

# 1 Designing for Web 2.0

CSC443: Web Programming

## Navigation and links

- Menus:
  - ▣ Horizontal
  - ▣ Vertical
  - ▣ Flyout
- Efficient forms
  - ▣ Proper input elements
  - ▣ Min number of fields
  - ▣ Javascript/PHP for validation

CSC443: Web Programming

## Usability

2

- Summarize
- Organize
- Write compactly
- Don't be too creative!

CSC443: Web Programming

## 4 Visual Effects

CSC443: Web Programming

## Scriptaculous overview

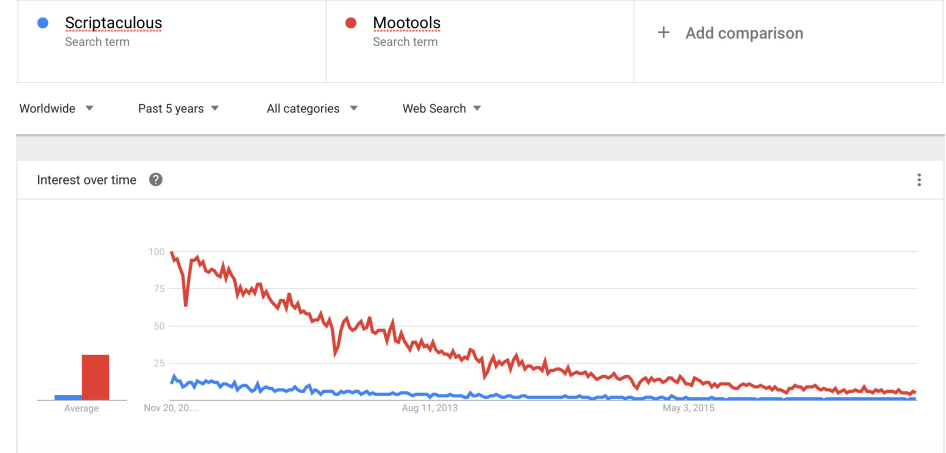
5

- ❑ Scriptaculous: a JavaScript library, built on top of Prototype, that adds:
  - visual effects (animation, fade in/out, highlighting)
  - drag and drop
  - some DOM enhancements
  - other stuff (unit testing, etc.)
- ❑ People are moving away from Scriptaculous towards newer frameworks such as jQuery UI

CSC443: Web Programming

## Trends: Scriptaculous vs. Mootools

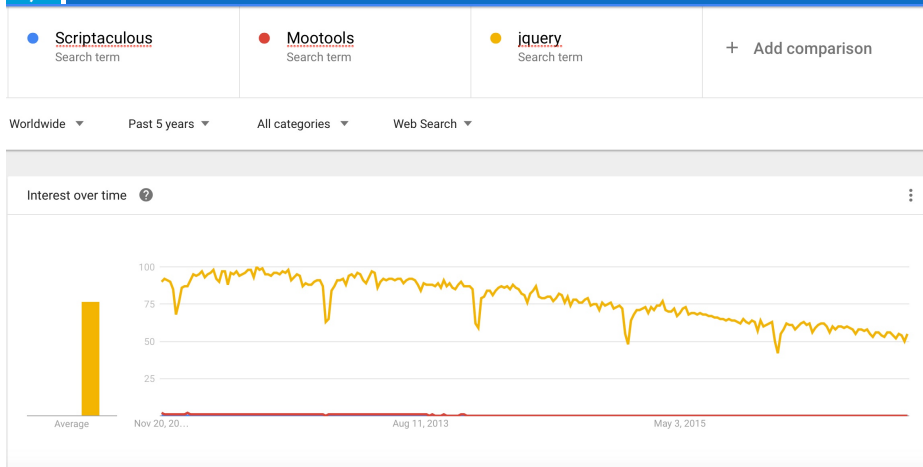
6



CSC443: Web Programming

## Trends: Scriptaculous vs. Mootools vs. jQuery

7



CSC443: Web Programming

## Downloading and using Scriptaculous

8

```
<script  
src="http://ajax.googleapis.com/ajax/libs/scriptaculous/1.  
9.0/scriptaculous.js" type="text/javascript"></script>
```

JS

- ❑ documentation available on their [wiki](#)
- ❑ [Scriptaculous Effects Cheat Sheet](#)

CSC443: Web Programming

## Visual Effects

9

appear blindDown grow slideDown (appearing)

blindUp dropOut fade fold puff

shrink slideUp squish switchOff (disappearing)

highlight pulsate shake morph

Effect.Move Effect.Scale Effect.toggle (blind) (Getting attention)



CSC443: Web Programming

## Adding effects to an element

10

```
element.effectName(); // for most effects

// some effects must be run the following way:
new Effect.name(element or id); JS

$("sidebar").shake();
var buttons = $$("results > button");
for (var i = 0; i < buttons.length; i++) {
    buttons[i].fade();
} JS
```

- the effect will begin to animate on screen (asynchronously) the moment you call it
- six core effects are used to implement all effects on the previous slides:

Effect.Highlight, Effect.Morph, Effect.Move, Effect.Opacity, Effect.Parallel, Effect.Scale

## Adding effects to an element

11

```
element.effectName(
{
    option: value,
    option: value,
    ...
}
); JS
```

```
$("#my_element").pulsate({
    duration: 2.0,
    pulses: 2
}); JS
```

- many effects can be customized by passing additional options (note the {})
- options (wiki): delay, direction, duration, fps, from, queue, sync, to, transition

CSC443: Web Programming

## Adding effects to an element

12

```
$("#my_element").fade({
    duration: 3.0,
    afterFinish: displayMessage
});
function displayMessage(effect) {
    alert(effect.element + " is done fading now!");
} JS
```

- all effects have the following events that you can handle:
  - beforeStart, beforeUpdate, afterUpdate, afterFinish
- the afterFinish event fires once the effect is done animating
  - useful do something to the element (style, remove, etc.) when effect is done

CSC443: Web Programming

## Adding effects to an element

13

```
$( "my_element" ).fade({
    duration: 3.0,
    afterFinish: displayMessage
});
function displayMessage(effect) {
    alert(effect.element + " is done fading now!");
}
```

JS

- each of these events receives the Effect object as its parameter
  - its properties: element, options, currentFrame, startOn, finishOn
  - some effects (e.g. Shrink) are technically "parallel effects", so to access the modified element, you write `effect.effects[0].element` rather than just `effect.element`

CSC443: Web Programming

## Draggable

15

```
new Draggable(element or id,
    { options }
);
```

JS

- specifies an element as being able to be dragged
- options: handle, revert, snap, zIndex, constraint, ghosting, starteffect, reverteffect, endeffect
- event options: onStart, onDrag, onEnd
  - each handler function accepts two parameters: the Draggable object, and the mouse event

CSC443: Web Programming

## Drag and drop

14

Scriptaculous provides several objects for supporting drag-and-drop functionality:

- [Draggable](#) : an element that can be dragged
- [Draggables](#) : manages all Draggable objects on the page
- [Droppables](#) : elements on which a Draggable can be dropped
- [Sortable](#) : a list of items that can be reordered
- [Puzzle Game demo](#)

CSC443: Web Programming

## Draggable Example

16

```
<div id="draggableDemo1">Draggable demo. Default options.</div>
<div id="draggableDemo2">Draggable demo.
{snap: [40,40], revert: true}</div>
```

HTML

```
document.observe("dom:loaded", function() {
    new Draggable("draggableDemo1");
    new Draggable("draggableDemo2", {revert: true,
    snap: [40, 40]});
});
```

JS

scriptaculous  
Draggable demo.  
Default options.

scriptaculous  
Draggable demo.  
{snap:[60, 60],  
revert:true}

CSC443: Web Programming

# Draggables

17

- a global helper for accessing/managing all Draggable objects on a page
- **properties:** drags, observers
- **methods:** register, unregister, activate, deactivate, updateDrag, endDrag, keyPress, addObserver, removeObserver, notify

CSC443: Web Programming

## Draggable Example

19

```


<div id="droptarget"></div>
```

HTML

```
document.observe("dom:loaded", function() {
    new Draggable("product1");
    new Draggable("product2");
    Droppables.add("droptarget", {onDrop:
productDrop});
});
function productDrop(drag, drop, event) {
    alert("You dropped " + drag.id);
}
```

JS

CSC443: Web Programming

# Droppables

18

```
Droppables.add(element or id,
    { options }
);
```

JS

- To make an element react when a **Draggable** is dropped onto it, you'll add it to the *Droppables* of the page
- **options:** accept, containment, hoverclass, overlap, greedy
- **event options:** onHover, onDrop
  - each callback accepts three parameters: the **Draggable**, the **Droppable**, and the event

CSC443: Web Programming

## Sortable

20

```
Sortable.create(element or id of list,
    { options }
);
```

JS

- specifies a list (ul, ol) as being able to be dragged into any order
- implemented internally using Draggables and Droppables
- **options:** tag, only, overlap, constraint, containment, format, handle, hoverclass, ghosting, dropOnEmpty, scroll, scrollSensitivity, scrollSpeed, tree, treeTag
- to make a list un-sortable again, call **Sortable.destroy** on it

CSC443: Web Programming

## Sortable demo

21

```
<ol id="simpsons">
  <li id="simpsons_0">Homer</li>
  <li id="simpsons_1">Marge</li>
  <li id="simpsons_2">Bart</li>
  <li id="simpsons_3">Lisa</li>
  <li id="simpsons_4">Maggie</li>
</ol>
```

HTML

```
document.observe("dom:loaded", function() {
  Sortable.create("simpsons");
});
```

JS

CSC443: Web Programming

## Sortable demo

22

event	description
onChange	when any list item hovers over a new position while dragging
onUpdate	when a list item is dropped into a new position (more useful)

```
document.observe("dom:loaded", function() {
  Sortable.create("simpsons", {
    onUpdate: listUpdate
  });
});
```

JS

CSC443: Web Programming

## Sortable list events example

23

```
document.observe("dom:loaded", function() {
  Sortable.create("simpsons", {
    onUpdate: listUpdate
  });
});
function listUpdate(list) {
  // can do anything I want here; effects, an Ajax
  request, etc.
  list.shake();
}
```

JS

CSC443: Web Programming

## Auto-completing text fields

24

- Scriptaculous offers ways to make a text box that auto-completes based on prefix strings:
  - Autocompleter.Local: auto-completes from an array of choices
  - Ajax.Autocompleter: fetches and displays list of choices using Ajax

*ajax autocompletion demo*

To:

Ada Noel  
ada@noel.fake

Adlai Cathy  
adlai@cathy.fake

Adrian Audrey  
adrian@audrey.fake

Adrian Clyde  
adrian@clyde.fake

Adrian Ramneek  
adrian@ramneek.fake

Adrienne Amos  
adrienne@amos.fake

Adrienne Conrad  
adrienne@conrad.fake

Agatha Lesley  
agatha@lesley.fake

CSC443: W

## Using Autocompleter.Local

25

```
new Autocompleter.Local(  
    element or id of text box,  
    element or id of div to show completions,  
    array of choices,  
    { options }  
);
```

JS

- you must create an (initially empty) div to store the auto-completion matches
  - ▣ it will be inserted as a ul that you can style with CSS
  - ▣ the user can select items by pressing Up/Down arrows; selected item is given a class of selected
- pass the choices as an array of strings
- pass any extra options as a fourth parameter between { }
  - ▣ **options:** choices, partialSearch, fullSearch, partialChars, ignoreCase

CSC443: Web Programming

## Using Autocompleter.Local

26

```
<input id="bands70s" size="40" type="text" />  
<div id="bandlistarea"></div>
```

HTML

```
document.observe("dom:loaded", function() {  
    new Autocompleter.Local(  
        "bands70s",  
        "bandlistarea",  
        ["ABBA", "AC/DC", "Aerosmith", "America",  
        "Bay City Rollers", ...],  
        {}  
    );  
});
```

JS

CSC443: Web Programming

## Using Autocompleter.Local

27

```
<input id="bands70s" size="40" type="text" />  
<div id="bandlistarea"></div>
```

HTML

```
#bandlistarea {  
    border: 2px solid gray;  
}  
/* 'selected' class is given to the autocomplete item  
currently chosen */  
#bandlistarea .selected {  
    background-color: pink;  
}
```

CSS

CSC443: Web Programming

## Using Ajax.Autocompleter

28

```
new Ajax.Autocompleter(  
    element or id of text box,  
    element or id of div to show completions,  
    url,  
    { options }  
);
```

JS

- when you have too many choices to hold them all in an array, you can instead fetch subsets of choices from the server using Ajax
- instead of passing choices as an array, pass a URL from which to fetch them
  - ▣ the choices are sent back from the server as an HTML ul with li elements in it
- **options:** paramName, tokens, frequency, minChars, indicator, updateElement, afterUpdateElement, callback, parameters

CSC443: Web Programming

## Playing sounds (API)

29

method	description
Sound.play("url");	plays a sound/music file
Sound.disable();	stops future sounds from playing (doesn't mute any sound in progress)
Sound.enable();	re-enables sounds to be playable after a call to Sound.disable()

```
Sound.play("music/java_rap.mp3");  
Sound.play("music/wazzaaaaaaap.wav");
```

PHP

- to silence a sound playing in progress, use  
Sound.play('', {replace: true});
- cannot play sounds from a local computer (must be uploaded to a web site)

CSC443: Web Programming

## Ajax.InPlaceEditor

30

```
new Ajax.InPlaceEditor(element or id,  
    url,  
    { options }  
);
```

JS

- **options:** okButton, okText, cancelLink, cancelText, savingText, clickToEditText, formId, externalControl, rows, onComplete, onFailure, cols, size, highlightcolor, highlightendcolor, formClassName, hoverClassName, loadTextURL, loadingText, callback, submitOnBlur, ajaxOptions
- **event options:** onEnterHover, onLeaveHover, onEnterEditMode, onLeaveEditMode

CSC443: Web Programming

## Ajax.InPlaceEditor

31

```
new Ajax.InPlaceCollectionEditor(element or id,  
    url,  
    {  
        collection: array of choices,  
        options  
    }  
);
```

JS

- a variation of Ajax.InPlaceEditor that gives a collection of choices
- requires collection option whose value is an array of strings to choose from
- all other options are the same as Ajax.InPlaceEditor

CSC443: Web Programming

## Ajax.InPlaceEditor

32

- **slider control:**

```
new Control.Slider("id of knob", "id of track",  
    {options});
```

JS

- **Builder** - convenience class to replace document.createElement:

```
var img = Builder.node("img", {  
    src: "images/lolcat.jpg",  
    width: 100, height: 100,  
    alt: "I can haz Scriptaculous?"  
});  
$("main").appendChild(img);
```

JS

- **Tabbed UIs**

CSC443: Web Programming