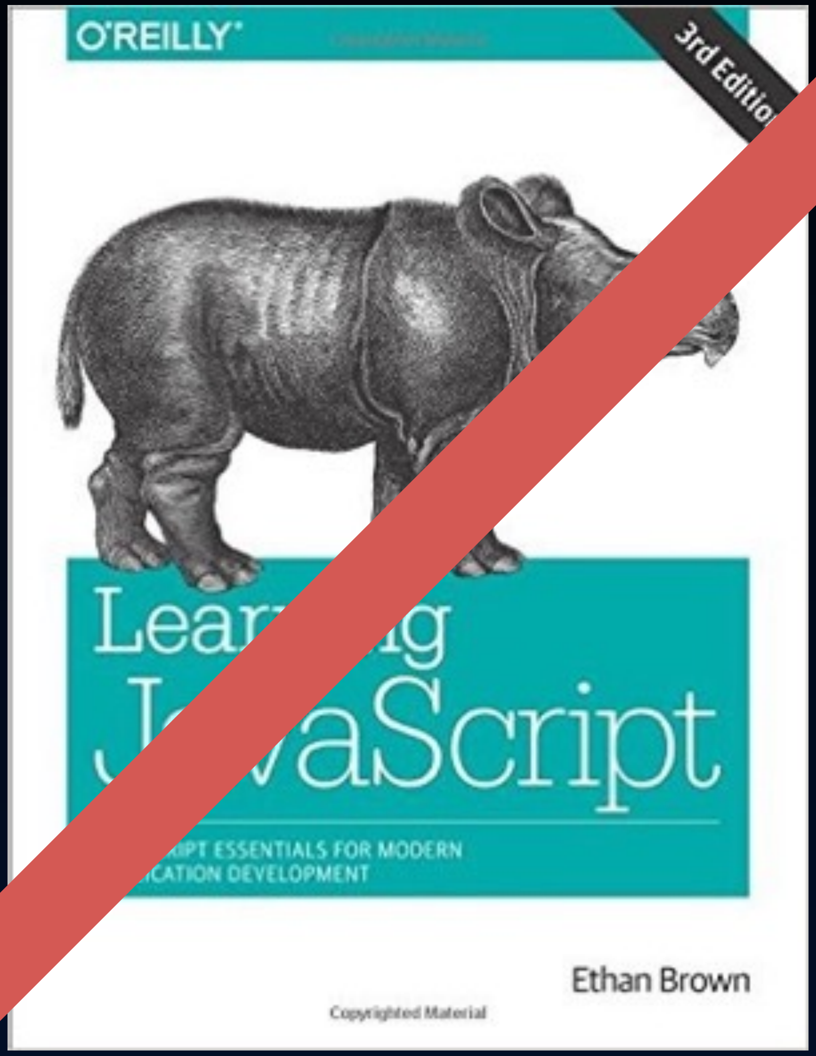
CS50C Web Dev 3

Fall 2017 ~ Ethan Wilde

Week 6

SANTA ROSA
JUNIOR COLLEGE

No Textbook Reading This Week



Learning JavaScript
Ethan Brown
ISBN 978-1-491-91491-5

Reading Online at jQuery.com

1. Plugins
   https://learn.jquery.com/plugins/

2. Finding & Evaluating Plugins
   https://learn.jquery.com/plugins/finding-evaluating-plugins/

3. How to Create a Basic Plugin
   https://learn.jquery.com/plugins/basic-plugin-creation/

jQuery Plugins

1. Creating a jQuery Plugin
2. jQuery Plugin Basics: Finding + Using Other People’s Plugins

Creating a jQuery Plugin

1. **Adding a method to the prototype jQuery $() function.**

```
$.fn.flipify = function() {
    this.css( "transform", "rotate(180deg)" );
    return this;
}
```

Creating a jQuery Plugin

1. **Adding a method to the prototype jQuery $() function.**

```
$.fn.flipify = function() {
    this.css( "transform", "rotate(180deg)" );
    return this;
}
```

$.fn

***$.fn** provides access to the prototype constructor for the jQuery $() function.*

Creating a jQuery Plugin

2. **Call the new method of the jQuery $() function.**

```
$.fn.flipify = function() {
    this.css( "transform", "rotate(180deg)" );
    return this;
}

$("#mything").flipify();
```

jQuery Plugin Basics

1. **Finding Plugins to Use**
2. **Steps to Use a Plugin**
3. **The Way Most Plugins Works**
4. **Before + After of HTML Modified by a Plugin**
5. **Categories of Plugins by Functionality**
6. **Examples**

http://plugins.jquery.com/

jQuery Plugin Basics

1. **Finding Plugins to Use**
2. **Steps to Use a Plugin**
3. **The Way Most Plugins Works**
4. **Before + After of HTML Modified by a Plugin**
5. **Categories of Plugins by Functionality**
6. **Examples**

Finding Plugins to Use

URL	Description
jQuery Plugin Registry (old)	http://plugins.jquery.com/
Peer-to-peer recommendations (Stack Overflow)	http://stackoverflow.com/
Examples and demonstrations	http://codepen.io/ https://jsfiddle.net/
Reviews and articles	http://jqueryrain.com/

Search for "jQuery FUNCTIONALITY"
where you substitute anything for word 2.

jQuery Plugin Basics

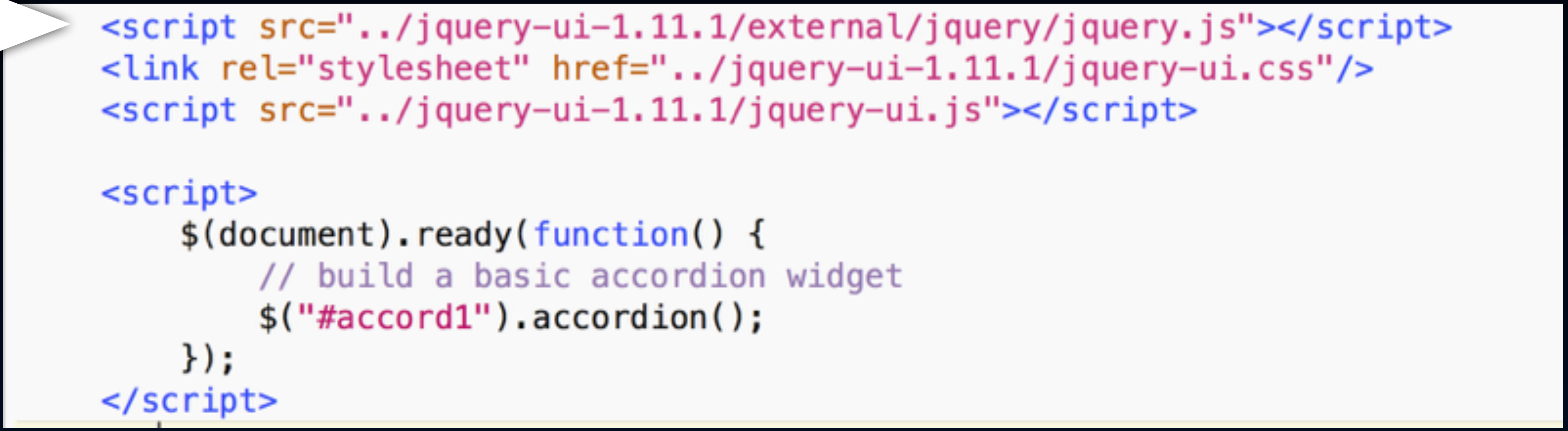
1. Finding Plugins to Use
2. Steps to Use a Plugin
3. The Way Most Plugins Works
4. Before + After of HTML Modified by a Plugin
5. Categories of Plugins by Functionality
6. Examples

Steps to Use a Plugin

1. **Load jQuery library (local or remote-hosted via CDN).**
2. **Load all JS and CSS files required by the jQuery plugin you are using.**
3. **Include HTML elements in correct pattern plugin is expecting.**
4. **Make sure the HTML elements in the body have loaded before your script runs.**
    1. **Use $(document).ready(function() {}) to wrap your script.**
    2. **Or place <script></script> element immediately before </body> closing tag.**

Make sure you are loading the jQuery library file first.

Steps to Use a Plugin

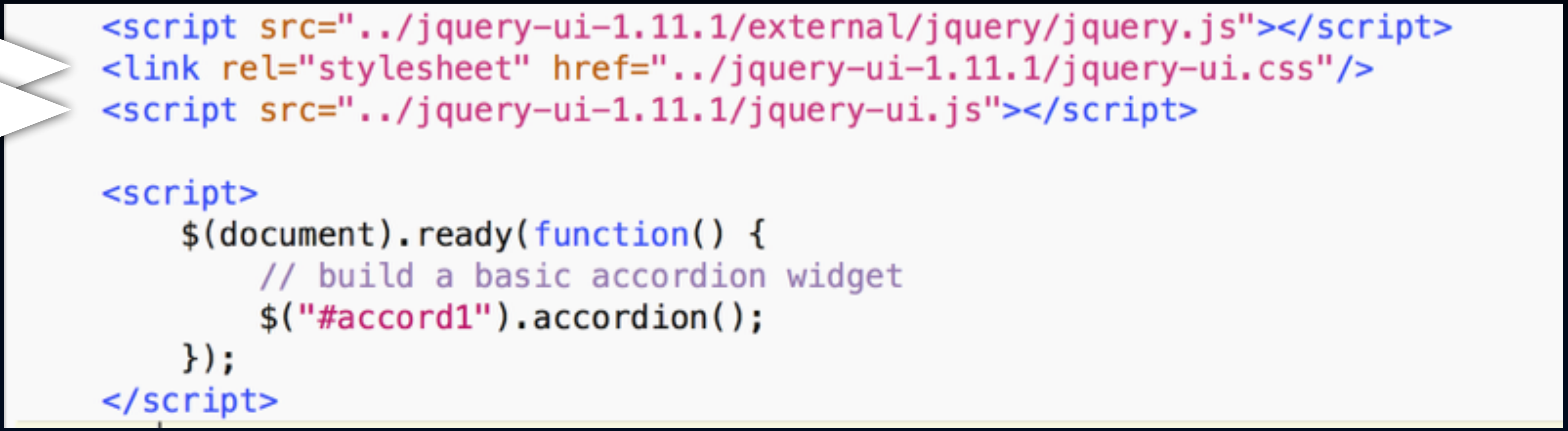


```
<script src="../jquery-ui-1.11.1/external/jquery/jquery.js"></script>
<link rel="stylesheet" href="../jquery-ui-1.11.1/jquery-ui.css"/>
<script src="../jquery-ui-1.11.1/jquery-ui.js"></script>

<script>
    $(document).ready(function() {
        // build a basic accordion widget
        $("#accord1").accordion();
    });
</script>
```

1. Load jQuery Library

Steps to Use a Plugin

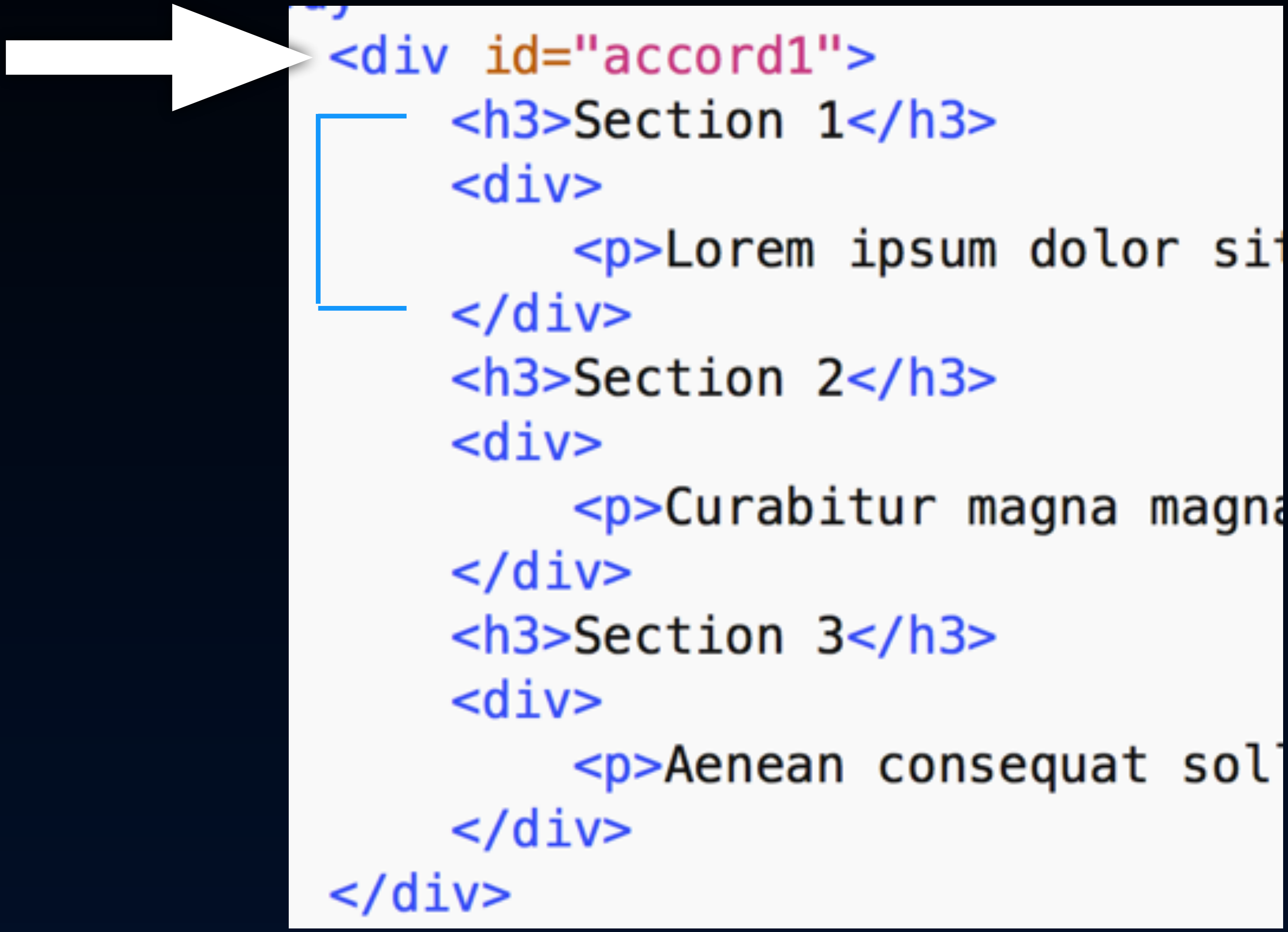


```
<script src="../jquery-ui-1.11.1/external/jquery/jquery.js"></script>
<link rel="stylesheet" href="../jquery-ui-1.11.1/jquery-ui.css"/>
<script src="../jquery-ui-1.11.1/jquery-ui.js"></script>

<script>
    $(document).ready(function() {
        // build a basic accordion widget
        $("#accord1").accordion();
    });
</script>
```

2. Load all CSS and JS files for jQuery plugin.

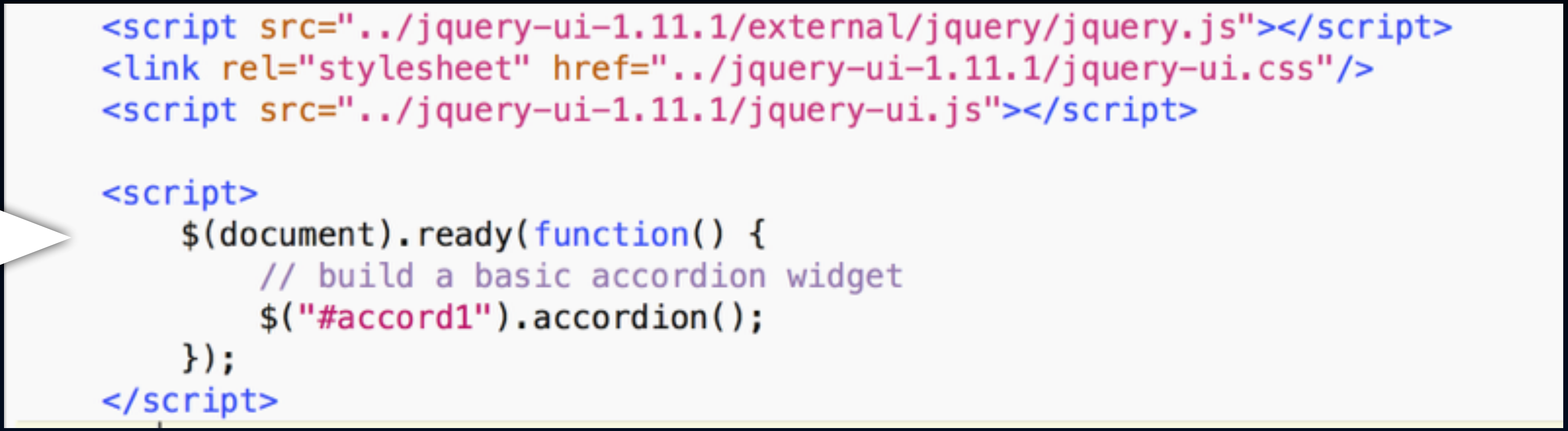
Steps to Use a Plugin



```
<div id="accord1">
    <h3>Section 1</h3>
    <div>
        <p>Lorem ipsum dolor si
    </div>
    <h3>Section 2</h3>
    <div>
        <p>Curabitur magna magn
    </div>
    <h3>Section 3</h3>
    <div>
        <p>Aenean consequat sol
    </div>
</div>
```

3. Make sure you include HTML elements following pattern plugin is expecting.

Steps to Use a Plugin



```
<script src="../jquery-ui-1.11.1/external/jquery/jquery.js"></script>
<link rel="stylesheet" href="../jquery-ui-1.11.1/jquery-ui.css"/>
<script src="../jquery-ui-1.11.1/jquery-ui.js"></script>

<script>
    $(document).ready(function() {
        // build a basic accordion widget
        $("#accord1").accordion();
    });
</script>
```

4. Make sure HTML in body is loaded before your script runs.

jQuery Plugin Basics

1. **Finding Plugins to Use**
2. **Steps to Use a Plugin**
3. **The Way Most Plugins Works**
4. **Before + After of HTML Modified by a Plugin**
5. **Categories of Plugins by Functionality**
6. **Examples**

The Way Most Plugins Work

The JavaScript code in each plugin typically performs the following operations:

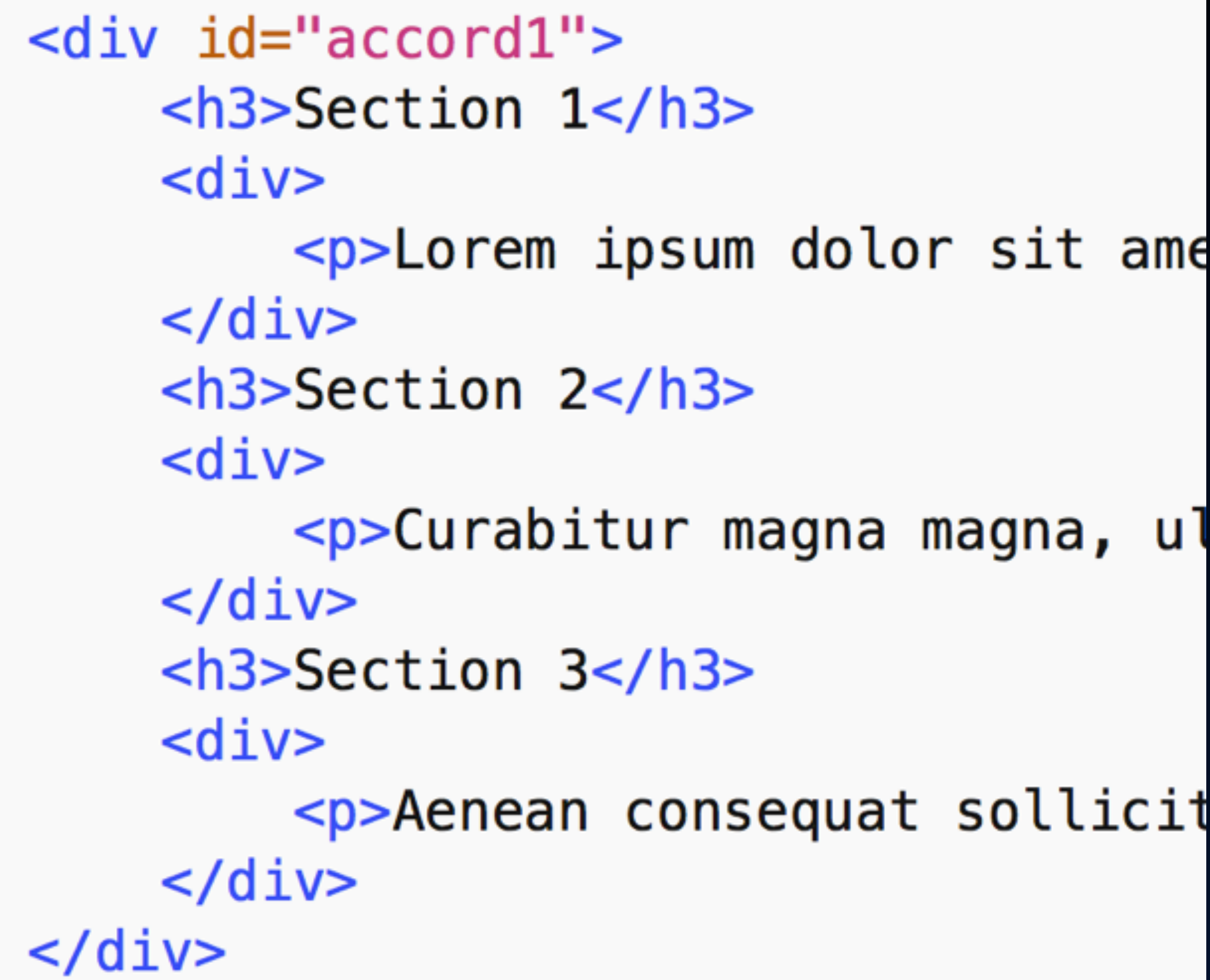
1. **Enhancement of HTML elements by adding and modifying HTML**
2. **Enhancement of HTML elements by application of CSS rules and properties**
3. **Response to range of event triggers: clicks, swipes, etc.**

You usually code an HTML element and its children before calling the JavaScript in the jQuery plugin.

jQuery Plugin Basics

1. Finding Plugins to Use
2. Steps to Use a Plugin
3. The Way Most Plugins Works
4. **Before + After of HTML Modified by a Plugin**
5. Categories of Plugins by Functionality
6. Examples

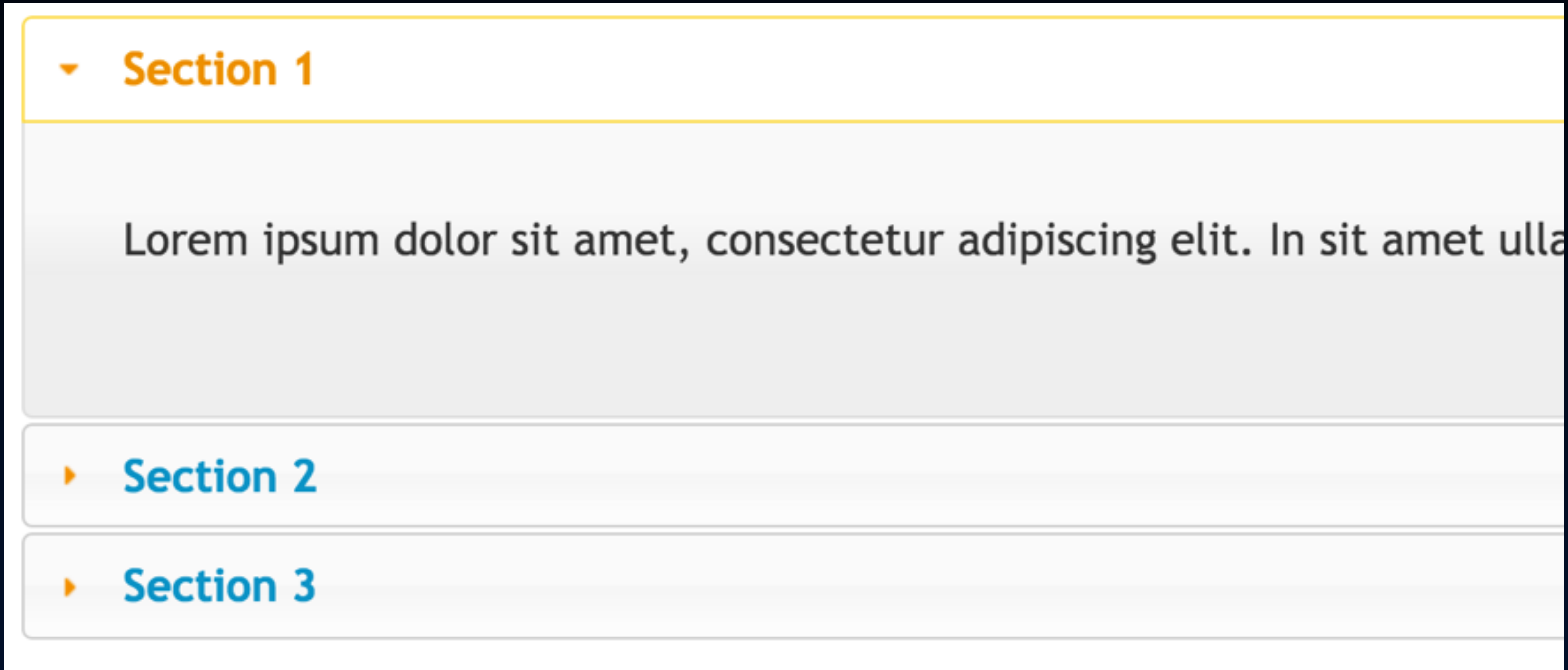
Before + After of HTML Modified by a Plugin



```
<div id="accord1">
    <h3>Section 1</h3>
    <div>
        <p>Lorem ipsum dolor sit ame
    </div>
    <h3>Section 2</h3>
    <div>
        <p>Curabitur magna magna, ul
    </div>
    <h3>Section 3</h3>
    <div>
        <p>Aenean consequat sollicit
    </div>
</div>
```

You write HTML code to match the pattern the jQuery plugin expects.

Before + After of HTML Modified by a Plugin



Section 1

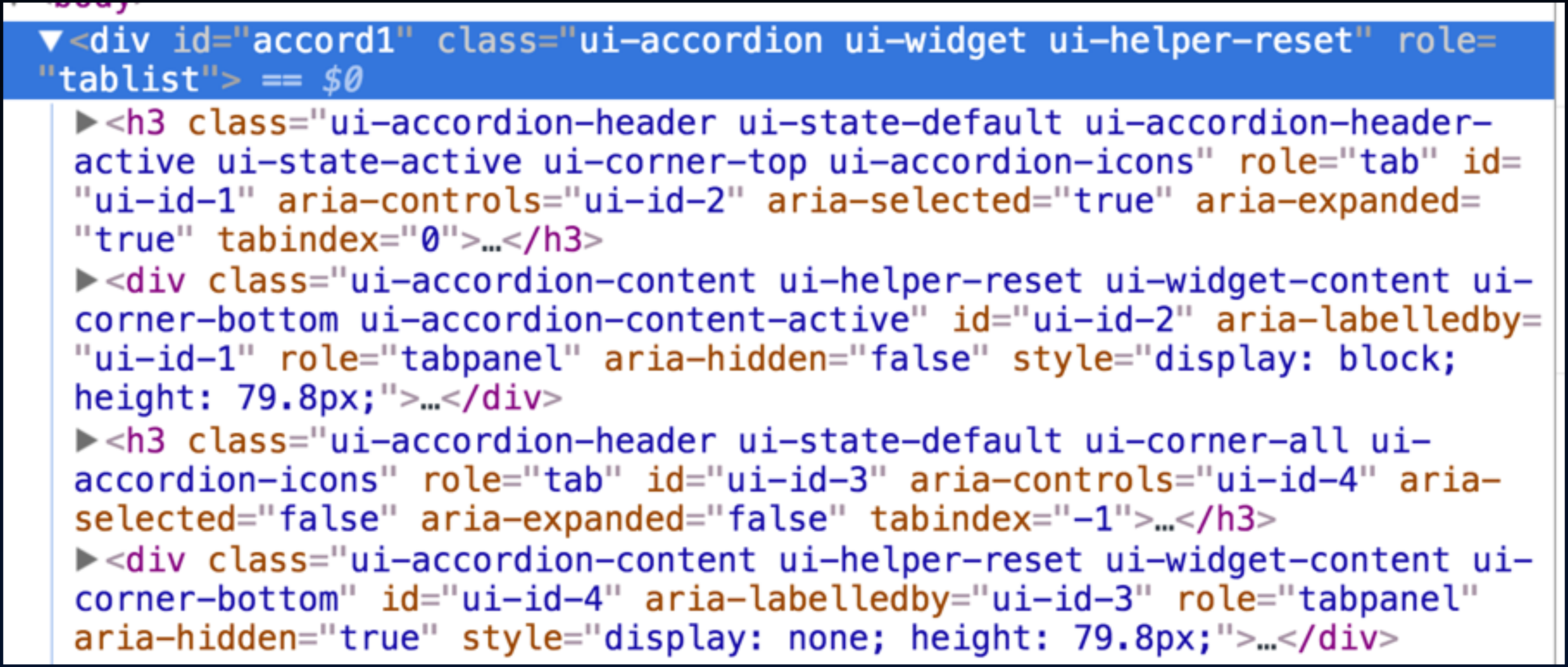
Lorem ipsum dolor sit amet, consectetur adipiscing elit. In sit amet ulla

Section 2

Section 3

The final user interface element as rendered in the browser is actually composed of modified HTML with CSS applied.

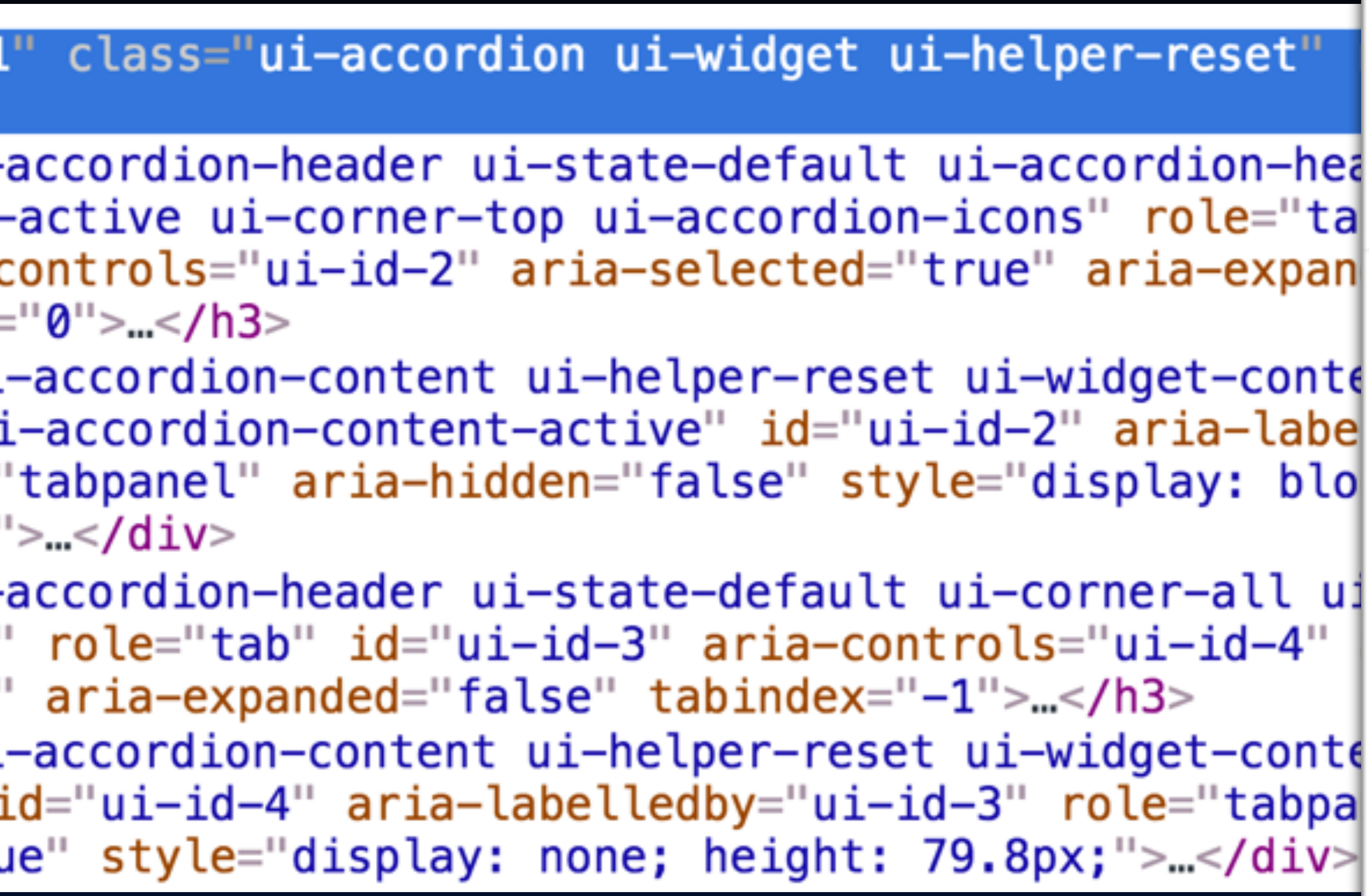
Before + After of HTML Modified by a Plugin



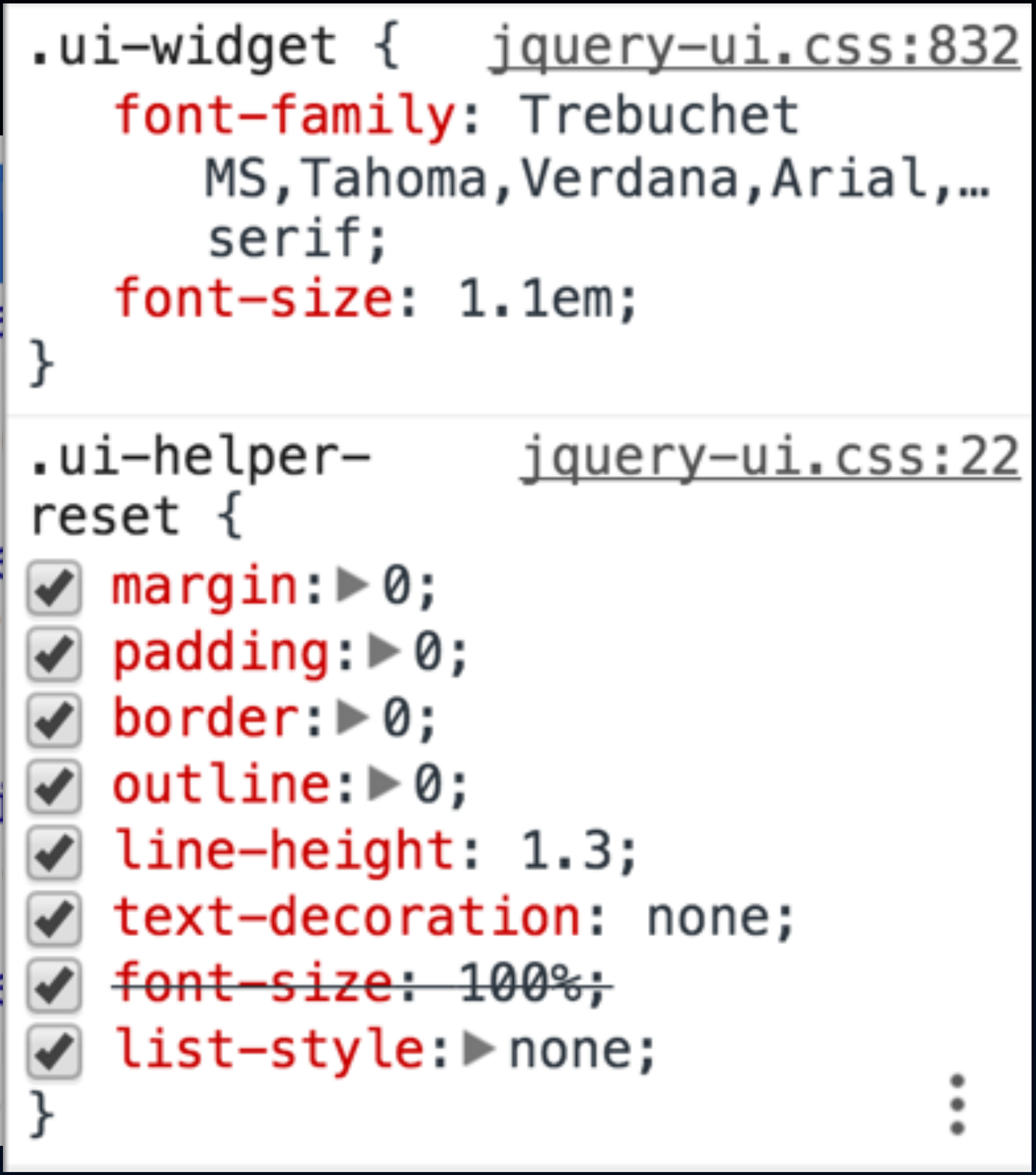
```
▼ <body>
  ▼<div id="accord1" class="ui-accordion ui-widget ui-helper-reset" role=
  "tablist"> == $0
    ▶<h3 class="ui-accordion-header ui-state-default ui-accordion-header-
    active ui-state-active ui-corner-top ui-accordion-icons" role="tab" id=
    "ui-id-1" aria-controls="ui-id-2" aria-selected="true" aria-expanded=
    "true" tabindex="0">…</h3>
    ▶<div class="ui-accordion-content ui-helper-reset ui-widget-content ui-
    corner-bottom ui-accordion-content-active" id="ui-id-2" aria-labelledby=
    "ui-id-1" role="tabpanel" aria-hidden="false" style="display: block;
    height: 79.8px;">…</div>
    ▶<h3 class="ui-accordion-header ui-state-default ui-corner-all ui-
    accordion-icons" role="tab" id="ui-id-3" aria-controls="ui-id-4" aria-
    selected="false" aria-expanded="false" tabindex="-1">…</h3>
    ▶<div class="ui-accordion-content ui-helper-reset ui-widget-content ui-
    corner-bottom" id="ui-id-4" aria-labelledby="ui-id-3" role="tabpanel"
    aria-hidden="true" style="display: none; height: 79.8px;">…</div>
```

The JavaScript code in the jQuery UI plugin's accordion() widget method modifies your original HTML code.

Before + After of HTML Modified by a Plugin



```
l" class="ui-accordion ui-widget ui-helper-reset"
-accordion-header ui-state-default ui-accordion-hea
-active ui-corner-top ui-accordion-icons" role="ta
controls="ui-id-2" aria-selected="true" aria-expan
="0">…</h3>
-accordion-content ui-helper-reset ui-widget-conte
i-accordion-content-active" id="ui-id-2" aria-labe
'tabpanel" aria-hidden="false" style="display: blo
'>…</div>
-accordion-header ui-state-default ui-corner-all ui
' role="tab" id="ui-id-3" aria-controls="ui-id-4"
' aria-expanded="false" tabindex="-1">…</h3>
-accordion-content ui-helper-reset ui-widget-conte
id="ui-id-4" aria-labelledby="ui-id-3" role="tabpa
ue" style="display: none; height: 79.8px;">…</div>
```



```
.ui-widget {     jquery-ui.css:832
   font-family: Trebuchet
      MS,Tahoma,Verdana,Arial,…
      serif;
   font-size: 1.1em;
}
.ui-helper-       jquery-ui.css:22
reset {
☑ margin:▶0;
☑ padding:▶0;
☑ border:▶0;
☑ outline:▶0;
☑ line-height: 1.3;
☑ text-decoration: none;
☑ ~~font-size: 100%;~~
☑ list-style:▶none;
}
```

The modified HTML code contains class attributes that cause the browser to apply CSS rules found in the plugin's CSS file.

jQuery Plugin Basics

1. Finding Plugins to Use
2. Steps to Use a Plugin
3. The Way Most Plugins Works
4. Before + After of HTML Modified by a Plugin
5. **Categories of Plugins by Functionality**
6. Examples

http://plugins.jquery.com/

Categories of Plugins

1. **User Interface Elements**
2. **Effects including Parallax**
3. **Carousels (Touch Gestures)**
4. **Animations**
5. **Transitions + Backgrounds**
6. **Layouts**
7. **Validation**

By Functionality

jQuery Plugin Basics

1. **Finding Plugins to Use**
2. **Steps to Use a Plugin**
3. **The Way Most Plugins Works**
4. **Before + After of HTML Modified by a Plugin**
5. **Categories of Plugins by Functionality**
6. **Examples**

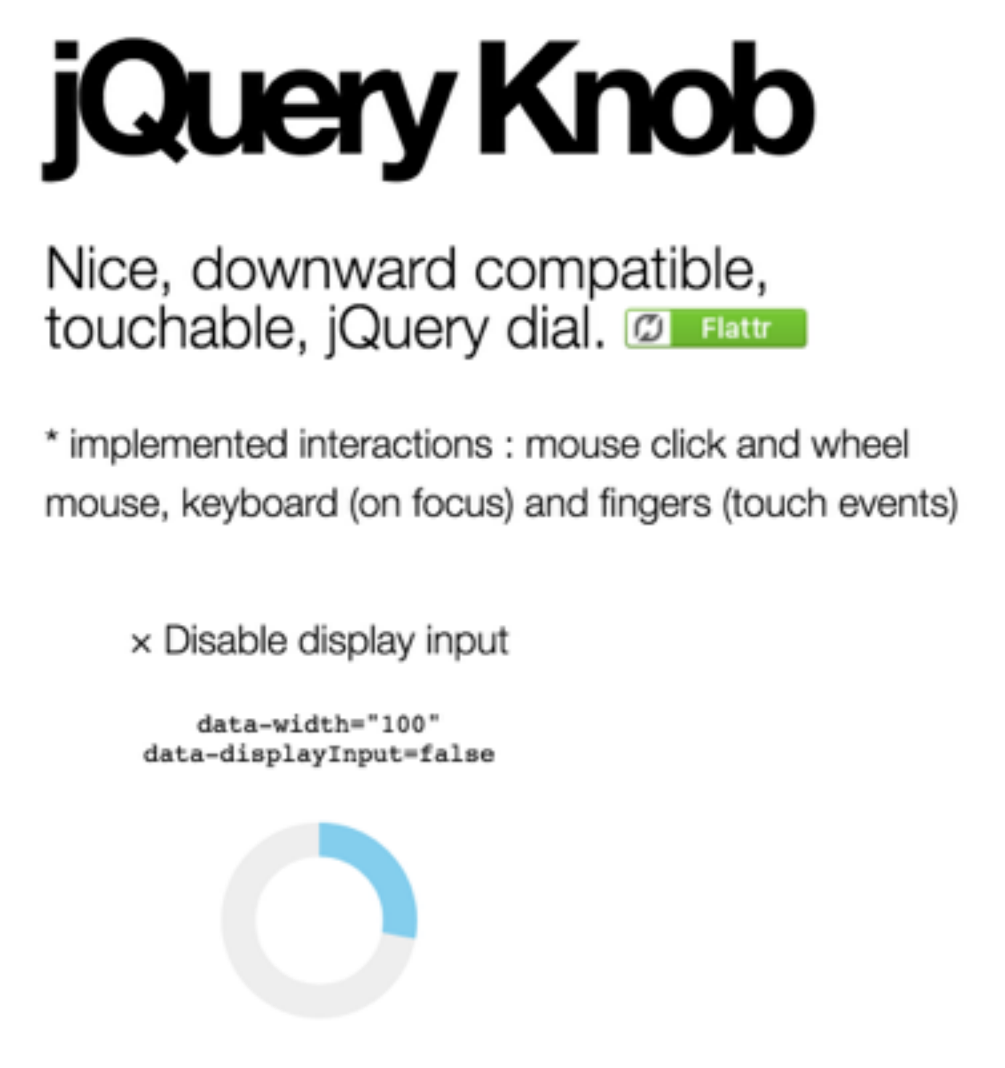
jQuery Plugin Examples

1. User Interface: jQuery Knob
2. Effects including Parallax: Stellar.js
3. Carousels and Touch Gestures: Swipebox
4. Animations: Textillate.js
5. Animations: Super Scrollorama
6. Transitions + Backgrounds: Supersized
7. Layouts: Gridster.js
8. Layouts: Freetile
9. Validation (Web Form): Parsley.js

User Interface:
jQuery Knob

```
<input class="knob"
    data-width="200" value="44"
    data-min="1" data-max="50">
```

```
$(".knob").knob({
    change : function (value) {
        console.log("c:" + value);
    },
    release : function (value) {
        console.log("r:" + value);
    }
});
```



jQuery Knob

Nice, downward compatible, touchable, jQuery dial.

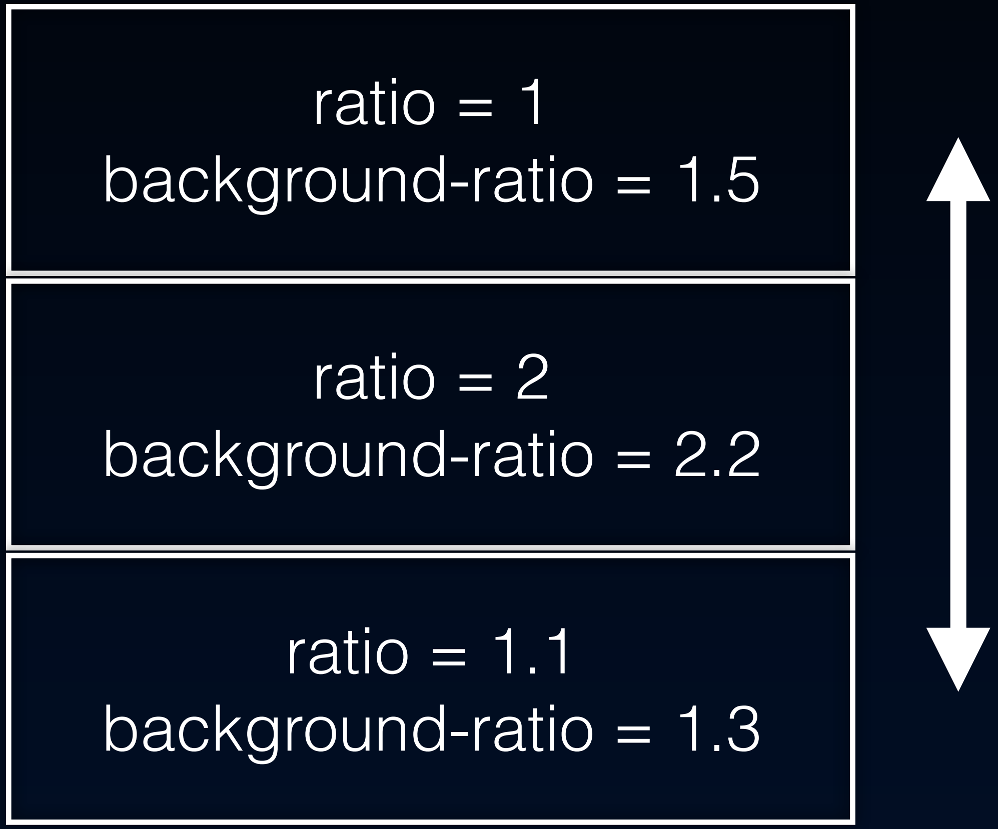
* implemented interactions : mouse click and wheel mouse, keyboard (on focus) and fingers (touch events)

× Disable display input

```
data-width="100"
data-displayInput=false
```

https://github.com/aterrien/jQuery-Knob

Effects including Parallax: Stellar.js



ratio = 1
background-ratio = 1.5

ratio = 2
background-ratio = 2.2

ratio = 1.1
background-ratio = 1.3

ratio + background-ratio control vertical speed in relation to scrolling

Parallax scrolling involves the background moving at a different rate than the foreground, creating a 3D effect as you scroll down the page.

https://github.com/markdalgleish/stellar.js

Effects including Parallax: Stellar.js

```
<div class="slide"
  data-stellar-ratio="1.0"
  data-stellar-vertical-offset="10">
     <img src="image1.jpg" alt="">
</div>

$(window).stellar();
```

https://github.com/markdalgleish/stellar.js

Carousels and Touch Gestures: Swipebox

```
<a href="image1.jpg"
➔ class="swipebox" title="Fog">
    <img src="image1.jpg" alt="image">
</a>

$( '.swipebox' ).swipebox();
```

https://github.com/brutaldesign/swipebox

Animations:
Textillate.js

```
<div class="demo">
    Each of these letters will animate.
</div>

$('.demo').textillate({
    in: { effect: 'rollIn' },
    out: { effect: 'hinge' }
});
```

https://github.com/jschr/textillate

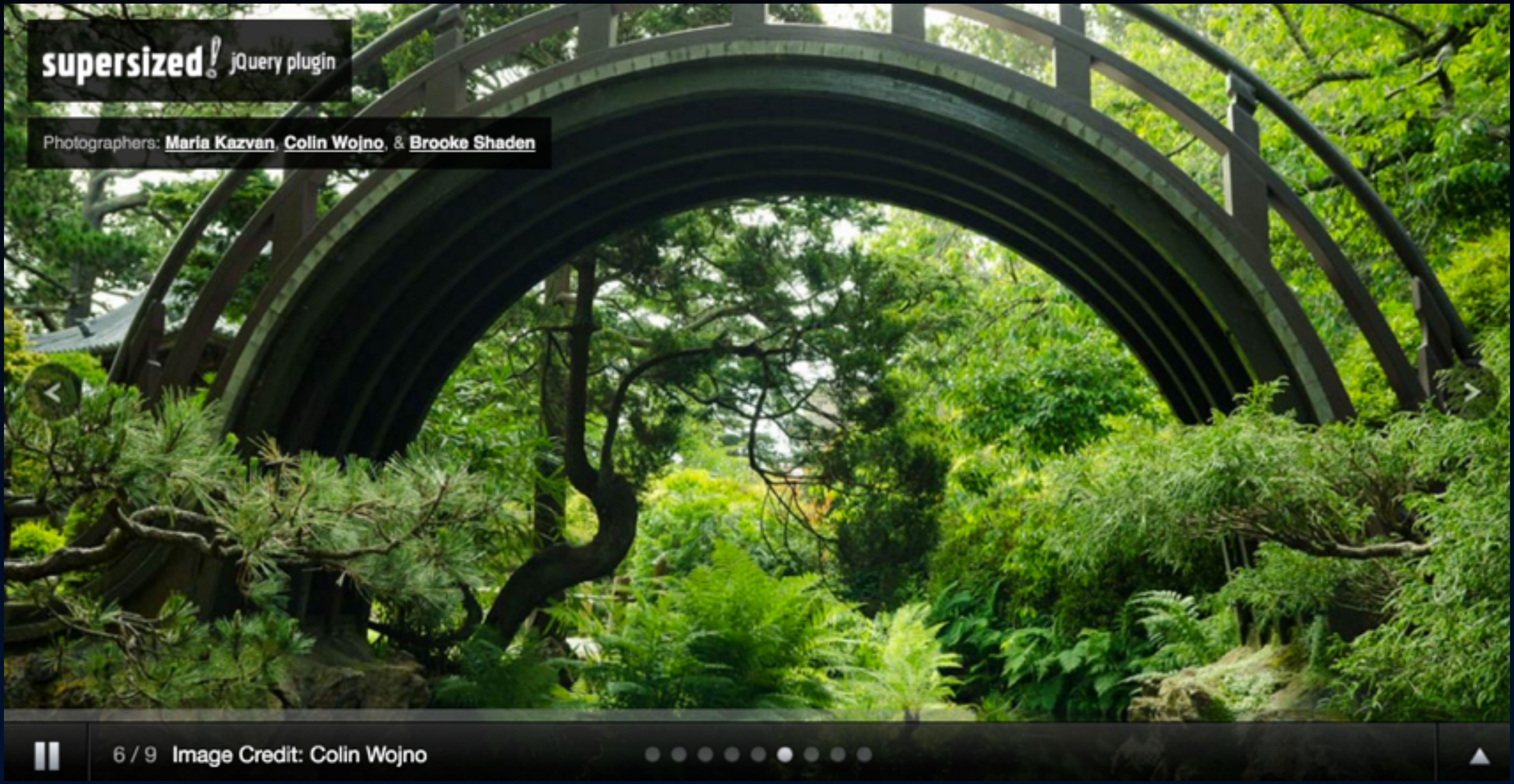
Animations:
Super Scrollorama

```
<h2 id="fly-it">Fly It</h2>

$.superscrollorama().addTween(
    '#fly-it',
    TweenMax.from(
        $('#fly-it'), .25, {
            css:{right:'1000px'},
            ease:Quad.easeInOut
        }
    )
);
```

https://github.com/johnpolacek/superscrollorama

Transitions + Backgrounds: Supersized



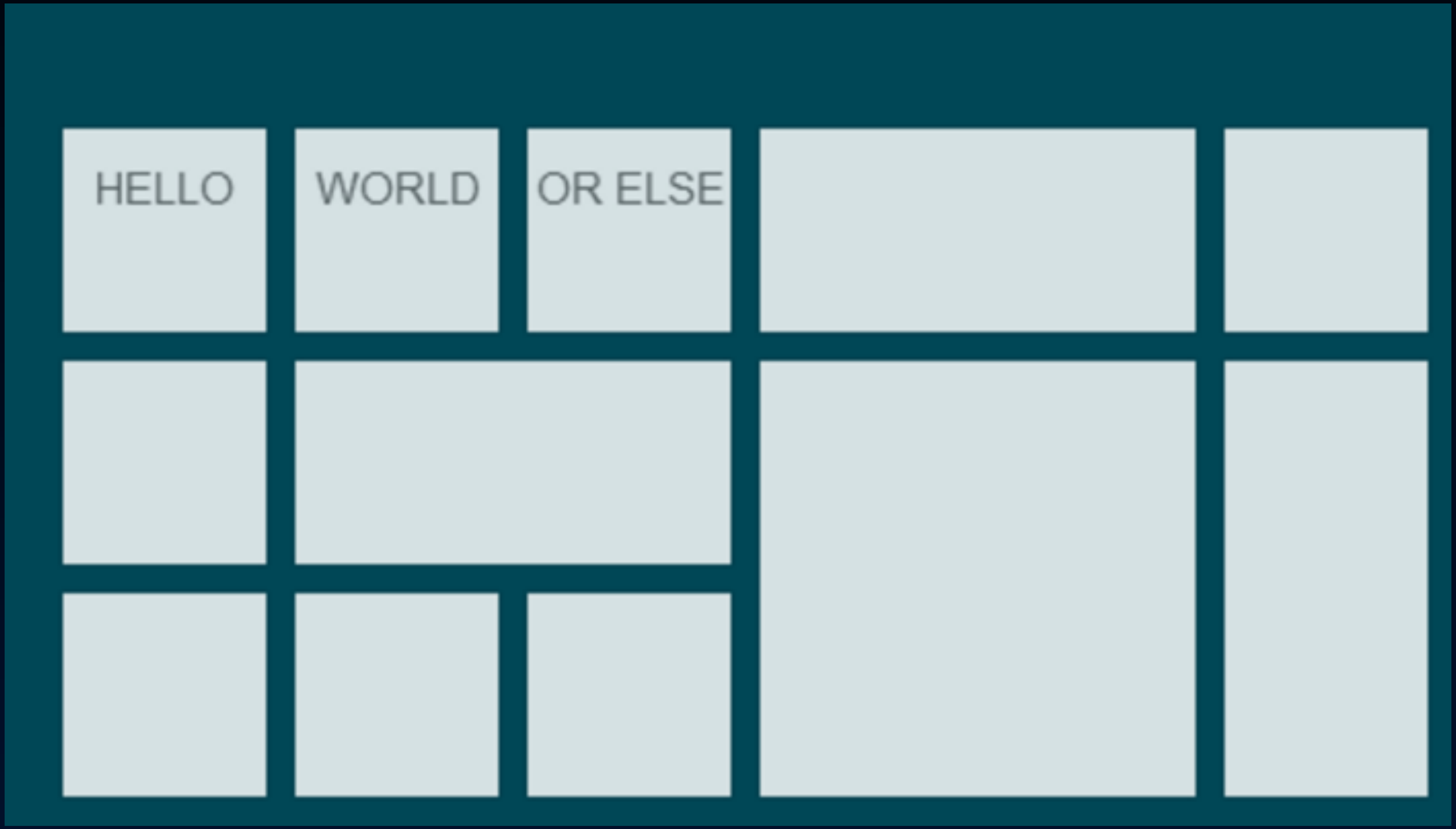
supersized! jQuery plugin

Photographers: Maria Kazvan, Colin Wojno, & Brooke Shaden

6 / 9 Image Credit: Colin Wojno

https://github.com/buildinternet/supersized

Layouts:
Gridster.js



https://github.com/ducksboard/gridster.js

Layouts:
Gridster.js

```
<ul id="grid1">
  <li data-row="1" data-col="2"
      data-sizex="2" data-sizey="1">
     Content Here
  </li>
</ul>

var gridster = $("#grid1").gridster({
   widget_margins: [10, 10],
   widget_base_dimensions: [140, 140],
   min_cols: 6
}).data('gridster');
```

https://github.com/ducksboard/gridster.js

Layouts:
Freetile

```
<div id="demo">
    <div style="width:200px;height:100px;"></div>
    <div style=“width:400px;height:200px;"></div>
</div>

$( '#demo' ).freetile();
```

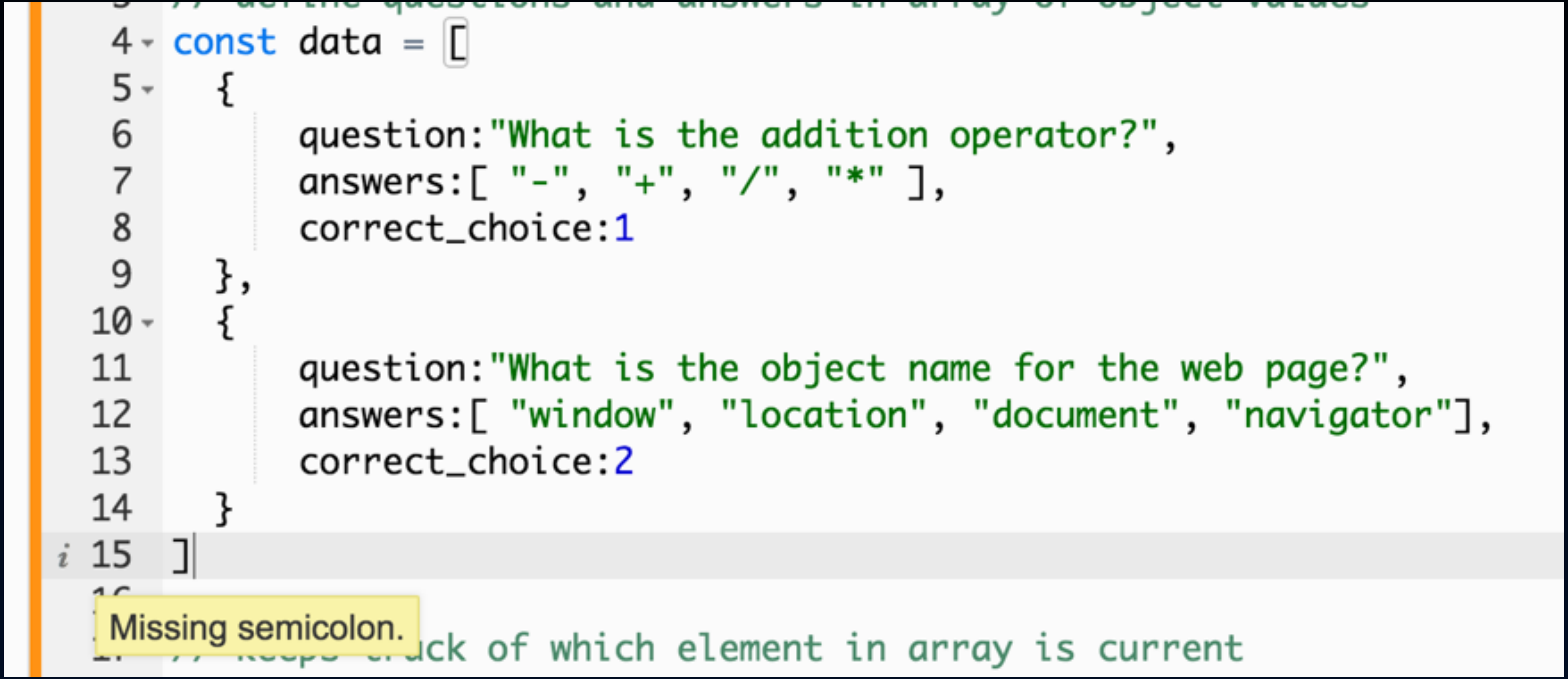
https://github.com/yconst/Freetile

Validation (Web Form): Parsley.js

```
<form data-parsley-validate >
  <label>First Name</label>
  <input type="text"
    name="first_name"
    placeholder="First Name"
    autocomplete="off"
    data-parsley-required />
</form>
```

https://github.com/guillaumepotier/Parsley.js/

JavaScript Coding in Cloud9



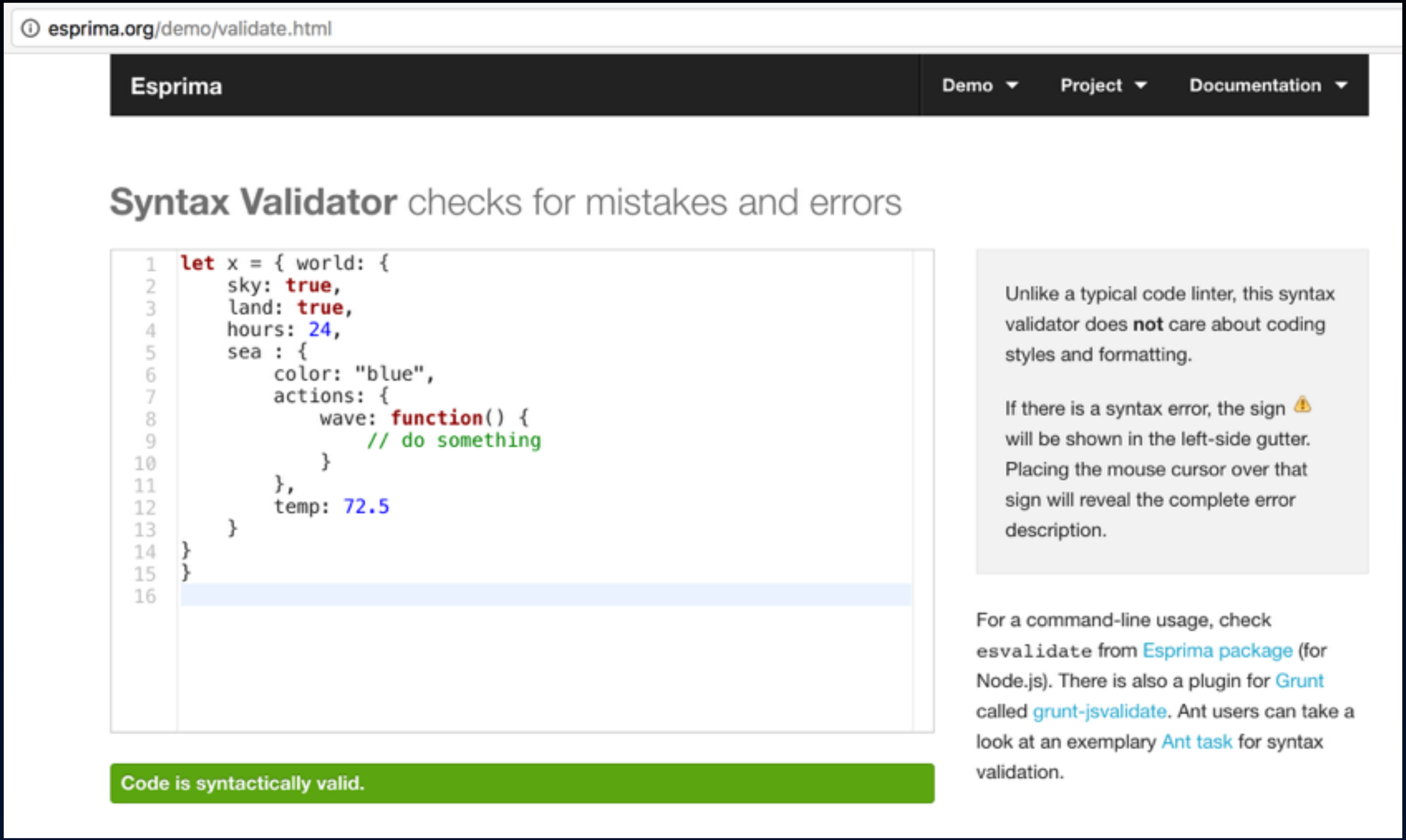
```
// define questions and answers in array of object values
const data = [
  {
      question:"What is the addition operator?",
      answers:[ "-", "+", "/", "*" ],
      correct_choice:1
  },
  {
      question:"What is the object name for the web page?",
      answers:[ "window", "location", "document", "navigator"],
      correct_choice:2
  }
]

// keeps track of which element in array is current
```

Missing semicolon.

Cloud9 offers built-in JavaScript debugging of and syntax checking.

Validating JavaScript Code



Syntax Validator checks for mistakes and errors

```
let x = { world: {
    sky: true,
    land: true,
    hours: 24,
    sea : {
        color: "blue",
        actions: {
            wave: function() {
                // do something
            }
        },
        temp: 72.5
    }
}
}

```

Code is syntactically valid.

Unlike a typical code linter, this syntax validator does **not** care about coding styles and formatting.

If there is a syntax error, the sign ⚠ will be shown in the left-side gutter. Placing the mouse cursor over that sign will reveal the complete error description.

For a command-line usage, check `esvalidate` from Esprima package (for Node.js). There is also a plugin for Grunt called grunt-jsvalidate. Ant users can take a look at an exemplary Ant task for syntax validation.

esprima.org offers live JavaScript code validation.