

# LESSON 11

## Practical JavaScript





# CODEALONG REVIEW

1- Create a dropdown menu with three national park options (acadia, banff, zion).

2- If user chooses one option, display corresponding image and a message.

3- Let's make this page actually look good with some custom fonts and thoughtful use of color.

# CODEALONG REVIEW

Now lets make the dropdown into a multi-select element. What do we have to change about our program?





**MAKING COOL STUFF**



Check Out

# **ASSIGNMENTS #2-6**

While the Lecture is Going On

# WHY JQUERY PLUGINS?

Many developers have hit the same walls as you - stand on the shoulders of giants instead of rebuilding components you don't have to.

# + 'S AND - 'S

- + You will create awesome stuff faster
- + This is how real developers get websites done so quickly
- + Open-source ethos is a great thing
- You won't learn as much doing this
- Long term plugin reliability is always a question vs doing it yourself

# WHAT TO LOOK FOR

- Make sure your plugin has some Github stars, usually a good sign other devs have found it useful
- See how many versions have been published - look for at least 3-4 so the bugs are gone



# HOW TO PARTICIPATE

- READ THE DOCS CAREFULLY!
- If you have issues 95% of the time it's you. Sorry but it's true.
- If you still have questions, read the docs again!
- Ok, now make an issue for the dev if you really really really think you've found a bug
- Better yet, give them a pull request that fixes the problem you've found.

# HOW PLUGINS WORK

Typically they define a new function and use JSON to include options inside of that new function:

```
$( '.selector' ).pluginName ( {  
    // options  
    option1: 123,  
    option2: 'xyz',  
    option3: true  
} );
```



# FLICKITY

Physics-based carousel, smooth on mobile:

<https://github.com/metafizzy/flickity>

# FLICKITY USAGE

```
// Flickity attaches to jQuery selector  
just like any method
```

```
// Notice how the options supplied to  
flickity look just like .css() - using  
JSON notation
```

```
$('.carousel-container').flickity({  
  // options  
  autoPlay: 2000,  
  cellAlign: 'left',  
  contain: true,  
  setGallerySize: false,  
  wrapAround: true  
});
```



# SCROLLTO

Moves user around the page fluidly,  
used often for those 'back to top'  
buttons

[https://github.com/flesler/  
jquery.scrollTo](https://github.com/flesler/jquery.scrollTo)

# SCROLLTO() USAGE

```
// scrollTo requires two parameters  
// 1- the scroll target, in this case ID2  
// 2- duration of the scrollTo action (ms)
```

```
$('#ID1').click(function() {  
    $.scrollTo('#ID2', 600);  
});
```



# WAYPOINTS

Detect scroll points on a page as a user moves through it

[https://github.com/  
imakewebthings/waypoints](https://github.com/imakewebthings/waypoints)

# WAYPOINTS USAGE

// You don't have to use the conditional directions but I use them every time with waypoints, so you probably will too

```
var firstWaypoint = $('#ID-selector').waypoint({  
  handler: function(direction) {  
    if (direction === 'down') {  
      // do something  
    }  
    if (direction === 'up') {  
      // do something  
    }  
  }  
});
```



# CODE ALONG

Look at the Plug-in Assignments #2-5

**YOU DO IT**

Make Assignment #6 Work

# START ON HW 6

Work together on this assignment



# NEXT TIME

Finish HW 5 - Due Mon 12/19

Wireframes for Project - Due Wed 12/21

Lesson 12

this and Function returns

Arrays and Loops

HTML Forms