

# CSS3: Estelle Weyl

(Video: Introduction.mp4): **Introduction**

- 00:00:00-00:05:51: Estelle gives a brief introduction to the course and a number of resources. Exercise files are incorporated within the slide deck.
  - o Workshop Files: <http://estelle.github.com/CSS-Workshop>
  - o <http://twitter.com/estellevw>
  - o <http://twitter.com/standardista>

## Part 1: Selectors (00:05:52-01:26:34)

(Video: 1\_A.mp4): **Basic selectors, relational selectors**

- 00:05:52-00:08:54: Definition and explanation of both basic and relational selectors
  - o <http://estelle.github.com/selectors/>
- 00:08:55-00:10:50: Demonstration of basic and relational selectors
- 00:10:51-00:12:11: Estelle shows a brief list of the additional selectors that she will be covering and provides a reference for determining browser support
  - o <http://standardista.com>

(Video: 1\_B.mp4): **Attribute selectors**

- 00:12:12-00:20:37: Attribute selectors can be used to style elements based on the existence or value of an attribute.
- 00:20:38-00:25:13: Demonstration of attribute selectors

(Video: 1\_C.mp4): **UI pseudo-classes**

- 00:25:14-00:29:59: UI pseudo classes are based on the current state or structure of the UI. For example, a style may be applied if an item is enabled, disabled or checked.
- 00:30:00-00:33:54: HTML5 web forms also introduce a number of additional pseudo-classes that accompany the new input types.
- 00:33:55-00:37:23: Demonstration of UI selectors and pseudo-classes

(Video: 1\_D.mp4): **Structural UI pseudo-classes**

- 00:37:24-00:43:27: UI pseudo-classes can also target elements based on their relationship to other elements in the DOM. The first step is to understand the difference between *child* and *type*.
- 00:43:28-00:48:18: Nth pseudo-classes target elements based on arguments passed to the selector.

(Video: 1\_E.mp4): **Demonstration of Structural UI pseudo-classes**

- 00:48:19-01:00:35: Estelle demonstrates some techniques using structural pseudo classes to create different column widths and design a US Flag

(Video: 1\_F.mp4): **Negation and empty pseudo-classes**

- 01:00:36-01:06:58: The negation pseudo-class can add exclusion to a CSS rule.
- 01:06:59-01:08:25: The empty pseudo-class, used more rarely, will match any element that doesn't contain any content or whose content is only an HTML comment.

(Video: 1\_G.mp4): **Other pseudo-classes**

- 01:08:26-01:15:13: Estelle finishes up covering a few additional pseudo-classes and demonstrates some uses cases for each one. The additional pseudo elements include :lang, :target, and :root.

(Video: 1\_H.mp4): **Pseudo-elements**

- 01:15:14-01:20:12: While pseudo-classes classes select elements that already exist, pseudo-elements create "faux" elements that you can style. Pseudo elements are often indicated by double-colon notation however for IE support, single colons should be used.
- 01:20:13-01:26:34: It's a good idea to use Selectivizr along with other JavaScript libraries for backward compatibility.
  - o <http://selectivizr.com>

## **Part 2: Specificity (01:26:35-01:37:57)**

(Video: 2\_A.mp4): **Understanding Specificity**

- 01:26:35-01:35:52: Element selectors carry the least "weight" when it comes to specificity. From there, adding classes or pseudo-classes increase the weight of a selector. IDs carry the most weight in specificity. Estelle walks through her "Speci-FISH-ity" chart describing how these rules relate to specificity.
- 01:35:53-01:37:57: Specificity for the newer selectors like the global selector, combinatory, and negation selector are not as obvious. Remember that specificity is not inheritance.

## **Part 3: Specificity (01:37:58-02:34:56)**

(Video: 3\_A.mp4): **Generated content**

- 01:37:58-01:42:05: Generated content is content created with CSS that can be placed before or after the content of an element. The difference is the generated content is not part of the DOM.
- 01:42:06-01:48:34: Essentially you have two stylable elements (<before> and <after>) for every non-empty element. Within the content property, you can put any text, unicode character, attribute, or counter.
  - o <http://coppypastecharacter.com>

(Video: 3\_B.mp4): **Other values for generated content**

- 01:48:35-01:52:47: Incorporating images and counters in the generated content can be useful. Estelle demonstrates a few of these techniques and also describes a use-case for sprite sheets.
- 01:52:48-01:55:26: Counters allow generated content rules to “count” elements in the DOM and display a numeric value.
- 01:55:27-01:57:56: While the attr(X) syntax for generated content is not widely supported yet, in the future it will add some extremely powerful techniques since it can pull from any attribute value within an element.

(Video: 3\_C.mp4): **Use-cases for generated content**

- 01:57:57-02:10:05: Estelle demonstrates some common use-cases for generated content. These include adding iconography and tool-tip hover effects based on href values.

(Video: 3\_D.mp4): **Generated content example by Chris Coyer**

- 02:10:06-02:18:32: There are many more generated content examples on the CSS-Tricks.com website. Some incorporate shapes created with CSS. Others add complex graphical effects or interactive content.
  - o <http://css-tricks.com/examples/ShapesOfCSS/>
  - o <http://css-tricks.com/css-content/>
  - o <http://css-tricks.com/pseudo-element-roundup/>

(Video: 3\_E.mp4): **Generated content code example**

- 02:18:33-02:28:55: Estelle codes a more complex generated content hover interaction.
- 02:28:56-02:34:56: While answering an audience question, Estelle demonstrates how to use the debugger with hover pseudo-classes

## **Part 4: Media Queries (02:34:57-02:48:54)**

(Video: 4\_A.mp4): **Introduction to media queries**

- 02:34:57-02:41:14: Print and screen media queries have been used for many years in web development. With the multi-device landscape, media queries around screen width, height, and orientation are more widely used.
- 02:41:15-02:45:42: Media queries can also be written based on device capabilities.

(Video: 4\_B.mp4): **The viewport**

- 02:45:43-02:48:54: The viewport of a webpage has traditionally been set in an HTML meta element, however this is not appropriate for presentation. CSS adds a “@viewport” declaration that allows the viewport to be set via CSS.
  - o <http://mediaqueri.es>

## Part 5: Debugging (02:34:57-03:02:58)

(Video: 5\_A.mp4): **Debugging in the browser**

- 02:48:55-02:54:29: Most modern browser, both mobile and desktop, have tools for debugging content in-browser. Some browser debuggers are better than others.

(Video: 5\_B.mp4): **Debugging mobile browsers**

- 02:54:30-03:02:58: With mobile devices, it’s helpful to use a remote debugger to inspect the DOM of the content as it’s rendered in the device’s browser. Using Weinre, remote debugging can be done very easily.
  - o <http://people.apache.org/~pmuellr/weinre/docs/latest/>

## Part 6: Colors (03:02:59-03:18:15)

(Video: 6\_A.mp4): **CSS color formats**

- 03:02:59-03:07:54: There are a variety of color formats in CSS include hexadecimal, RGB, RGBA, HSA, and HSLA. The “A” in RGBA and HSLA stands for “alpha” or transparency.
- 03:07:55-03:12:46: HSL stands for hue, saturation, and luminosity. It’s a more intuitive way to select and alter colors as opposed to hexadecimal.

(Video: 6\_B.mp4): **Transparency and color tips**

- 03:12:47-03:16:11: Reducing an object’s opacity will make both the object and its children transparent. It’s often better to use a background color with transparency so the children of the object aren’t affected. Using RGBA or HSLA is often best for shadows.
- 03:16:12-03:18:15: The appearance property in CSS allows developers to change the look of an element. Currently, the appearance property requires vendor prefixes.

## Part 7: Fonts, Shadows, and Text Effects (03:18:16-03:49:36)

(Video: 7\_A.mp4): **CSS Fonts**

- 03:18:16-03:24:45: The font-face rule declares a font family that can be used within a CSS document. It's important to know what font types are compatible in different browsers.

(Video: 7\_B.mp4): **Font Services**

- 03:24:46-03:30:47: Font licensing can be an issue with web fonts. Also, font files are large and can take a while to download. Using a font service like Google Fonts will help with both of these issues.
  - o <http://www.google.com/fonts/>
- 03:30:48-03:35:26: Font Squirrel font-face generator will generate the font files needed for a webpage and optimize the files for the smallest download size
  - o <http://www.fontsquirrel.com/>

(Video: 7\_C.mp4): **Text Shadows and Box Shadows**

- 03:35:27-03:38:19: The key to a good-looking shadow is transparency. Don't use opaque shadows. The four settings for a shadow are left, top, blur and color.
- 03:38:20-03:39:52: Create better looking text-shadows by including multiple shadows. Multiple shadows can be added by separating each text shadow with a comma.
  - o <http://mothereffingtextshadow.com/>
- 03:39:53-03:45:24: Box shadows are similar in syntax to text-shadows, however, they add two additional values: inset and spread.
- 03:45:25-03:46:50: Estelle experiments with generated content and box-shadow.
- 03:46:51-03:49:35: Text stroke in CSS allows you to outline text. However, it currently only works in webkit and can be re-created using text-shadow.
  - o <http://peter.sh/experiments/vendor-prefixed-css-property-overview/>

## Part 8: Borders and Backgrounds (03:49:36-04:29:40)

(Video: 8\_A.mp4): **Background Images**

- 03:49:36-03:53:05: Previously, developers often used the background shortcut in css to declare their backgrounds. Estelle describes why that's not a best practice any more.
- 03:53:06-03:59:31: There are many additional background-image values added in CSS3 including gradients, SVG, and sprites.

(Video: 8\_B.mp4): **Background repeat, attachment and position**

- 03:59:32-04:01:27: Repeated backgrounds are controlled with the background-repeat property. This enables vertical/horizontal tiling or allows developers to turn off repeating.
- 04:01:28-04:05:39: Using background-attachment allows images to be fixed to the background of an element or its parent. If you use percentages to position your background images, the browser uses a ratio for positioning.
- 04:05:40-04:09:12: With the introduction of CSS3, some browsers now support background positioning relative to any corner.

(Video: 8\_C.mp4): **Background clip, origin and size**

- 04:09:13-04:15:05: The background-clip property in CSS indicates which part of the background is showing. Background origin indicates where you the background image begins. When considering hiDPI images, the background-size property determines how backgrounds will grow or shrink.

(Video: 8\_D.mp4): **Border properties**

- 04:15:05-04:20:12: When declaring border properties, they follow the top-right-bottom-left shorthand rules. Unlike backgrounds, it's acceptable to use the border shorthand syntax.
- 04:20:13-04:23:41: Estelle experiments with different border-radius values.

(Video: 8\_E.mp4): **Border images**

- 04:23:42-04:29:40: Border images are use a 9-slice scaling technique where the images is "sliced" into nine regions and the corners maintain their aspect ratio while the other regions shrink/grow.

## **Part 9: Gradients (04:29:41-04:58:56)**

(Video: 9\_A.mp4): **Gradients**

- 04:29:41-04:35:29: Gradients can theoretically be used anywhere an image can be used. There are four gradient types. Currently there are three different syntaxes to support. The first is the original Webkit syntax.

(Video: 9\_B.mp4): **Prefixed, linear gradients**

- 04:35:30-04:41:17: The prefixed, older linear gradient syntax is another version of the gradient syntax that is still supported in web browsers. This syntax requires a vendor prefix

(Video: 9\_C.mp4): **W3C Standard syntax**

- 04:41:18-04:50:22: The final linear gradient syntax is the W3C standard syntax which enables a number of additional techniques like hard-stops, color-stops and endpoints.
  - o <http://standardista.com/cssgradients/>

(Video: 9\_D.mp4): **Radial Gradients**

- 04:50:23-04:50:23: Estelle demonstrates a number of useful techniques for using radial gradients in CSS.

## **Part 10: Transforms (04:58:57-05:15:43)**

(Video: 10\_A.mp4): **Transforming elements**

- 04:58:57-05:09:55: In CSS, there are a number of two-dimensional transform functions that will move, scale, rotate and skew elements.

(Video: 10\_B.mp4): **2D vs. 3D transformations**

- 05:09:56-05:13:39: Estelle begins experimenting with different 2d transformations and explains various customizations like transform-origin
- 05:13:40-05:15:43: The main difference between 2d and 3d transformations is that 3d transformations are rendered by the GPU, not the CPU.

## **Part 11: Transitions & Animations (05:15:44-05:50:33)**

(Video: 11\_A.mp4): **Transition Basics**

- 05:15:44-05:23:12: Transitions can be as simple as changing the color of a link on hover. Adding a duration makes those transitions animate. Any property with an intermediate value can be transitioned.
  - o <http://estelle.github.com/animation/>

(Video: 11\_B.mp4): **Transitions in the real world**

- 05:23:13-05:28:24: Transitions enable web developers to produce effects with CSS that would typically require a lot of JavaScript code to create.

(Video: 11\_C.mp4): **Animation principles**

- 05:28:25-05:34:56: CSS animations can have single or infinite iterations and have more granular control than transitions. They also have a more robust API for hooking into JavaScript. Animations are defined with the @keyframe property.

(Video: 11\_D.mp4): **Attaching animations to elements**

- 05:34:57-05:43:05: Animations are attached to elements using the animation-name property as well as animation-duration property. In addition, the animation-timing-function can be applied to change the way the animation plays.
  - o <http://cubic-bezier.com>

(Video: 11\_E.mp4): **Advanced animation properties**

- 05:43:06-05:50:33: While hard to read, the animation shorthand syntax can make your code less verbose. There are also other properties like animation-fill-mode and animation-delay that give you a more granular control of your animations.

## **Part 12: Other CSS Features (05:50:34-06:15:37)**

(Video: 12\_A.mp4): **Other CSS features, part 1**

- 05:50:34-05:58:45: Estelle quickly covers a number of additional CSS features including multi-column content, box sizing, CSS regions, and CSS exclusions.

(Video: 12\_B.mp4): **Other CSS features, part 2**

- 05:58:46-06:00:53: Estelle continues here brief coverage of additional CSS features that currently are not widely supported but worth noting. They include the new flexible box model, grid layouts, CSS shaders, cursors, text wrapping, the calc method, and rem units

(Video: 12\_C.mp4): **Conclusion**

- 06:10:08-06:15:37: Estelle concludes with a few additional resources and a brief quiz to test your recognition of CSS properties.