Learning Objectives:
- Understand the difference between prompt-based programming and event-based programming.
- Learn the HTML syntax for creating widgets on the screen (text boxes, drop-downs, buttons, etc.)
- Be introduced to the concept of a "library" like jQuery, and how it can be included in your code just like your own .js file.
- Learn the jQuery code to create an event-handler, wire it up to a button, and extract user data from widgets.
I/O: the dorky way

Input via prompt() boxes and output via alert() boxes is a clumsy and non-optimal way to interact with the user. Not only is it ugly, but it disrupts the user's workflow and takes control away from them, where it belongs.

I/O: the cool way

A better approach is to create on-page widgets that the user can interact with, and then write output to a chosen location on the screen as needed. This allows the user to interact with the page as the application, control their own flow, and not be pestered by pesky dialog boxes.

The basic kinds of widgets are:
- text boxes
- checkboxes (used for non-mutually-exclusive alternatives)
- radio buttons (used for mutually-exclusive alternatives)
- drop-down lists (used for mutually-exclusive alternatives)
- buttons
Creating on-screen widgets

A textbox:

```html
<input type=text size=10 id="lastName" />
```

A checkbox:

```html
<input type=checkbox id="graduated" />
```

A radio button:

```html
<input type=radio name="gender" id="female" />
```

A push button:

```html
<button id="makePurchase">Buy it!</button>
```

A drop-down list:

```html
<select id="iceCreamFlavor">
  <option value="choc">Chocolate</option>
  <option value="van">Vanilla</option>
  <option value="straw">Strawberry</option>
  <option value="gar">Garlic</option>
</select>
```
An "empty div"

<div> is an HTML element that does nothing by itself. It's a grouping mechanism that makes organizing and updating your page easier. By creating an empty <div> with an id, you can set aside a place for output text to appear.

<div id="outputGoesHere" />
jQuery: a JavaScript "library"

A library is basically a repository of useful functions that someone has made available for use. jQuery contains JavaScript functions that we will use.

It's open source and freely available online here:

http://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jquery.min.js

All you have to do is include this as the "src" attribute of a <script> tag, and voila!

jQuery allows us to access the elements on a page, find out what values the user has entered, and change them if necessary.
To find out what value is in a textbox or dropdown:

\[
\text{\$\("#idOfTextboxOrDropDown\"\).val()}
\]

To find out whether a radio button or checkbox is checked:

\[
\text{\$\("#idOfCheckboxOrRadioButton\"\).attr\("checked"\)}
\]

(This is useful with an "if" statement.)
Output

To write text to an empty div:

```javascript
$('idOfEmptyDiv').text('some text');
```

To append text to a (possibly non-empty) div:

```javascript
$('idOfDiv').append('some text');
```
Connecting event handlers

When we want a button to do something, we wire it to an "event handler." An event handler is just a function that you write!

To connect the button to your event handler:

```
$("#idOfButton").click(nameOfFunction);
```
Wiring it all together

Finally, you need to get JavaScript to call your code that sets up your event handlers.

Make a function called "init()" that has your event handler connection code. Then:

$$\texttt{\$(document).ready(init);}$$

at the bottom of your JavaScript file. That's it!