Day 4: Build Your Own Web Inspector

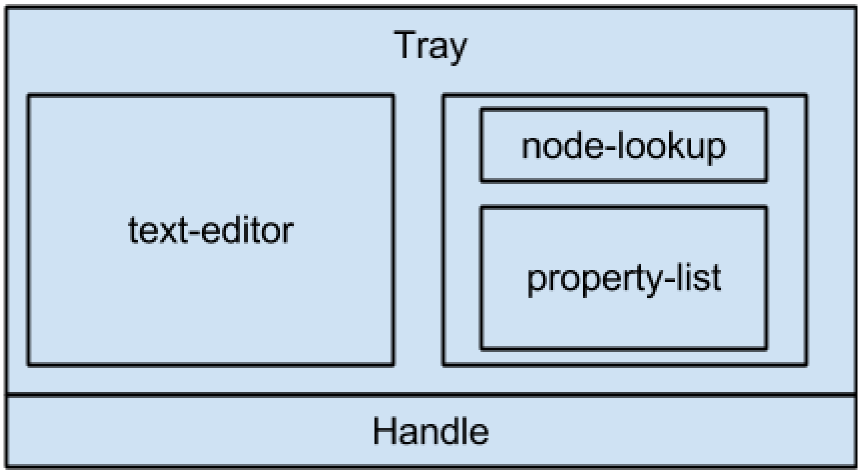
In this exercise you'll work on building an inspector similar to the one provided by web browsers. To get started, copy these three files into your project:

https://github.com/eob/6.mitx

0. [morning] Modify the Widget so that it adds the following HTML structure to its container

```
<div class='tray'>
  <textarea class='text-editor'></textarea>
    <div class='property-editor'>
      <div class='node-lookup'>
        <input class="selector" /><input class='nth' />
        <button>Lookup</button>
      </div>
      <div class='property-list'>
     </div>
   </div>
 </div>
 <div class="handle"></div>
```

The CSS is already provided so that it will have the following structure:



1. [morning] Animate the inspector sliding in and out when you click the tray

- Inspect the CSS file to see how tall the inspector is
- jQuery methods to keep in mind: **animate, bind**
- CSS properties to keep in mind: **top** (remember negative pixel values!)

Make sure you don’t move the inspector so far off the screen that it dissappears!

2. [morning] Search for the provided selector and show it’s html in the text-editor

- jQuery methods to keep in mind: **bind, html, val, eq**

3. Make the HTML editor live!

Listen for keyboard events on the text editor and *change* the HTML of the node it represents in response to those events.

4. Add property displays to the property view

Show us some useful properties about the element.

Be creative here -- what is useful data? Are there cool ways to show it (e.g.: showing a colored box for a color, showing a diagram of aspect ratio).

At a minimum, show these properties:

- Size (height, width)
- Position (Top, Left)
- Spacing (margin, padding)
- Background color, foreground color
- Tag name
- Number of children

5. Add a visual selection method using mouse events

- Add a button at the top of the property editor.
- When you click the button, register an event handler on mouse events on all elements. You will want to explore this page:

http://api.jquery.com/category/events/mouse-events/

- When you mouse over an element, give it a thick, red border.
- When you are no longer over an element, remove the border.
- When you click on an element, suppress any default actions of the browser and load it into the HTML view in the editor
- Once the user has clicked on an element, stop highlighting when you move the mouse around

- jQuery methods to keep in mind: **bind, unbind**
- Events to consider: mouseenter, mouseleave