# Introduction to CSS, Part 1 - Carrie Dils

## QA

1) When the author describes the benefits of CSS, what does he say CSS controls?

CSS controls the style, presentation, and visual aspects of web pages.

2) [True or False] Styles are applied in the order in which they are found, cascading down from external to local styles.

True.

3) Why do browsers give end users so much control over how the page is viewed?

Because the end goal of any website should be to make content accessible to individual users

4) What 2 parts make up a CSS rule?

A property-value pair separated by a colon and ending in a semi-colon.

5) What separates the property from the value in a CSS rule?

A colon.

6) As described by the author, what type of selector is global in nature?

Element Selector

7) [Critical Thinking] Write a selector that only applies to p tags with a class named blue.

p.blue {property: value;}

8) [True or False] descendant selectors apply to any nested element, no matter how deep it’s found within the page structure.

True

9) [Critical Thinking] Write an example of a group selector.

h1, p, .content {property: value;}

10) [True or False] The author advises to try to structure your code to add meaning to the content, without even thinking of the styling.

True

11) Where are embedded styles typically located?

In the head of an HTML document

12) Why does the author discourage the use of inline styles?

Because they’re not efficient, not scalable, and difficult to overwrite.

13) What is the single sentence the author gives that sums up how rules are applied?

The last rule applied wins

14) [True or False] Inheritance says that the child element will inherit the properties applied to the parent.

True, unless overwritten

15) What does it mean when the author says that styles are cumulative?

Styles applied to broader elements will combine with styles that apply to more specific elements, resulting in multiple rules being affected so long as they aren’t just overwriting one another.

16) What are at least two ways to ensure you CSS looks good everywhere?

Using resources such as <https://caniuse.com> and W3C to ensure that the features you’re using are compatible with browsers.

17) [Critical Thinking] Between normalize and reset stylesheets do you think you would use and why?

I think Normalize would suit my needs best as I might not want to have to overwrite *every* bit of HTML styling right from the get-go.

18) Why does the author specify fallback fonts?

So that in the event that a user agent can’t access the specified font, it has other options to use as substitutes.

19) What do margins represent and define?

The spaces around and between page elements.

20) How many properties does the border have?

3

21) [True or False] Padding is the space added to an element outside the border and is part of the overall width and height of an element's calculation.

False, padding adds space within a border. It does contribute to the overall width and height of an element.

22) [Critical Thinking] The box sizing property accepts two values - which would you use and why?

Probably border-box because content-box performs the same function as width and height, while border-box allows you to manipulate the overall size of an element.

23) [Critical Thinking] How does additional padding effect an element set to a width of 100% inside a parent element?

It will make the child element wider than the parent because padding is applied *outside* of width

24) Which margins potentially collapse, horizontal or vertical?

Vertical

25) [Critical Thinking] Show an example rule that sets all 3 border properties to all 4 borders of an element.

border: 1px solid black

26) When does the outline property provide?

Not sure what this question is asking, but the outline property is similar to a border except that it goes around *all* sides of an element and doesn’t affect the positioning of elements. It’s used primarily for accessibility purposes, like emphasizing links or form fields.

## Video Notes

[What is CSS? (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/what-is-css) - 3:41

* How CSS fits into Web Development
  + CSS: Cascading Style Sheets
    - Controls presentation of web files
    - Separating structure and presentation allows changes to be made to one without affecting the other
      * Centralization of styles allows for quicker development
      * Styles can be easily tailored to varying devices
      * Essential for front-end web dev
    - Style Sheet Language
      * Has it’s own formatting rules
    - Ways of Applying Styles
      * Inline
      * Style Blocks
      * External Style Sheets
    - Single web pages could have multiple style sheets
      * Most specific rule is always used in cases of conflicts
        + Similar to spectrum of Federal, State, and Local laws

[Default Browser Styles (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/default-browser-styles) - 2:47

* “Unstyled” Content
  + Not accurate descriptor
    - HTML files without styling uses browser defaults
  + CSS overwrites default browser styles
    - With browser defaults, it’s easy to miss what hasn’t been styled
    - CSS Reset or Normalize
      * Allows browser defaults to be ignored
    - Users can overrule CSS at any time
      * Goal of any website should be to make content accessible to individual user
    - Each browser has unique rendering engine
      * Same CSS file may vary across browsers
      * Web standards help to minimize these differences nowadays

[CSS Syntax (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/css-syntax) - 3:09

* CSS Syntax
  + Selector and declaration block
    - Selector: Indicates element to be styled
      * Multiple selectors can be grouped together to be edited with a single declaration block
      * p, h {property: value;}
    - Declaration block: Contained within curly braces with one or more declarations (rules)
      * Declaration: property-value pairs separated by colon and ending in semicolon
      * e.g. p: v;
    - Valid properties and values vary for each element
  + Spaces/Whitespace
    - Primarily for readability for coders and don’t matter
    - Whitespaces do matter with selectors though
  + Commenting
    - /\* Comment \*/
  + Additional Syntax Rules
    - Shorthand
    - Pseudo-elements
    - Pseudo-classes
    - Inline style rules

[Basic Selector Types (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/basic-selector-types) - 6:37

* Selectors are used to target HTML elements
* Most common selectors
  + Element selector
    - Global applied, very broad
    - Typically used for site-wide rules
  + Class selector
    - HTML attributes applied to any HTML element
      * Very flexible; can exist multiple times on a page
      * Targeted with a dot selector
      * .class
  + ID selector
    - Represent an HTML attribute like class selectors, but must be unique to a page
      * Identified using the pound selector
      * #idselector {}
      * Very specific scope; can override class styles
    - Both ID and Class selectors can be limited to specific elements
      * h1.page-title {}
      * div#content {}
        + No space between combination selectors
        + Case insensitive
  + Descendant selector
    - Target elements based on location within other elements
      * div p a {}
      * Finds anchor elements within paragraph elements within div element
      * No limit to how many elements can be strung together, but good practice to max at 3
      * Allows for high specificity
        + Can apply to any nested element regardless of how deep it exists within a page structure
  + Grouping Selectors
    - Can be done by separating selectors with commas
      * h1, p, .content {}

[How CSS works with HTML structure (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/how-css-works-with-html) - 4:39

* Best Practices for writing HTML
  + Use meaningful class and ID names
    - Not every line needs an ID or class
    - ID and class names should provide semantic value for scalability
  + Use consistent HTML structure
    - Try to stay consistent across pages to make global styles easier to apply
  + Structure first and design second
    - Don’t think about styling when structuring code

[CSS Authoring Options (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/css-authoring-options) - 2:41

* Styles can be contained in three locations:
  + External stylesheets
    - Text files with .css extension
    - Can hold multiple styles to affect entire site, or sections of a site
    - Most efficient way of styling across a site
  + Embedded styles (exist in head of HTML document)
    - AKA internal style sheet
    - Only exists in singular document; can’t apply global site-wide rules
  + Inline styles
    - Exists within HTML
    - Difficult to overwrite and maintain over time
      * Not scalable or efficient
      * Commonly used for HTML emails

[How Browsers Apply Styles (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/how-browsers-apply-styles) - 5:19

* When a browser loads a page, it displays content and then loads style
  + CSS reads right-to-left
    - #main-content p span {}
    - Looks for span within p within #main-content
  + Typical order of application: External > Embedded > Inline
    - Last rule applied wins
      * Recent styles overwrite earlier styles
      * If external style is linked *after* embedded style, it’ll overwrite the embedded style
  + Inheritance
    - Child rules override parent rules in cases of conflicts
      * Typically, child elements inherit rules of parents
      * If it didn’t work this way, every single element would need to be individually defined (very repetitive)
  + Specificity
    - If Cascade or Inheritance can’t decide in a conflict, specificity wins
      * Each CSS rule has weight score based on elements of selector
        + ID = 100
        + Class = 10
        + Element = 1

#content p .alert {} = 111

#content .alert {} = 110

p .alert {} = 11

.alert = 10

p = 1

* + Styles are Cumulative
    - Styles are the sum of specific selector rules and inherited rules

[Checking Browser Support (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/checking-browser-support) - 4:16

* Check W3C site to know which CSS is currently recommended and what browsers support it
  + <https://caniuse.com> is another tool that is useful for checking compatibility
    - Properties can be searched to see which browsers support it
      * Even if a browser doesn’t support a feature, fallback features can be used
  + <https://www.quirksmode.org/>
* When to Use:
  + Basic CSS will likely work with all browsers, so above resources not necessary
  + When finalizing site designs
  + Using new or advanced CSS features

***Common CSS concepts***

[CSS normalization and resets (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/css-normalization-and-resets) - 4:35

* CSS Reset are small stylesheets that can be added to projects in order to wipe out browsers’ built-in HTML styling
  + <https://meyerweb.com/eric/tools/css/reset/>
  + HTML5 doctor reset is more updated
  + Tim Murtaugh takes inspiration from HTML5 Doctor and Meyer’s resets
* <https://necolas.github.io/normalize.css/> keeps some useful default
* Normalize and Reset are both good options, but should not be used in conjunction
  + Use whichever fits your needs best

[Working with fonts (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/working-with-fonts) - 4:48

* Basic Font Properties
  + Four most common properties
    - Font family (font-family)
      * font-family: “Times New Roman”, Times, serif;
      * Specifies list of fallback fonts in case one isn’t available
    - Font weight (font-weight)
      * Specifies weight of font
        + 100-900 = Lightest-Heaviest
        + If font-weight selected doesn’t apply to current font, it will default to nearest match
    - Font style (font-style)
      * normal | italic | oblique
        + If style doesn’t exist, browser may try to fake it

In case of italic or oblique, browser make tilt font to mimic style

* + - Font size (font-size)
  + Web fonts
    - External style sheets that need to be imported using one of the following methods
      * @font-face rule
      * @import
        + Not optimal for performance
      * Link to stylesheet
      * Load via JavaScript

[Understanding the box model (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/understanding-the-box-model) - 4:02

* Box model refers to physical properties of an element’s rectangular box
  + Every element has a “box” around it
  + Properties
    - Margin: Space around and between element
      * Not used to calculate full width of an element
    - Borders: Outside edge of element
      * border-width
      * border-style
      * border-color
    - Padding: Space added to element inside border
      * Used to calculate overall width of an element, like borders
    - Width and Height
      * Content width varies from total width of an element
      * Width combined with padding and borders can make elements larger than their parents
    - Box Sizing
      * content-box
        + Calculates width and height using default method of width
      * border-box
        + Border-box allows programmers to control total width of an element

[Margins & Padding (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/margins-and-padding) - 3:41

* Padding holds an element’s contents away from its edge
  + Can be defined from each side (-top, -right, -bottom, -left)
    - Setting one does not zero the other options
    - Shorthand notation
      * padding: 10px 20px 15 px 10px
        + Goes in TRBL order
      * 10px 20px 15px
        + Sets top, left & right, bottom
      * 10px 20px
        + Sets top/bottom, left/right
    - By default most elements are block-level elements (e.g.)
      * Will expand to fit width of parents
      * If width is not defined, padding will be applied and shrink width of content
      * If width is defined as 100%, it will make element wider than element as padding is added *outside* of width
* Margins
  + Represents space between elements
  + Apply outside of element’s edge
  + Does not factor into total width of an element, but affects spacing
  + Most elements have default margin to be accounted for
  + Margin use same syntax as padding
  + Vertical margins collapse
    - Two elements stacked on top of each other will overlap their margins

[Borders & Backgrounds (Links to an external site.)](https://www.linkedin.com/learning/introduction-to-css/borders-and-backgrounds) - 5:49

* Borders define outside width of an element
  + Defined with three properties: -style, -width, -color
  + Has same syntax as padding and margins
  + Border width
    - Borders affect overall width of an element
    - Begins at padding width and extends outward
  + Border styles and colors
    - Solid, dotted, and dashed styles create single flat line
    - Double, groove, ridge, inset, and outside creates dimensional effect that often requires width value of more than one
    - Color is independent of foreground and background colors
* Outline
  + Draws line outside of element
  + Similar to border, but goes around *all* sides unlike borders
  + Not part of box model, doesn’t affect positioning
  + Used for accessibility reasons to emphasize links or form fields
* Backgrounds
  + Backgrounds extend all the way to the inside edge of an element’s border
  + Display behind an element’s content
  + Default style typically has transparent background
  + Can be used tdisplay solid colors, gradients, or even images
  + Background colors lie beneath background images
  + Site title can be moved off-screen to be replaced visually with a logo but remain on the page for the sake of SEO