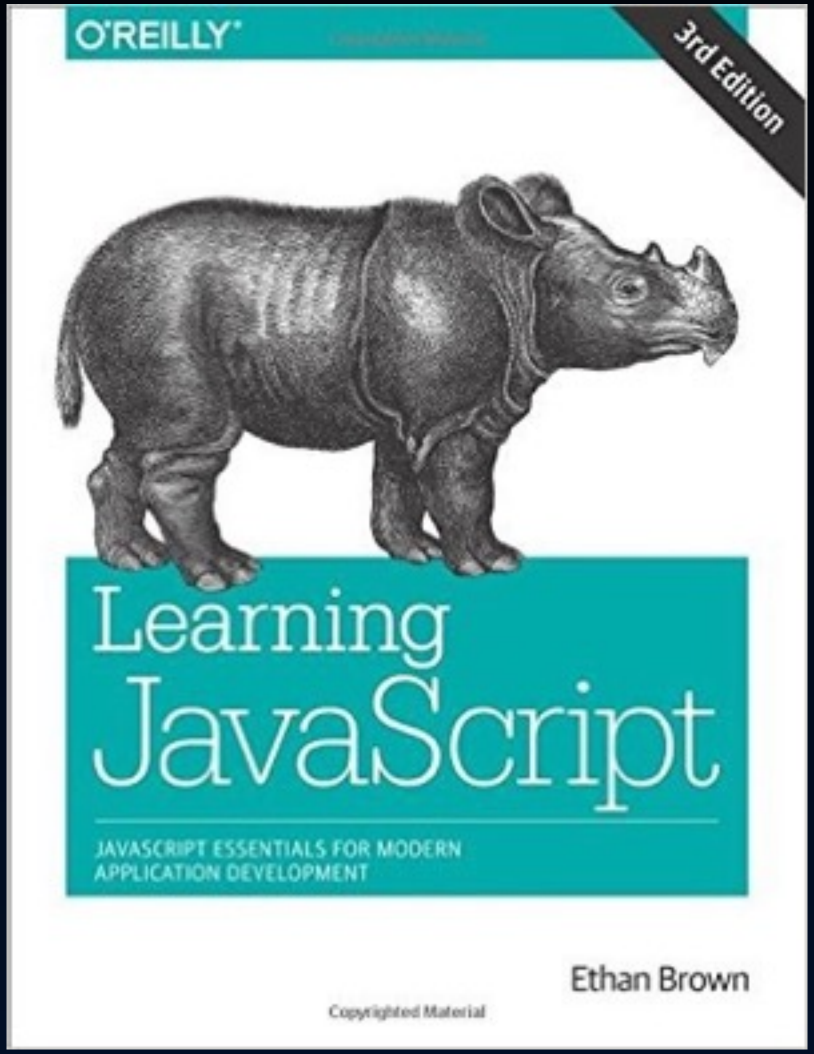
CS55.11 JavaScript

Spring 2017 ~ Ethan Wilde

Week 5

SANTA ROSA
JUNIOR COLLEGE

Textbook



Learning JavaScript

Ethan Brown

ISBN 978-1-491-91491-5

Textbook

- **Reading for Week 2**
- ***Learning JavaScript***
  - **Ch 9: Objects and Object-Oriented Programming**

Object
Classes + Instances

1. **Objects as value type**
2. **Object Properties**
3. **Object Property Values**
4. **Dot Notation to access nested values**
5. **Object Methods**
6. **Object Classes + Instances**

Objects
Object Value Type

1. **JavaScript Object Values**

```
var car = {
    type:"Fiat",
    model:"500",
    color:"white"
};
```

Objects
Object Properties

2. **Object Properties**

```
var car = {
    type:"Fiat", ←    name and value pair
    model:"500",
    details: { color: "white", wheels: 4 }
};
```

Object properties are name-value pairs.

Objects
Object Property Values

3. **Object Property Values**

```
var car = {
    type:"Fiat",
    model:"500",
    details: { color: "white", wheels: 4 }
};
```

object value (nested)

Property values may themselves be object value type.

Objects
Using Dot Notation to Traverse

4. **Dot Notation used to access nested properties**

```
var car = {
    type:"Fiat",
    model:"500",
    details: { color: "white", wheels: 4 }
};

alert( car.details.wheels );
```

dot notation to access

Dot notation lets us access nested object values.

Objects
Methods

5. **Object Methods**

```
var car = {
    type: "Fiat",
    model: "500",
    color: "white",
    getName: function() {
        return this.color + ' ' + this.type + ' ' + this.model;
    }
};

alert( car.getName() );
```

objects can
contain methods
(anonymous functions)

*JavaScript **this** refers to current instance of object.*

Objects
Classes + Instances

ES5

6. **Object Classes**
   We can create an object class that can spawn new instances.

```
function Person(name){
  this.name = name;
  this.getName = function(){
    return this.name;
  }
}
```

a constructor function lets us define a reusable object class (in ES5)

ES5 uses **function** to define a class

```
var person1 = new Person("Fred");
```

*JavaScript **this** refers to current instance of object.*

Objects
Classes + Instances

ES6

6. **Object Classes**
   We can create an object class that can spawn new instances.

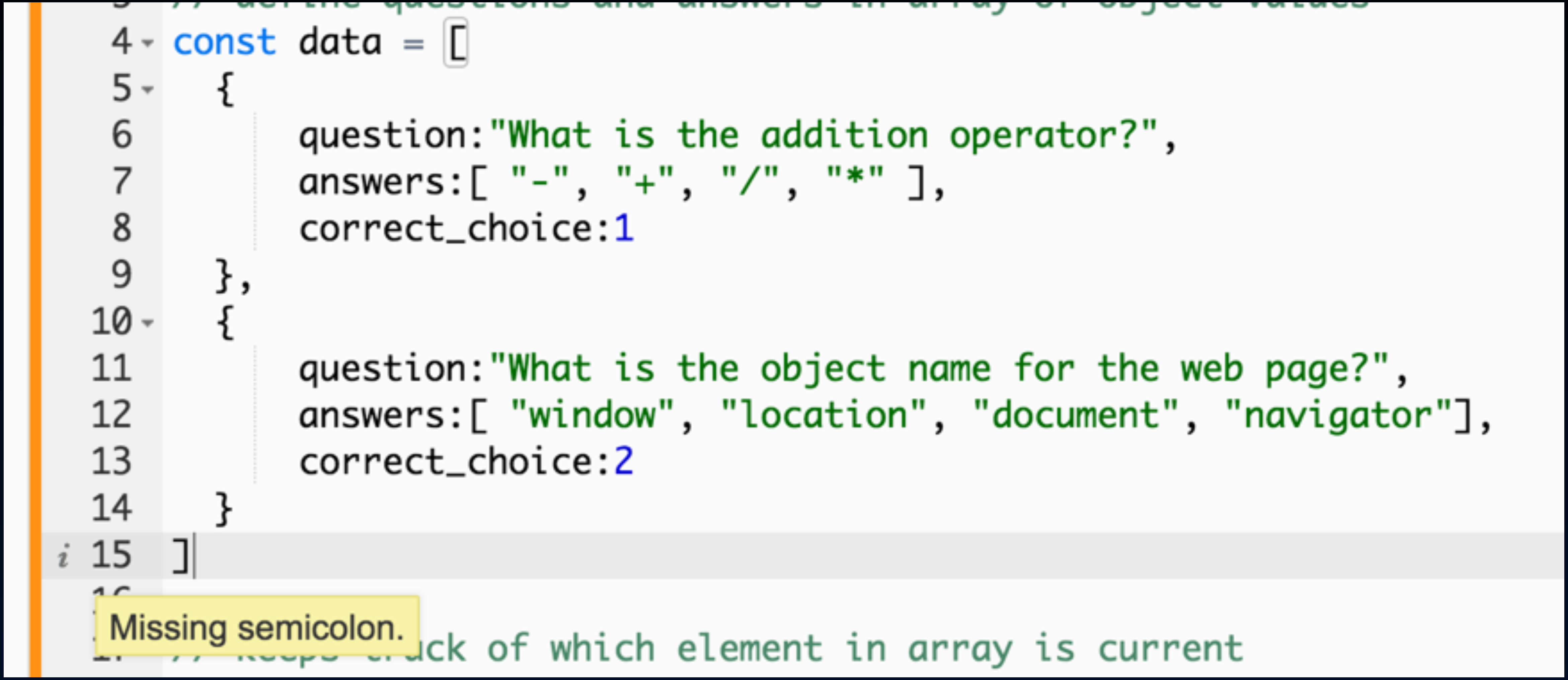
```
class Person() {
  constructor(name) {
    this.name = name;
    this.getName = function(){
      return this.name;
    }
  }
}

var person1 = new Person("Fred");
```

← a constructor function lets us define a reusable object class (in ES6)
ES6 introduces **class** and **constructor**

*JavaScript **this** refers to current instance of object.*

JavaScript Coding in Cloud9



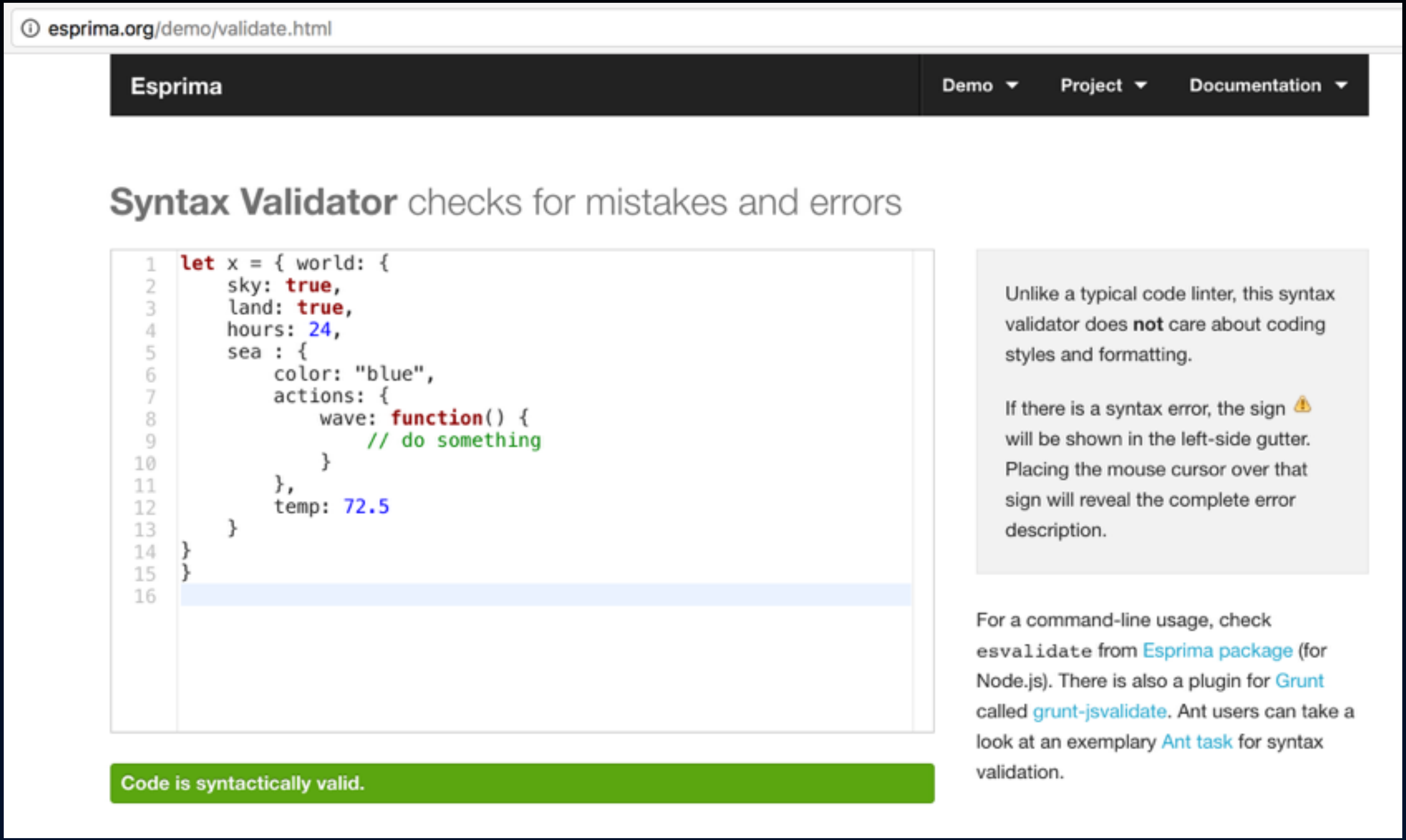
```
// define questions and answers in array of object values
const data = [
  {
      question:"What is the addition operator?",
      answers:[ "-", "+", "/", "*" ],
      correct_choice:1
  },
  {
      question:"What is the object name for the web page?",
      answers:[ "window", "location", "document",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

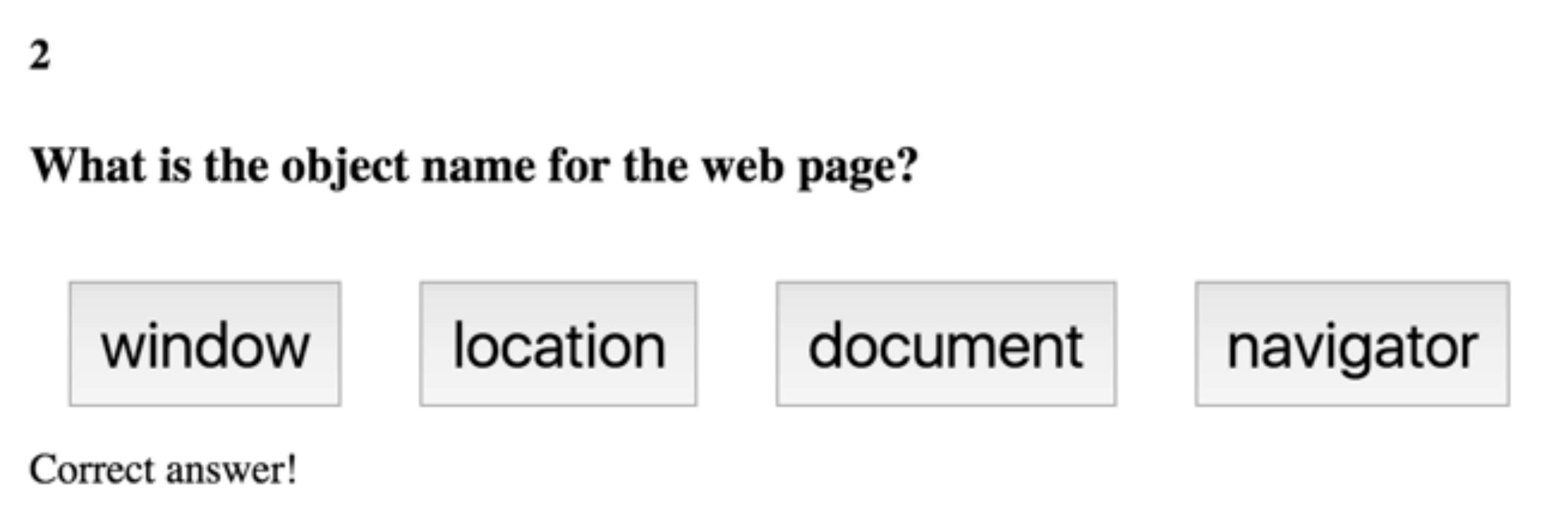
If there is a syntax error, the sign ⚠ will be shown in the left-side gutter. Placing the mouse cursor over that sign will reveal the complete error description.

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.

Refactoring Procedural Code into Object-Oriented

“Let’s re-write our quiz as a reusable class…”



2

What is the object name for the web page?

Correct answer!

Transform your code from last week into a fully object-oriented Quiz() class.

jQuery + jQuery UI

1. **jQuery Selector Recap**
2. **jQuery UI Basics**
3. **jQuery UI Widgets**
4. **jQuery UI Interactions**
5. **jQuery UI Effects**
6. **jQuery UI Custom Themes**

jQuery Selectors: Recap

1. **DOM objects**
2. **CSS selectors**
3. **other special selectors**

jQuery Selectors

```
$(document).ready();

$('.pickles').html('<p>hey</p>');

$('#thing').clear();
```

DOM OBJECTS OR CSS SELECTORS

jQuery Selectors

```
<p class='pickles'>Happy</p>

<script>
$('.pickles').css('color', 'red');
</script>
```

CSS CLASS SELECTORS

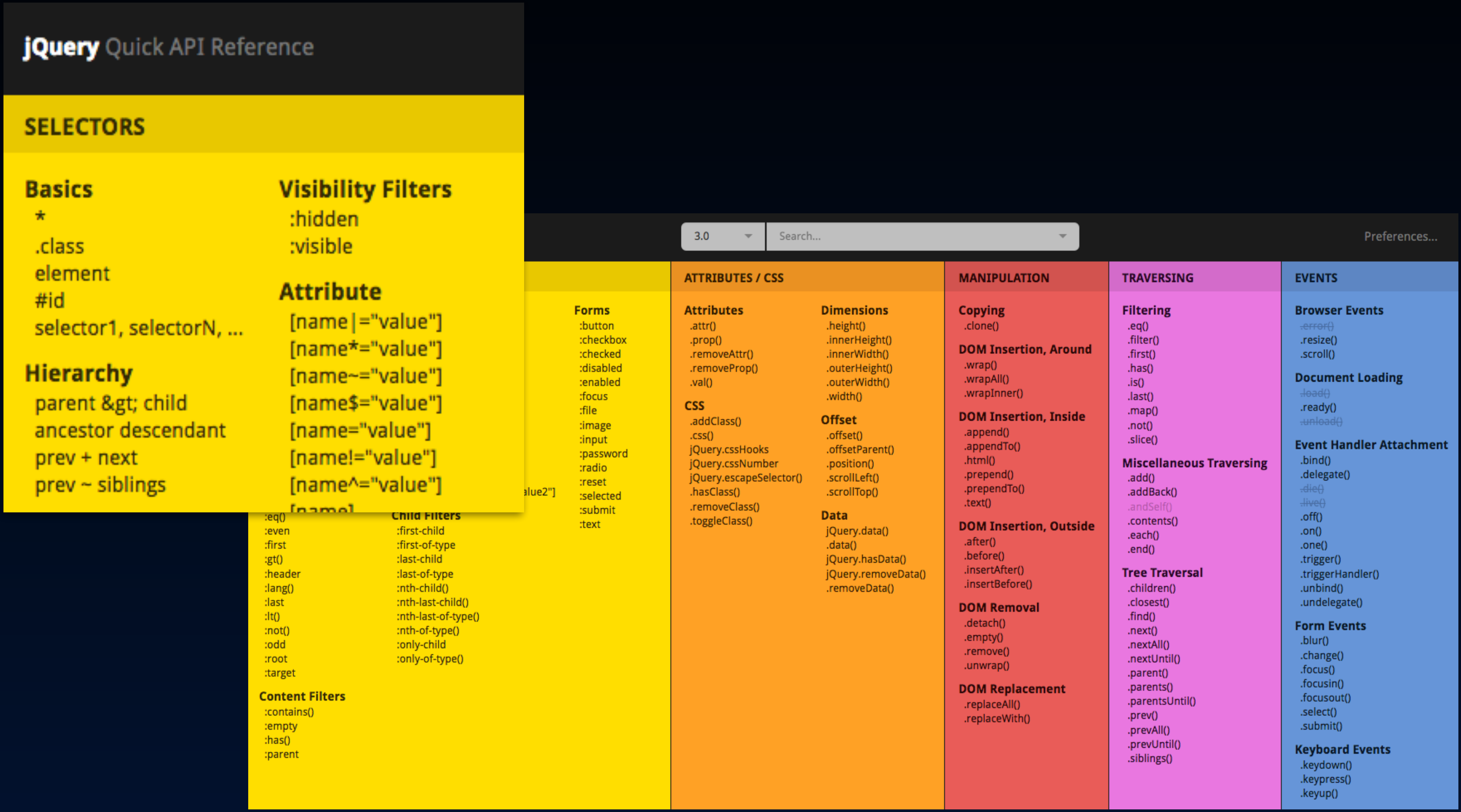
jQuery Selectors

```
<p id='thing'>Happy</p>

<script>
$('#thing').css('color', 'red');
</script>
```

CSS ID SELECTORS

jQuery Selectors



jQuery Quick API Reference

SELECTORS

Basics

- *
- .class
- element
- #id
- selector1, selectorN, ...

Hierarchy

- parent &gt; child
- ancestor descendant
- prev + next
- prev ~ siblings

Visibility Filters

- :hidden
- :visible

Attribute

- [name|="value"]
- [name*="value"]
- [name~="value"]
- [name$="value"]
- [name="value"]
- [name!="value"]
- [name^="value"]
- [name]

alue2"]

:eq()
:even
:first
:gt()
:header
:lang()
:last
:lt()
:not()
:odd
:root
:target

Content Filters

:contains()
:empty
:has()
:parent

Child Filters

:first-child
:first-of-type
:last-child
:last-of-type
:nth-child()
:nth-last-child()
:nth-last-of-type()
:nth-of-type()
:only-child
:only-of-type()

Forms

:button
:checkbox
:checked
:disabled
:enabled
:focus
:file
:image
:input
:password
:radio
:reset
:selected
:submit
:text

ATTRIBUTES / CSS

Attributes

.attr()
.prop()
.removeAttr()
.removeProp()
.val()

CSS

.addClass()
.css()
jQuery.cssHooks
jQuery.cssNumber
jQuery.escapeSelector()
.hasClass()
.removeClass()
.toggleClass()

Dimensions

.height()
.innerHeight()
.innerWidth()
.outerHeight()
.outerWidth()
.width()

Offset

.offset()
.offsetParent()
.position()
.scrollLeft()
.scrollTop()

Data

jQuery.data()
.data()
jQuery.hasData()
jQuery.removeData()
.removeData()

MANIPULATION

Copying

.clone()

DOM Insertion, Around

.wrap()
.wrapAll()
.wrapInner()

DOM Insertion, Inside

.append()
.appendTo()
.html()
.prepend()
.prependTo()
.text()

DOM Insertion, Outside

.after()
.before()
.insertAfter()
.insertBefore()

DOM Removal

.detach()
.empty()
.remove()
.unwrap()

DOM Replacement

.replaceAll()
.replaceWith()

TRAVERSING

Filtering

.eq()
.filter()
.first()
.has()
.is()
.last()
.map()
.not()
.slice()

Miscellaneous Traversing

.add()
.addBack()
.andSelf()
.contents()
.each()
.end()

Tree Traversal

.children()
.closest()
.find()
.next()
.nextAll()
.nextUntil()
.parent()
.parents()
.parentsUntil()
.prev()
.prevAll()
.prevUntil()
.siblings()

EVENTS

Browser Events

~~.error()~~
.resize()
.scroll()

Document Loading

~~.load()~~
.ready()
~~.unload()~~

Event Handler Attachment

.bind()
.delegate()
~~.die()~~
~~.live()~~
.off()
.on()
.one()
.trigger()
.triggerHandler()
.unbind()
.undelegate()

Form Events

.blur()
.change()
.focus()
.focusin()
.focusout()
.select()
.submit()

Keyboard Events

.keydown()
.keypress()
.keyup()

https://oscarotero.com/jquery/

jQuery UI Basics

1. **History**
2. **Version Compatibility + jQuery**
3. **Widgets**
4. **Interactions**
5. **Effects**
6. **Custom Themes**

http://jqueryui.com

jQuery UI Basics

1. History
2. Version Compatibility + jQuery
3. Widgets
4. Interactions
5. Effects
6. Custom Themes

History of jQuery UI

version	date
1	Sept 2007
1.5	Jun 2008
1.9	Oct 2012
1.12	Jan 2016

jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.

jQuery UI Basics

1. History
2. Version Compatibility + jQuery
3. Widgets
4. Interactions
5. Effects
6. Custom Themes

Version Compatibility with jQuery

jQuery UI	jQuery
version 1.6	version 1.2.6+
versions 1.7 – 1.8.x	version 1.3.2+
versions 1.9 – 1.11.x	version 1.6+
version 1.12.x	version 1.7+

Make sure you are using compatible versions.

Using jQuery UI

You must load

(1) jQuery library .js file

(2) jQuery UI .js files *(1 or more)*

(3) jQuery UI .css files *(1 or more)*

jQuery UI Basics

1. History
2. Version Compatibility + jQuery
3. Widgets
4. Interactions
5. Effects
6. Custom Themes

jQuery UI Widgets

Accordion

Autocomplete

Button

Checkboxradio

Controlgroup

Datepicker

Dialog

Menu

Progressbar

Selectmenu

Slider

Spinner

Tabs

Tooltip

jQuery UI Widgets

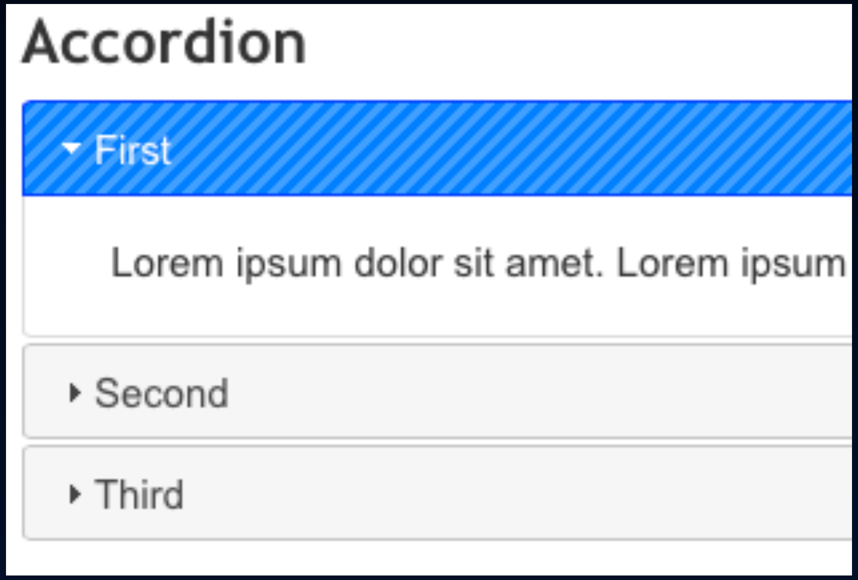
Accordion
Autocomplete
Button
Checkboxradio
Controlgroup
Datepicker
Dialog

Menu
Progressbar
Selectmenu
Slider
Spinner
Tabs
Tooltip

jQuery UI Widgets

```
<div id='accord1'>
    <h3>Panel 1</h3>
    <div>Contents 1</div>
    <h3>Panel 2</h3>
    <div>Contents 2</div>
</div>

<script>
    $('#accord1').accordion();
</script>
```



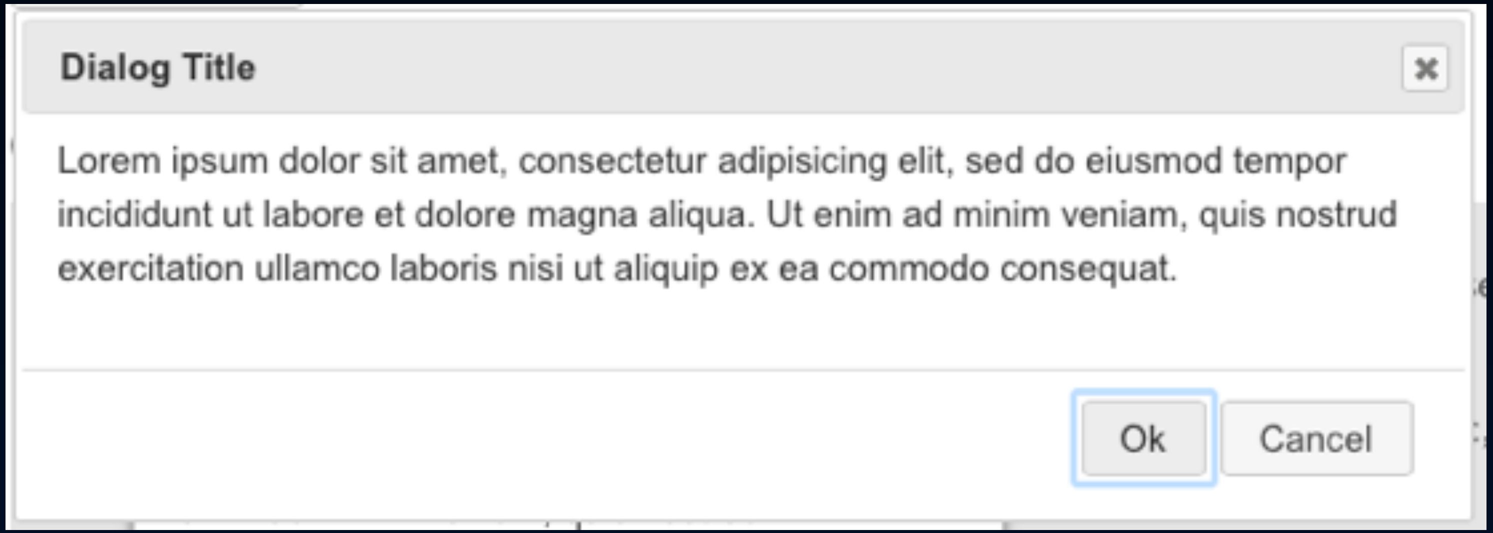
Accordion

Lorem ipsum dolor sit amet. Lorem ipsum

Accordion

jQuery UI Widgets

```
<div id='dialog1' title='Dialog Title'>
    <p>Contents of dialog</p>
</div>
```



Dialog Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

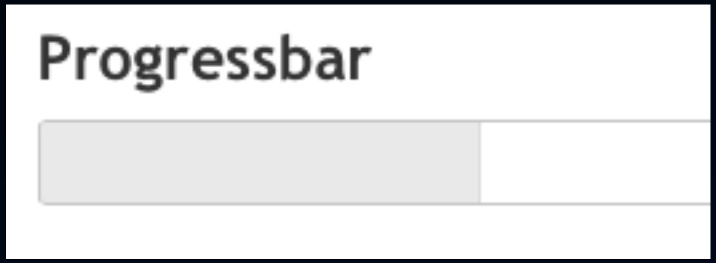
```
<script>
    $('#dialog1').dialog();
</script>
```

Dialog

jQuery UI Widgets

```
<div id='prog1'></div>

<script>
    $('#prog1').progressbar(
        {
            value: 37,
            max: 100,
        }
    );
</script>
```

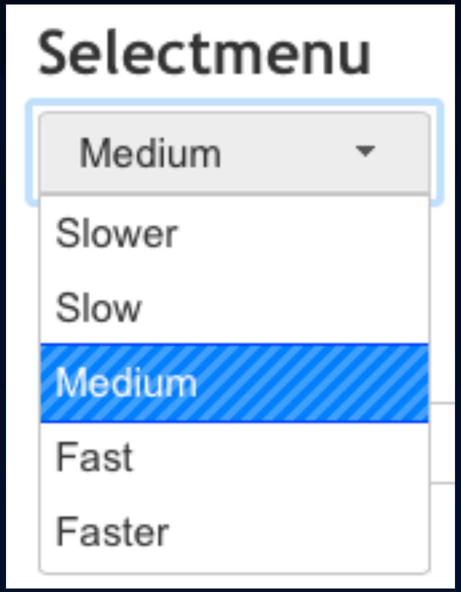


Progressbar

jQuery UI Widgets

```
<label for="level1">Select level</label>
<select id="level1">
    <option>Slow</option>
    <option>Medium</option>
    <option selected>Fast</option>
</select>

<script>
    $('#level1').selectmenu();
</script>
```

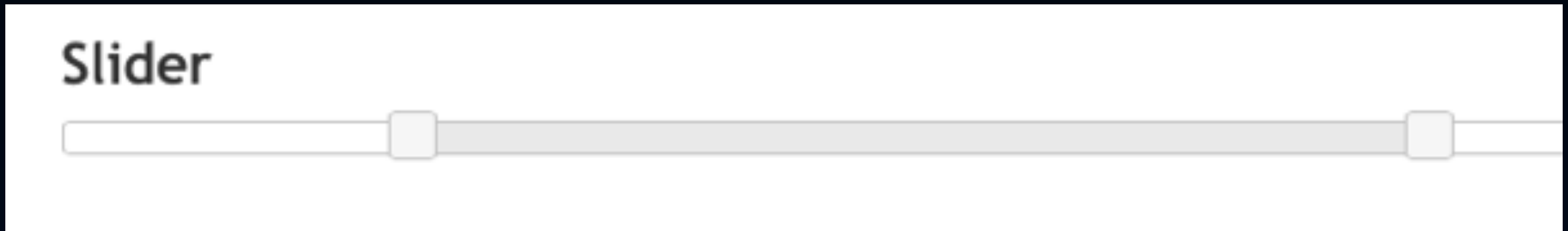


Selectmenu

jQuery UI Widgets

```
<div id='slide1'></div>

<script>
    $('#slide1').slider(
        {
            min: 1,
            max: 10
        }
    );
</script>
```

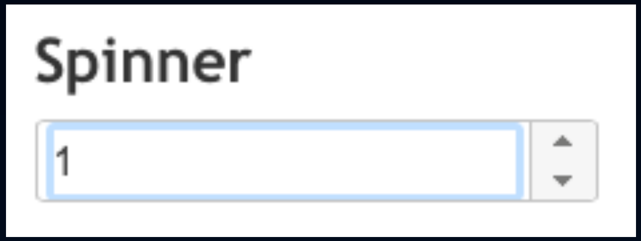


Slider

jQuery UI Widgets

```
<label for="spin1">Select value</label>
<input id="spin1">

<script>
    $('#spin1').spinner(
        {
            min: 1,
            max: 10
        }
    );
</script>
```



Spinner

1

Spinner

jQuery UI Basics

1. **History**
2. **Version Compatibility + jQuery**
3. **Widgets**
4. **Interactions**
5. **Effects**
6. **Custom Themes**

jQuery UI Interactions

Draggable

Droppable

Resizable

Selectable

Sortable

Widgets

jQuery UI Interactions

Draggable

Droppable

Resizable

Selectable

Sortable

Widgets

jQuery UI Interactions

```
<div id="drag1">
    <p>Drag me!</p>
</div>

<script>
    $('#drag1').draggable();
</script>
```

Draggable

jQuery UI Interactions

```
<div id="drag1">
    <p>Drag me!</p>
</div>
<div id="drop1">
    <p>Drop it here</p>
</div>

<script>
    $('#drag1').draggable();
    $('#drop1').droppable(
        drop: function() {
            alert("Dropped!");
        }
    );
</script>
```

Droppable

jQuery UI Interactions

```
<div id="drag1">
    <p>Drag me!</p>
</div>
<div id="drop1">
    <p>Drop it here</p>
</div>

<script>
    $('#drag1').draggable();
    $('#drop1').droppable(
        drop: function() {
            alert("Dropped!");
        }
    );
</script>
```

Droppable

jQuery UI Basics

1. History
2. Version Compatibility + jQuery
3. Widgets
4. Interactions
5. **Effects**
6. Custom Themes

jQuery UI Effects

Add Class
Color Animation
Easing
Effect
Hide
Remove Class
Show
Switch Class
Toggle
Toggle Class

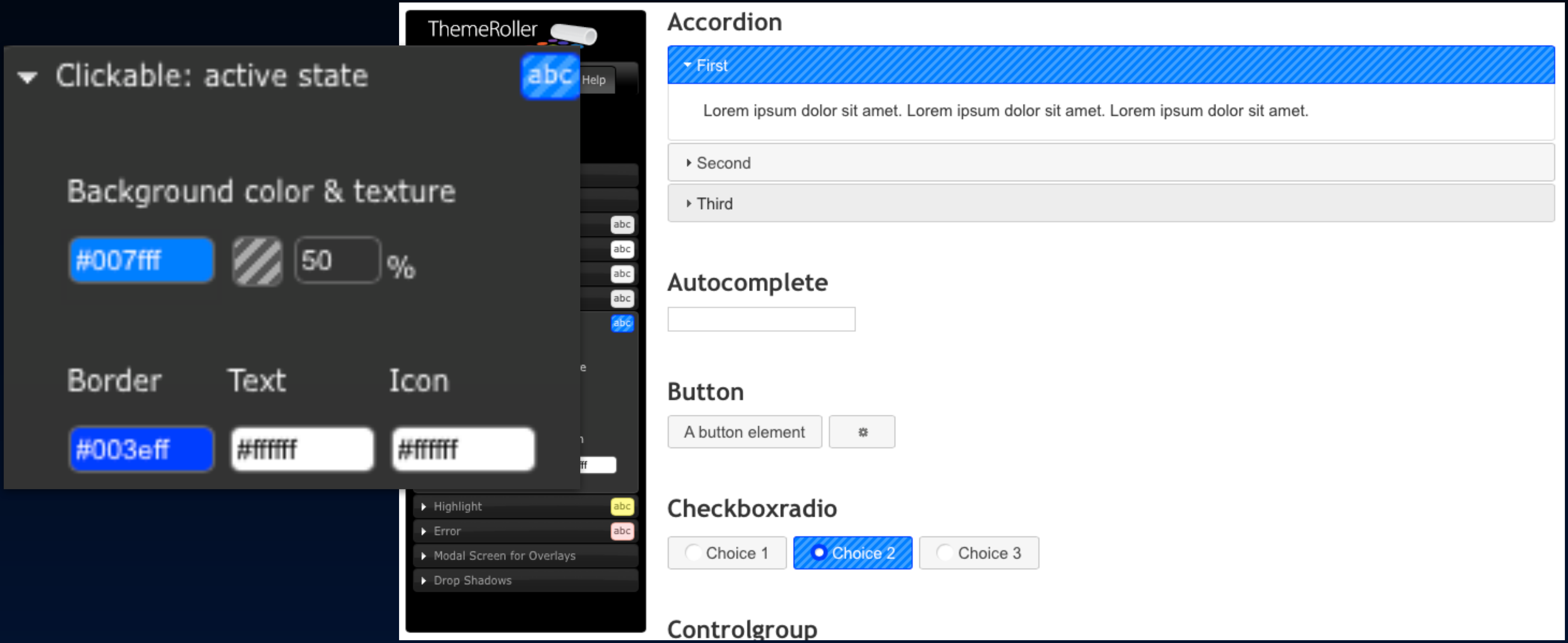
jQuery Library includes some of these effects already.

jQuery UI Basics

1. History
2. Version Compatibility + jQuery
3. Widgets
4. Interactions
5. Effects
6. **Custom Themes**

Custom Themes

The jQuery UI ThemeRoller



Clickable: active state

Background color & texture

#007fff ▨ 50 %

Border: #003eff

Text: #ffffff

Icon: #ffffff

Accordion

▾ First

Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

▸ Second

▸ Third

Autocomplete

Button

A button element

Checkboxradio

○ Choice 1 ● Choice 2 ○ Choice 3

Controlgroup

http://jqueryui.com/themeroller/