').slideUp();`
`.slideDown()`	`$('#box1').slideDown('slow');`
`.slideToggle()`	`$('p').slideToggle(300);`

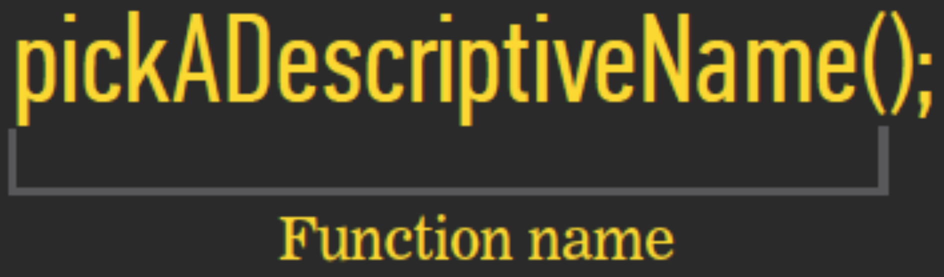
SYNTAX — DECLARING A FUNCTION



```
function pickADescriptiveName() {
    // Series of statements to execute
}
```

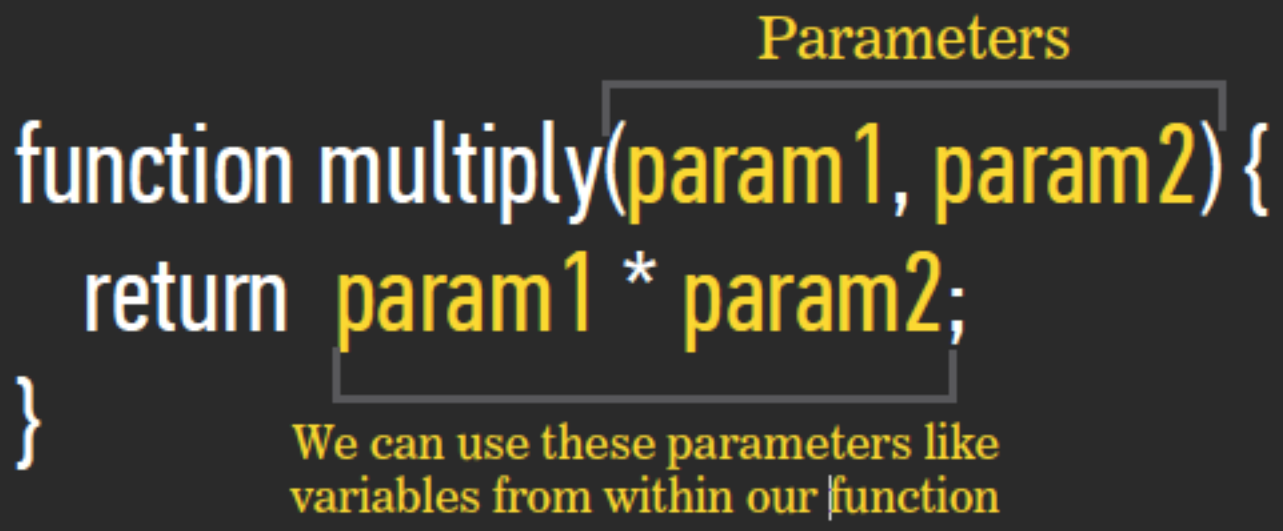
SYNTAX — CALLING A FUNCTION

- To run the code in a function, we 'call' the function by using the function name followed by parenthesis.



```
pickADescriptiveName();
```

SYNTAX — DECLARING A FUNCTION (WITH PARAMETERS)

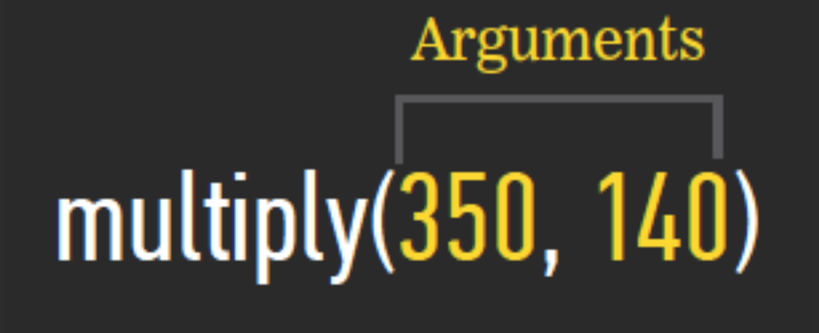


Parameters

```
function multiply(param1, param2) {
   return  param1 * param2;
}
```

We can use these parameters like variables from within our function

SYNTAX — CALLING A FUNCTION (WITH ARGUMENTS)



```
multiply(350, 140)
```

STORING LISTS OF VALUES

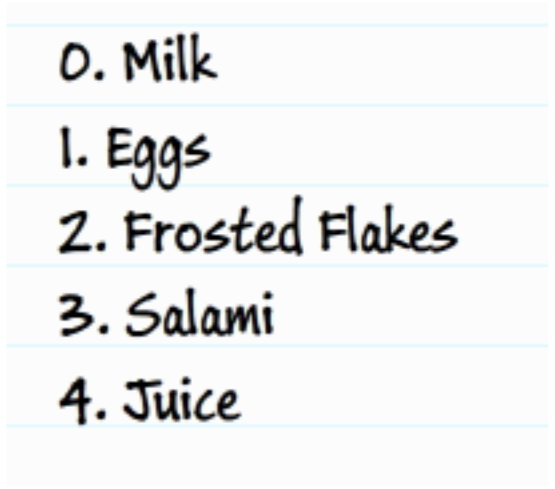
- An array can be used to **store a list of values in a single variable**
- Holds an ordered collection of values
- Can hold numbers, strings, even other arrays!
- Good for things like a grocery list, a list of states, or any other list

DECLARING ARRAYS

```
var descriptiveNameHere = [item1, item2, item3];
```

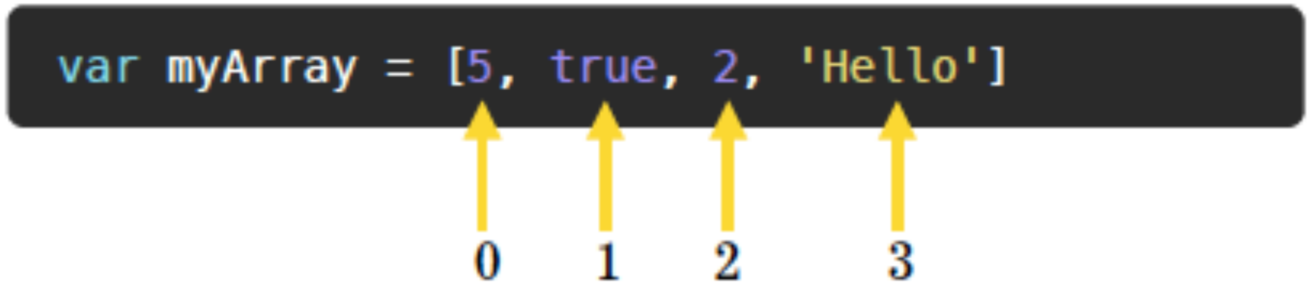
ARRAYS - INDEXING

- Each item in an array has an **index**, by which you can access that item.
- The first item has an index of **0**, the second item 1, the third item 2, etc.


0. Milk
1. Eggs
2. Frosted Flakes
3. Salami
4. Juice

ARRAYS - ACCESSING ITEMS BY INDEX

- Each item in an array has an **index**, by which you can access that item.
- The first item has an index of **0**, the second item 1, the third item 2, etc.



```
var myArray = [5, true, 2, 'Hello']
```

ARRAYS — ACCESSING ITEMS IN AN ARRAY

Accessing items in array:

`myArray[1]` → true

`myArray[2]` → 2

`myArray[0]` → 5

`myArray[3]` → 'Hello'

```
var myArray = [5, true, 2, 'Hello']
```

ARRAYS - ADDING A VALUE/REPLACING A VALUE

INSERTING A NEW VALUE

- We can insert new values into any space in the array using the positions index.

```
myArray[1] = 'Hello';
```

UPDATING VALUES

- If there's already an item at that position, it will be replaced with the new value.

```
var myArr = [65, 'hello', true];
myArr[1] = 'goodbye';
// myArr[1] now holds 'goodbye' instead of 'hello'
```

ARRAYS - LENGTH

- We can use the .length property to find out how many items are in an array

```
var shapes = ['circle', 'triangle', 'square'];
```

```
shapes.length;
```

=> 3

- Accessing the last element in an array:

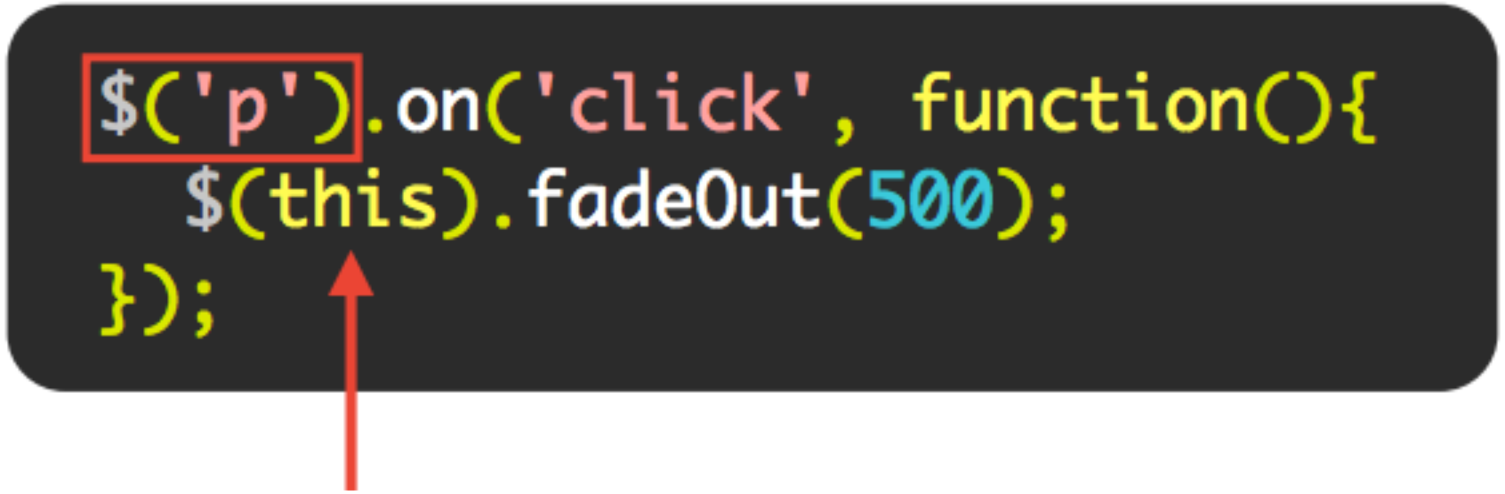
```
console.log(shapes[shapes.length-1]);
```

=> Prints 'square'

‘THIS’ KEYWORD

THE KEYWORD 'THIS'

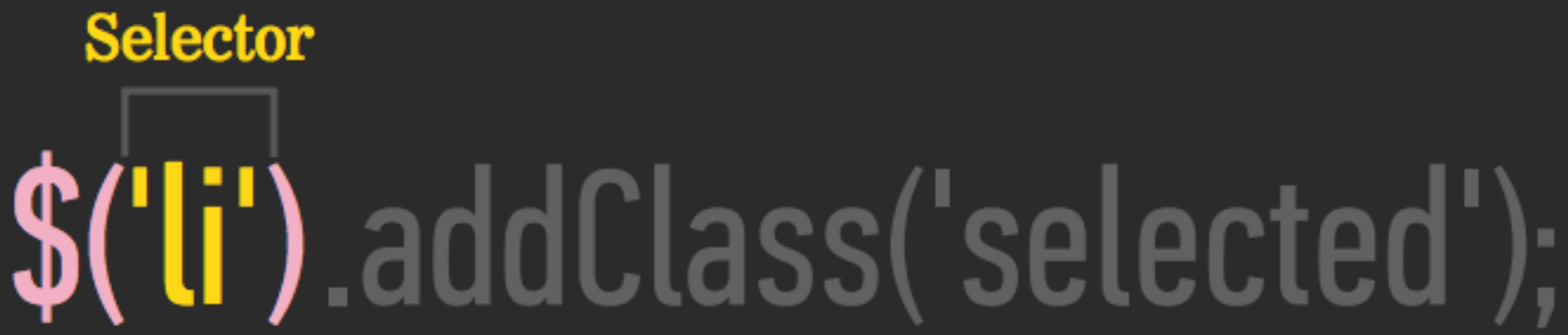
this refers to whatever you *selected* with jQuery



```
$('p').on('click', function(){
  $(this).fadeOut(500);
});
```

Notice — no quotes around this!

JQUERY — SELECTING ELEMENTS



```
$('li').addClass('selected');
```

OBJECTS

OBJECTS

- OOP- Object oriented Programming
- Lets us write reusable code to keep track of data

Everything is an object!

BULLDOG AS AN OBJECT

Objects have traits that are common to versions of itself
These traits are called properties in javascript

Bulldog Properties

- Legs - 4
- Sound - “Bark”
- Food - “Dog Food"

DECLARING OBJECTS

```
function myObject(){

};
```

ASSIGNING PROPERTIES

```
function myObject(){
   this.property = value;
};
```

MAKING NEW OBJECTS

```
var newObject = new object();
```

GETTING OBJECT VALUES

```
var newObject = new object();

newObject.propertyName;
```

HOMEWORK

FINISH RESPONSIVENESS THIS WEEKEND

EXIT TICKETS