

# 1 Introduction to CSS3

Client: Intuit (Mt. View Office);  
Instructor: Denise R. Jacobs;  
Company: Knowledge United;  
Dates: 22-23 September 2011;  
Day: 2;

## 2 Agenda

### 3 Agenda: Day 2

#### **Hands-on Lab time**

- Work on example page as needed

#### **More CSS3 Properties**

- Advanced Visual Effects
  - border-image
  - multiple backgrounds
  - background-size
  - reflection
  - masks
  - *Hands-on Lab*
- Layout
  - Multi-column layout
  - *Hands-on Lab*

### 4 Agenda: Day 2

#### **CSS3 Properties (cont'd)**

- 2D Transformations
  - rotate
  - scale
  - skew
  - translate
  - *Hands-on Lab*
- Animation
  - transitions
  - animation and keyframes
  - *Hands-on Lab*

### 5 Agenda: Day 2

#### **Responsive Design**

- @media queries for responsive design

#### **CSS3 Selectors**

- Selectors
  - General Sibling E~F
  - Attribute presence
  - Attribute substrings
  - *Hands-on Lab*

- Pseudo-elements
  - (now use :: instead of :)

## 6 Agenda: Day 2

- Structural
  - :nth-child(n)
  - :nth-last-child(n)
  - :nth-of-type(n)
  - :nth-last-of-type(n)
  - :last-child
  - :first-of-type
  - :last-of-type
  - :only-child
  - :only-of-type
  - :empty
- Hands-on Lab

## 7 Agenda: Day 2

- Pseudo-classes
  - Target
    - :target
  - Negation
    - :not(s)
  - State
    - :enabled
    - :disabled
    - :checked
    - :indeterminate
- Hands-on Lab

## 8 Agenda: Day 2

### **Hands-on Lab time**

- Continue to work on in class page example
- Students share work

### **Closing Information**

- Final thoughts
- Students fill-out feedback form

## 9 Resources: CSS3

### **General**

- <http://www.delicious.com/denisejacobs/css3+css3training>

### **CSS3 Best Practices**

- <http://www.webdesignerwall.com/trends/css3-examples-and-best-practices/>


- 10  Hands-on Lab Time!
- 11  More CSS3 Properties!
- 12  Advanced Visual Effects in CSS3
- 13  CSS3 for Advanced Visual Presentation
  - 1.border-image
  - 2.multiple background images
  - 3.reflection
  - 4.masks

14  border-image

15  border-image

16  border-image


```
div.note div.border {  
  -moz-border-image: url(/playground/awesome-overlays/border-image.png) 5 5 5 5  
  stretch;  
  -webkit-border-image: url(/playground/awesome-overlays/border-image.png) 5 5 5 5  
  stretch;  
  border-image: url(/playground/awesome-overlays/border-image.png) 5 5 5 5 stretch;  
}
```

17  Border-image.com

18  Resources: border-image  
[http://www.delicious.com/denisejacobs/  
border-image](http://www.delicious.com/denisejacobs/border-image)

19  multiple backgrounds

20  Multiple backgrounds

21  Multiple backgrounds


```
body {  
  background-color: #5ABBCF;  
  background: #5ABBCF url(..images/bokeh1.png) no-repeat; /* fallback image */  
  background: url(..images/bokeh4.png) no-repeat, url(..images/bokeh3.png) no-repeat 10% 0,  
  url(..images/bokeh2.png) no-repeat 20% 0, url(..images/bokeh1.png) no-repeat,  
  url(..images/glow.png) no-repeat 90% 0,  
  -moz-linear-gradient(0% 90% 90deg,#5ABBCF, #95E0EF);  
  background:  
  url(..images/bokeh4.png) no-repeat, url(..images/bokeh3.png) no-repeat 10% 0,  
  url(..images/bokeh2.png) no-repeat 20% 0, url(..images/bokeh1.png) no-repeat,  
  url(..images/glow.png) no-repeat 90% 0,  
  -webkit-gradient(linear, 0% 0%, 0% 90%, from(#95E0EF), to(#5ABBCF));  
}
```


}


- 22  Resources: multiple backgrounds  
<http://www.delicious.com/denisejacobs/multiplebackgrounds>

- 23  Background-size


- 24  background-size

- 25  background-size
- You can set the size of a background image and make sure it covers the entire background of a page, no matter what the size.


- 26  background-size
- Tips & Issues:
    - Vendor prefixes for mozilla, webkit, and opera
    - IE does not support

- 27  background-size values
- contain**
- Scales the image, while preserving its intrinsic aspect ratio (if any), to the largest size such that both its width and its height can fit inside the background positioning area.

Contain always fits the entire image within your viewport, leaving opaque borders on either the top-bottom or the left-right whenever the ratio of the background image and browser window are not the same.

- 28  background-size values
- cover**
- Scales the image, while preserving its intrinsic aspect ratio (if any), to the smallest size such that both its width and its height can completely cover the background positioning area.

Cover always fills the browser window. You can control how your image is aligned within the viewport by using the background-position property.

- 29  background-size
- Example:
- ```
body { background: #000 url(myBackground_1280x960.jpg)
center center fixed no-repeat;
-moz-background-size: cover;
-webkit-background-size: cover;
-o-background-size: cover; background-size: cover;
}
```

- 30  Resources: background-size
- <http://www.delicious.com/denisejacobs/backgroundsize>

- <http://www.alistapart.com/articles/supersize-that-background-please/>
- <http://www.w3.org/TR/css3-background/#the-background-size>

31  masks

32  masks

33  masks

- Masks allow you to overlay the content of a box with a pattern that can be used to knock out portions of that box in the final display.
- Masks work ONLY in Webkit browsers.
- You can use other images, gradients, and

34  mask types

New mask properties:

```
-webkit-mask (background)
-webkit-mask-attachment (background-attachment)
-webkit-mask-clip (background-clip)
-webkit-mask-origin (background-origin)
-webkit-mask-image (background-image)
-webkit-mask-repeat (background-repeat)
-webkit-mask-composite (background-composite)
-webkit-mask-box-image (border-image)
```

35  masks

Example code:

Using an image


```

```

Using a gradient

```

```

36  Resources: masks

<http://www.delicious.com/denisejacobs/masks>

37  box-reflect

38  box-reflect

39  box-reflect

- A *reflection* is a replica of the original object with its own specific transform and mask.

- Reflections will update automatically as the source changes. If you hover over links, you'll see the hover effect happen in the reflection.
- The reflection will have no effect on layout (other than being part of a container's overflow)

#### 40 box-reflect: syntax

Syntax:

```
-webkit-box-reflect: <direction> <offset> <mask-box-image>
```

<direction> can be one of above, below, left or right.

<offset> is a length or percentage that specifies the distance of the reflection from the edge of the original border box (using the direction specified). It can be omitted, in which case it defaults to 0.

<mask-box-image> is a mask-box-image that can be used to overlay the reflection. If omitted, the reflection has no mask.

#### 41 box-reflect

Example:

```

```

#### 42 Resources: box-reflect

<http://www.delicious.com/denisejacobs/box-reflect>

#### 43 Lab Time

- Let's add border-image, background-size, multiple backgrounds, and box-reflect to our example page.

#### 44 CSS3 Layout Properties

#### 45 multiple columns

#### 46 Multiple text columns

#### 47 Multiple text columns

You can have one div containing a number of paragraphs, with no float or height manipulations.

Example:

```
div.three-col {
  -webkit-column-count: 3;
  -webkit-column-gap: 15px;
  -moz-column-count: 3;
  -moz-column-gap: 15px;
}
```

## 48 Multiple text columns

Tips & Issues:

- The properties are not widely supported, and most of the related (like dividers, breakers, etc) haven't been implemented or aren't supported yet either.

## 49 Multiple column properties

- Column-number
- Column-width
- Columns (shorthand)
  - `<'column-width'> || <'column-count'>`

## 50 Column gap & rule properties

- column-gap
- Column-rule-width
- Column-rule-style
- Column-rule-color
- Column-rule (shorthand)
  - `<column-rule-width> || <border-style> || [ <color> | transparent ]`

## 51 Column span

- Can span multiple columns

## 52 Resources: multiple columns

<http://www.delicious.com/denisejacobs/multiplecolumns>

## 53 Lab Time

- Let's add multiple columns and @media queries to our page.

## 54 2d Transformations with CSS3

## 55 transform (2d)

## 56 2D Transformations

- 1.transform
  - rotate
  - scale
  - skew
  - translate
  - matrix

## 57 transform

- Tips & issues
  - Mozilla, webkit, and opera vendor prefixes; no standard yet.
- Browser Support
  - IE does not support, Opera 10.5 only

## 58 transform/rotate: Syntax breakdown

The generic syntax for transform is

```
<-prefix->transform: type(<value>) type(<value>) type(<value>)  
type(<value>);
```

For rotate specifically, here is the syntax:

```
<-prefix->transform: rotate(<value>)
```

Positive values will rotate the object clockwise to the right, and negative values will rotate the element counter-clockwise to the left.

## 59 transform: rotate

```
#photos img {  
-webkit-transform: rotate(-2deg);  
-moz-transform: rotate(-2deg);  
-o-transform: rotate(-2deg);  
}
```

## 60 Transform: graceful degradation

- Leave images/elements in standard orientation or shape
- There is an IE filter: matrix.
- Extra credit: serve images or image sprites with conditional comments

## 61 Transform: graceful degradation In modern browsers

## 62 transform/rotate: full solution

```
.rotate {  
-moz-transform: rotate(-5deg);  
-webkit-transform: rotate(-5deg);  
-o-transform: rotate(-5deg);  
transform: rotate(-5deg);  
filter: progid:DXImageTransform.Microsoft.Matrix(sizingMethod='auto expand',  
M11=0.9961946980917455, M12=0.08715574274765817, M21=-0.08715574274765817,  
M22=0.9961946980917455);  
-ms-filter: "progid:DXImageTransform.Microsoft.Matrix(M11=0.9961946980917455,  
M12=0.08715574274765817, M21=-0.08715574274765817, M22=0.9961946980917455, sizingMethod='auto  
expand')";  
zoom: 1;  
}
```

## 63 transform/scale: Syntax breakdown

The generic syntax for transform is

```
<-prefix->transform: type(<value>) type(<value>) type(<value>)
type(<value>;
```

For scale specifically, here is the syntax:

```
<-prefix->transform: scale(<width,height>)
```

Negative numbers will flip the element.

Example:

```
#scale {
transform:scale(1,0.5);
}
```

#### 64 transform/skew: Syntax breakdown

The generic syntax for transform is

```
<-prefix->transform: type(<value>) type(<value>) type(<value>)
type(<value>;
```

For skew specifically, here is the syntax:

```
<-prefix->transform: skew(<x-axis,y-axis>)
```

Example:

```
#skew {
transform: skew(35deg);
}
```

#### 65 transform/translate: Syntax breakdown

The generic syntax for transform is

```
<-prefix->transform: type(<value>) type(<value>) type(<value>)
type(<value>;
```

For translate specifically, here is the syntax:

```
<-prefix->transform: translate(<x-value, y-value>)
```

Example:

```
#translate {
transform:translate(10px, 20px);
}
```

#### 66 Transform: multiple

You can apply multiple transform properties to one element.

Example:

```
.enlargen:hover {
-webkit-transform: translate(-50%, -50%) scale(2) rotate(0);
-moz-transform: translate(-50%, -50%) scale(2) rotate(0);
-o-transform: translate(-50%, -50%) scale(2) rotate(0);
transform: translate(-50%, -50%) scale(2) rotate(0);
}
```

#### 67 Tools: transform

- <http://westciv.com/tools/transforms/index.html>

- <http://css3generator.com/>

- <http://css3please.com/>

## 68 Resources: 2D transforms

<http://www.delicious.com/denisejacobs/transform>

## 69 Lab Time

- Let's add transform to our example page.

## 70 Animation with CSS3

## 71 Animation

- 1.transitions
- 2.animation (the webkits only)

A Tip:

Be subtle – it's more effective

## 72 transition

## 73 transitions

You can create subtle transitions between hover states on elements. The transitions smooth out visual jumps.

## 74 transition

Example:

```
#id_of_element {  
-webkit-transition: all 1s ease-in-out;  
-moz-transition: all 1s ease-in-out;  
-o-transition: all 1s ease-in-out;  
transition: all 1s ease-in-out;  
}
```

## 75 Animation

## 76 animation

```
div {  
animation-name: diagonal-slide;  
animation-duration: 5s;  
animation-iteration-count: 10;  
}
```

```
@keyframes diagonal-slide {  
  from { left: 0; top: 0; }  
  to { left: 100px; top: 100px; }  
}
```

## 77 ☐ Lab Time

- Let's add a transition and animation to our example page.
- Make sure you view in either Safari or Google Chrome.

## 78 ☐ Resources: Animation

<http://www.delicious.com/denisejacobs/animation>

## 79 ☐ Responsive Design

### 80 ☐ Becoming Responsive

### 81 ☐ @media queries for responsive design

### 82 ☐ @media queries

@media queries are now being used as a basis for responsive web design: web interfaces that change with the size (and orientation) of the device.

### 83 ☐ Responsive Devices?

### 84 ☐ Responsive Devices!

### 85 ☐ Brain shift

### 86 ☐ Mobile first

### 87 ☐ Avoid this

desktop stylesheet + media queries  
= mobile site

### 88 ☐ ="Switchy" layout

### 89 ☐ Instead this

mobile stylesheet + media queries  
= desktop site

90  = Truly responsive

91  Watch out

- Switchy vs. responsive – keyhole effect
- Be aware of context – you'll probably want different content for different devices

92  3 components

93  Flexible grid

94  Flexible & Fluid

- Size everything in ems or percentages
- Flexible: Ems for everything
- Fluid: Percentages for width, ems for height

95  Various grids

- <http://delicious.com/denisejacobs/grids>

96  Flexible images

97  Flexible Images: Foreground

```
img {width: 100%;  
    max-width: 100%;}
```

98  Flexible Images: Background

Use background-position to selectively crop your backgrounds

99  @media queries

100  @media queries

@media queries are now being used as a basis for responsive web design: web interfaces that change with the size (and orientation) of the device.

101  How do they work?

Through media queries, the browser is served different styles or stylesheets based on the dimensions and the device.

The @media construct allows style sheet rules for various media in the same style sheet.

An @media rule specifies the target media types (separated by commas) of a set of statements (delimited by curly braces).

102  Simple @media rule examples

```
@media print {  
    body { font-size: 10pt }  
}
```

```

    @media screen {
        body { font-size: 13px }
    }

    @media screen, print {
        body { line-height: 1.2 }
    }

```

### 103 Syntax: @media queries

**Syntax:**

```

@media [operand] <media type> [operand]
<(media feature: value)>
{
    properties
}

```

### 104 @media queries

**Example:**

```

/* Smartphones (portrait and landscape) ----- */
@media only screen
and (min-device-width : 320px)
and (max-device-width : 480px)
{
    /* Styles */
}

```

### 105 Hardboiled's @media queries: Smartphone

```

/* Smartphones (portrait and landscape) ----- */
@media only screen
and (min-device-width : 320px)
and (max-device-width : 480px) {
    /* Styles */
}

/* Smartphones (landscape) ----- */
@media only screen
and (min-width : 321px) {
    /* Styles */
}

/* Smartphones (portrait) ----- */
@media only screen
and (max-width : 320px) {
    /* Styles */
}

```

### 106 Hardboiled's @media queries: iPad

```

/* iPads (portrait and landscape) ----- */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px) {
    /* Styles */
}

/* iPads (landscape) ----- */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (orientation : landscape) {
    /* Styles */
}

/* iPads (portrait) ----- */

```

```

@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (orientation : portrait) {
/* Styles */
}

```

## 107 Hardboiled's @media queries: Other

```

/* Desktops and laptops ----- */
@media only screen
and (min-width : 1224px) {
/* Styles */
}

/* Large screens ----- */
@media only screen
and (min-width : 1824px) {
/* Styles */
}

/* iPhone 4 ----- */
@media
only screen and (-webkit-min-device-pixel-ratio : 1.5),
only screen and (min-device-pixel-ratio : 1.5) {
/* Styles */
}

```

## 108 Steps to making it responsive

### 109 Plan the design(s)

- Need to plan out 3-4 iterations of a page design for each resolution/device instead of just one

### 110 Calculate

- Need to know dimensions of page elements to be able to calculate proportional relationship

## 111 The Golden Formula

**target ÷ context = result**

## 112 @media queries: Browser compatibility

1

**Yep**

- IE 9
- Opera 9.5+
- Opera Mobile
- Safari 3+
- Firefox 3.5+
- Chrome

2


**Nope**

- IE 8, 7, 6

- Safari 2
- Firefox 1, 2

113  **Css3-mediaqueries.js**


<http://code.google.com/p/css3-mediaqueries-js/>

114  **Inspiration: responsive design**


- **Mediaqueri.es**

115  **Tools: @media queries**

- <http://www.netmagazine.com/features/21-top-tools-responsive-web-design>
- <http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>

116  **Resources: @media queries**

- <http://www.delicious.com/denisejacobs/mediaqueries>

117  **CSS3 Selectors!**

118  **HTML Hierarchy Reminder**

119  **The Hierarchy “Family Tree”**

- An **ancestor** is any element that is connected to other elements but is higher up the document tree, no matter how many levels up. For example, in the document above, both the <html> and <body> tags are ancestors of the <p> tag.
- A **descendant** is any element connected to an ancestor, but lower in the document tree, no matter how many levels down. In our example, the <em>, <a>, and <img> are descendants of the <body> tag.
- A **parent** is an element directly above a connected element in the document tree. A parent element is also an ancestor, but an element can have ancestors that are not its parents.
- A **child** element is directly below a connected element. A child is a descendant, but an element can have descendants that are not its children.
- **Sibling** elements share the same parent, and are on the same level as each other in the hierarchy.

120  **CSS3 Advanced Selectors**

121  **CSS3 Selector Specification**

**1**

- **General sibling**  
E ~ F
- **Attribute presence**
  - a[attribute="value"]
  - a[attribute~="value"]
  - a[attribute|= "value"]

- Attribute substrings
  - `a[attribute^="value"]`
  - `a[attribute$="value"]`
  - `a[attribute*="value"]`

2

- Pseudo-elements\*

*\*all pseudo-elements indicated with :: in CSS3*

## 122 CSS3 Selectors

1

- Pseudo-classes

- Target
  - `:target`

- Negation
  - `:not(s)`

- State
  - `:enabled`
  - `:disabled`
  - `:checked`
  - `:indeterminate`

2

- Structural
  - `:nth-child(n)`
  - `:nth-last-child(n)`
  - `:nth-of-type(n)`
  - `:nth-last-of-type(n)`
  - `:last-child`
  - `:first-of-type`
  - `:last-of-type`
  - `:only-child`
  - `:only-of-type`
  - `:empty`

## 123 CSS3 Selector Support

### 124 Combinator Selectors: General Sibling

A general sibling selector (also known as adjacent) selects an element that shares the same parent as another element in the document tree.

Syntax:

```
sibling selector ~ sibling selector {property: value; }
E ~ F
```

Example:

```
p ~ dd {font-family: Georgia, serif;}
```

### 125 Combinator Selectors: Attribute presence

**`selector[attribute="value"]`** targets a selector with a particular attribute and specific value.

Example:

```
img[src="catchathief.jpg"] {border: 1px solid #999; }
```

**`selector [attribute~="value"]`** targets a selector with a particular attribute and value where the value is

one of multiple values separated by a space.

Example

```
img[alt~="CSI"] {border: 1px #ff8000 solid;}
```

## 126 Combinator Selectors: Attribute presence

**selector [attribute]="value"** targets an element with an attribute that contains values separated by a hyphen in a list.

Example:

```
img[title]="large" {border: 1px solid #000; }
```

## 127 Combinator Selectors: Attribute substrings

**selector [attribute^="value"]** targets an element with an attribute that begins with a prefix of "value".

Example:

```
img[title^="th_"] { border: 1px solid #000; }
```

**selector [attribute\$="value"]** targets an element with an attribute which ends with a suffix of "value."

Example:

```
img[title$=".png"] { border: 1px solid #000; }
```

## 128 Combinator Selectors: Attribute substrings

**selector [attribute\*="value"]** targets an element with an attribute that contains "value" as any part of a value string.

Example:

```
img[title*="large"] {border: 1px solid #000; }
```

## 129 Lab Time

Let's apply the general sibling and attribute selectors to our page.

Let's see how we can make the page look exactly the same but with the new CSS3 selectors.

## 130 CSS3 Pseudo-classes

### 131 Pseudo-classes: Structural

- Structural
  - :root
  - :nth-child(n)
  - :nth-last-child(n)
  - :nth-of-type(n)
  - :nth-last-of-type(n)
  - :last-child
  - :first-of-type
  - :last-of-type
  - :only-child

– :only-of-type

## 132 Nth child selector tester

## 133 The :nth-child() pseudo-class

The :nth-child() pseudo-class targets an element that has a certain number of siblings before it in the document tree. This argument, which is placed within the parentheses, can be a number, a keyword, or a formula.

A **number** *x* matches the *x*-th child.

Example:

```
• p:nth-child(3) { color:#f00; }
```

## 134 The :nth-child() pseudo-class

The keywords **odd** and **even** can be used to match child elements whose index is odd or even. The index of an element's first child is 1, so this rule will match any **p** element that is the first, third, fifth, and so on, child of its parent element.

Example:

```
p:nth-child(odd) { color:#f00; }
```

The formula **an + b** can be used to create more complex repeating patterns. In the formula, **a** represents a cycle size, **n** is a counter starting at 0, and **b** represents an offset value. All values are integers.

Example:

```
p:nth-child(3n+1) { color:#f00; }
```

## 135 The :nth-last-child() pseudo-class

The :nth-last-child() pseudo-class works just like the :nth-child() pseudo-class, except that it targets an element that has a certain number of siblings **after** it in the document tree.

In other words, it starts counting from the last child instead of the first, and counts backwards.

Example:

```
tr:nth-last-child(2) { background:#ff0; }
```

## 136 The :nth-of-type() pseudo-class

The :nth-of-type() pseudo-class works exactly like the :nth-child() pseudo-class, but only counts those elements that are of the same type as the element the rule is applied to.

Example:

```
p:nth-of-type(3) { background:#ff0; }
```

## 137 The :nth-last-of-type() pseudo-class

The :nth-last-of-type() pseudo-class targets an element that has a certain number of siblings of the same element type **after** it in the document tree.

Just like the :nth-last-child() pseudo-class, it starts counting from the last child instead of the first, and counts backwards.

Example:

```
p:nth-last-of-type(2){background:#ff0;}
```

### 138 The :last-child pseudo-class

The :last-child pseudo-class targets an element that is the last child of its parent element.

It is the same as :nth-last-child(1).

Example:

```
p:last-child {background:#ff0;}
```

### 139 The :first-of-type pseudo-class

The :first-of-type pseudo-class targets an element that is the first sibling of its type. it is the same as :nth-of-type(1).

Example:

```
p:first-of-type { background:#ff0;}
```

### 140 The :last-of-type pseudo-class

The :last-of-type pseudo-class targets an element that is the last sibling of its type. it is the same as :nth-last-of-type(1).

Example

```
p:last-of-type{background:#ff0;}
```

### 141 The :only-of-type pseudo-class

The :only-of-type pseudo-class targets an element whose parent element has no other children of the same element type.

It is the same (but with a lower specificity) as :first-of-type :last-of-type or :nth-of-type(1):nth-last-of-type(1).

Example:

```
p:only-of-type{background:#ff0;}
```

### 142 The :only-child pseudo-class

The :only-child pseudo-class targets an element whose parent element has no other element children.

It is the same (but with a lower specificity) as :first-child:last-child or :nth-child(1):nth-last-child(1).

Example:

```
p:only-child {background:#ff0;}
```

## 143 Lab Time

Let's apply the CSS3 pseudo-classes selectors to our page.

Let's see how we can make the page look exactly the same but with the new CSS3 selectors.

## 144 Pseudo-classes: State

- :root
- :not
- :target

## 145 The :root pseudo-class

The **:root** pseudo-class targets the document's root element. In HTML, the root element is always the HTML element.

:root actually has a higher specificity than html.

Example:

```
:root { background:#ff0; }
```

## 146 Pseudo-class: Negation

**:not** targets all elements that are not indicated. A very useful pseudo-class.

Example:

```
*:not(img) {text-align: left;}
```

(This would match all elements except images.)

## 147 Pseudo-class: Target

**:target** targets the target of a url on a page – most often used for fragment identifiers.

Example:

```
div.window:target {  
  opacity: 1;  
  z-index: 7;  
}
```

## 148 Pseudo-classes: State

- :enabled
- :disabled
- :checked
- :indeterminate

## 149 Pseudo-elements: State

The **:enabled** and **:disabled** pseudo-classes allow developers to specify the appearance of user interface elements (form controls) that are enabled or disabled, provided that the browser allows styling of form controls.

Example:

```
input[type="text"]:enabled { background:#ffc; }  
input[type="text"]:disabled { background:#ddd; }
```

## 150 The :checked pseudo-class

The **:checked** pseudo-class allows developers to specify the appearance of checked radio and checkbox elements. Again, this is provided that the browser allows styling of form controls.

Example:

```
input:checked {border:1px solid #090;}
```

## 151 CSS3 Selector Helper Script

## 152 Tools: CSS3 Selectors

- <http://www.quirksmode.org/compatibility.html>
- <http://www.evotech.net/blog/2009/02/css-browser-support/>
- <http://www.findmebyip.com/litmus>

## 153 Resources: CSS3 Selectors

- <http://www.delicious.com/denisejacobs/selectors+css3>
- <http://inspectelement.com/tutorials/a-look-at-some-of-the-new-selectors-introduced-in-css3/>
- <http://24ways.org/2009/cleaner-code-with-css3-selectors>

## 154 Final Hands-on Lab Time


## 155 Lab Time

We'll continue to work on the page example(s)

I will come around and help anyone who wants or needs it.

## 156 Resources: Each Other!

- Students share their pages with the class

157  The End?


158  This is just the beginning!

My Delicious links are HUGE compendia of all things related to CSS3, updated as I find new articles, resources and tools!

<http://delicious.com/denisejacobs/css3>


<http://delicious.com/denisejacobs/css3training>


159  Homework/Practice

160  Practice/Play

- Find a page on the Intuit site like the one we did in class, and replace all of the items that you can with CSS3 properties and techniques.

161  Books are not for the old-fashioned...

162  CSS3, hot off the presses!

163  Project-based CSS3

**Stunning CSS3**

by Zoe Mikely Gillenwater

164  CSS3 Condensed and Explained

**CSS3 For Web Designers**

by Dan Cederholm

165  CSS3 and Media Queries

166  The book on Responsive Web Design


**Responsive Web Design**

by Ethan Marcotte

167  Responsive Design and solid CSS

**Handcrafted CSS**

by Dan Cederholm with Ethan Marcotte

168  A Compendium of Flexible Layouts

**Flexible Web Design**

by Zoe Mickley Gillenwater

169  Timeless CSS Solutions

**CSS Mastery**

by Andy Budd, et al

- 170 ☐ Proactive coding against bugs  
and graceful degradation

**CssDetectiveGuide.com**

- 171 ☐ Holistic Web Design

**Interact with Web Standards: A Holistic Approach to Web Design**

- 172 ☐ Last but not least...

- 173 ☐ I hope you had a little of this:

- 174 ☐ And that you don't need this

- 175 ☐ I hope you enjoyed the training!  
If you want to follow me on Twitter, I'm  
**@denisejacobs**

- 176 ☐ Thank you for a great class!