**CSS VISUAL RULES**

**Review Visual Rules**

Incredible work! You used CSS to alter text and images throughout a website. Throughout this lesson, you learned concepts including:

* CSS declarations are structured into property and value pairs.
* The font-family property defines the typeface of an element.
* font-size controls the size of text displayed.
* font-weight defines how thin or thick text is displayed.
* The text-align property places text in the left, right, or center of its parent container.
* Text can have two different color attributes: color and background-color. color defines the color of the text, while background-color defines the color behind the text.
* CSS can make an element transparent with the opacity property.
* CSS can also set the background of an element to an image with the background-image property.
* The !important flag will override any style, however it should almost never be used, as it is extremely difficult to override.

<https://www.codecademy.com/learn/paths/web-development/tracks/styling-a-website/modules/learn-css-selectors-visual-rules/cheatsheet>

### Link Element <link>

The <link> element is used to link HTML documents to external resources like CSS files. It commonly uses:

* href attribute to specify the URL to the external resource
* rel attribute to specify the relationship of the linked document to the current document
* type attribute to define the type of content being linked

<!-- How to link an external stylesheet with href, rel, and type attributes --> <link href="./path/to/stylesheet/style.css" rel="stylesheet" type="text/css">

### Purpose of CSS

CSS, or Cascading Style Sheets, is a language that is used in combination with HTML that customizes how HTML elements will appear. CSS can define styles and change the layout and design of a sheet.

### Class and ID Selectors

CSS classes can be reusable and applied to many elements. Class selectors are denoted with a period . followed by the class name. CSS ID selectors should be unique and used to style only a single element. ID selectors are denoted with a hash sign # followed by the id name.

/\* Selects all elements with class="column" \*/ .column { } /\* Selects element with id="first-item" \*/ #first-item { }

### Write CSS in Separate Files

CSS code can be written in its own files to keep it separate from the HTML code. The extension for CSS files is **.css**. These can be linked to an HTML file using a <link> tag in the <head> section.

<head> <link href="style.css" type="text/css" rel="stylesheet"> </head>

### Groups of CSS Selectors

Match multiple selectors to the same CSS rule, using a comma-separated list. In this example, the text for both h1 and h2 is set to red.

h1, h2 { color: red; }

### Write CSS in HTML File

CSS code can be written in an HTML file by enclosing the code in <style> tags. Code surrounded by <style> tags will be interpreted as CSS syntax.

<head> <style> h1 { color: blue; } </style> </head>

### Selector Chaining

CSS selectors define the set of elements to which a CSS rule set applies. For instance, to select all <p> elements, the p selector can be used to create style rules.

### !important Rule

The CSS !important rule is used on declarations to override any other declarations for a property and ignore selector specificity. !important rules will ensure that a specific declaration always applies to the matched elements. However, generally it is good to avoid using !important as bad practice.

#column-one { width: 200px !important; } .post-title { color: blue !important; }

### Chaining Selectors

CSS selectors can be chained so that rule sets apply only to elements that match all criteria. For instance, to select <h3> elements that also have the section-heading class, the selector h3.section-heading can be used.

/\* Select h3 elements with the section-heading class \*/ h3.section-heading { color: blue; } /\* Select elements with the section-heading and button class \*/ .section-heading.button { cursor: pointer; }

### CSS Type Selectors

CSS type selectors are used to match all elements of a given type or tag name. Unlike for HTML syntax, we do not include the angle brackets when using type selectors for tag names. When using type selectors, elements are matched regardless of their nesting level in the HTML.

/\* Selects all <p> tags \*/ p { }

### CSS class selectors

The CSS class selector matches elements based on the contents of their class attribute. For selecting elements having calendar-cell as the value of the class attribute, a . needs to be prepended.

.calendar-cell { color: #fff; }

### HTML attributes with multiple values

Some HTML attributes can have multiple attribute values. Multiple attribute values are separated by a space between each attribute.

<div class="value1 value2 value3"></div>

### Inline Styles

CSS styles can be directly added to HTML elements by using the style attribute in the element’s opening tag. Each style declaration is ended with a semicolon. Styles added in this manner are known as inline styles.

<h2 style="text-align: center;">Centered text</h2> <p style="color: blue; font-size: 18px;">Blue, 18-point text</p>

### Selector Specificity

Specificity is a ranking system that is used when there are multiple conflicting property values that point to the same element. When determining which rule to apply, the selector with the highest specificity wins out. The most specific selector type is the ID selector, followed by class selectors, followed by type selectors. In this example, only color: blue will be implemented as it has an ID selector whereas color: red has a type selector.

h1#header { color: blue; } /\* implemented \*/ h1 { color: red; } /\* Not implemented \*/

### Separating HTML code from CSS code

It is common practice to separate content code in HTML files from styling code in CSS files. This can help make the code easier to maintain, by keeping the syntax for each file separate, and any changes to the content or styling can be made in their respective files.

### CSS ID selectors

The CSS ID selector matches elements based on the contents of their id attribute. The values of id attribute should be unique in the entire DOM. For selecting the element having job-title as the value of the id attribute, a # needs to be prepended.

#job-title { font-weight: bold; }

### CSS descendant selector

The CSS descendant selector combinator is used to match elements that are descended from another matched selector. They are denoted by a single space between each selector and the descended selector. All matching elements are selected regardless of the nesting level in the HTML.

div p { } section ol li { }

### CSS declarations

In CSS, a declaration is the key-value pair of a CSS property and its value. CSS declarations are used to set style properties and construct rules to apply to individual or groups of elements. The property name and value are separated by a colon, and the entire declaration must be terminated by a semi-colon.

/\* CSS declaration format: property-name: value; \*/ /\* CSS declarations \*/ text-align: center; color: purple; width: 100px;

### Font Size

The font-size CSS property is used to set text sizes. Font size values can be many different units or types such as pixels.

font-size: 30px;

### Background Color

The background-color CSS property controls the background color of elements.

background-color: blue;

### Opacity

The opacity CSS property can be used to control the transparency of an element. The value of this property ranges from 0 (transparent) to 1 (opaque).

opacity: 0.5;

### Font Weight

The font-weight CSS property can be used to set the weight (boldness) of text. The provided value can be a keyword such as bold or normal.

font-weight: bold;

### Text Align

The text-align CSS property can be used to set the text alignment of inline contents. This property can be set to these values: left, right, or center.

text-align: right;

### CSS Rule Sets

A CSS rule set contains one or more selectors and one or more declarations. The selector(s), which in this example is h1, points to an HTML element. The declaration(s), which in this example are color: blue and text-align: center style the element with a property and value. The rule set is the main building block of a CSS sheet.

h1 { color: blue; text-align: center; }

### Setting foreground text color in CSS

Using the color property, foreground text color of an element can be set in CSS. The value can be a valid color name supported in CSS like green or blue. Also, 3 digit or 6 digit color code like #22f or #2a2aff can be used to set the color.

p { color : #2a2aff ; } span { color : green ; }

### Resource URLs

In CSS, the url() function is used to wrap resource URLs. These can be applied to several properties such as the background-image.

background-image: url("../resources/image.png");

### Background Image

The background-image CSS property sets the background image of an element. An image URL should be provided in the syntax url("moon.jpg") as the value of the property.

background-image: url("nyan-cat.gif");

### Font Family

The font-family CSS property is used to specify the typeface in a rule set. Fonts must be available to the browser to display correctly, either on the computer or linked as a web font. If a font value is not available, browsers will display their default font. When using a multi-word font name, it is best practice to wrap them in quotes.

h2 { font-family: Verdana; } #page-title { font-family: "Courier New"; }

### Color Name Keywords

Color name keywords can be used to set color property values for elements in CSS.

**Project**

**Healthy recipes**

**STYLING A WEBSITE**

Healthy Recipes

Using CSS selectors, you’ll give a recipe website some new style!

If you get stuck during this project or would like to see an experienced developer work through it, click “**Get Help**“ to see a **project walkthrough video**.

### Tasks

**0/7 Complete**

Mark the tasks as complete by checking them off

**1.**

Before you begin, take a look at the site’s structure in **index.html**.

Start by making the image at the top of the page a little smaller. Navigate to **style.css** and write a CSS selector for the img tag.

Within its curly braces, write:

height: 150px;

Try experimenting with the 150 number and observing the results.

Hint

To select the img tag, use the following selector:

img {

}

**2.**

The font size of the recipe description should be larger. In **style.css**, write a CSS selector for the .description class.

Within its curly braces, add the following CSS:

font-size: 20px;

Hint

To select a class named main, the syntax would be:

.main {

}

**3.**

Next, let’s style the cooking time. The element on line 15 of **index.html** has an id attribute of cook-time. Navigate to **style.css** and add a cook-time ID selector.

Inside of its curly braces, write:

font-weight: bold;

Hint

The syntax for an ID selector uses a # before the ID name:

#element-id {

}

**4.**

Now, let’s change the bullet points of the ingredient list to squares instead of circles. Start by writing a selector for the li elements inside of the .ingredients element.

Then, write this inside of its curly braces:

list-style: square;

Hint

To select elements within a parent element, use a space between the selectors.

Be sure to put the outside parent element before the child element:

parent-element child-element {

}

**5.**

Next let’s make the time for each preparation step appear gray. In **style.css**, write a selector for p elements that also have a class of .time.

Then, inside of this selector’s curly braces, write:

color: gray;

Hint

To select tag elements with a specific class, chain the class selector to the end of the element selector with no space between:

element.class {

}

**6.**

At the bottom of the page, there’s a link to the full recipe. Let’s make the link a different color.

Notice that in **index.html**, on line 42, there is a p element with a class of citation, then an a element inside of it with a class of external-link. Navigate to **style.css** and write a selector using external-link class.

Then, add this code inside of the selector’s curly braces:

color: SeaGreen;

Hint

Class selectors use a . before the class name:

.class-name {

}

**7.**

Finally, let’s make the font Helvetica instead of the default Times New Roman. Instead of writing multiple selectors to apply the font-family property, write a selector that applies a font-family attribute to all text at once.

The selector should target the h1, h2, p, and li elements.

To change their font, include this line of code inside the curly braces:

font-family: Helvetica;

Hint

You can add the same styles to many elements by separating the selector targets with commas:

element1, element2, ... {

}

**STYLING A WEBSITE**

Olivia Woodruff Portfolio

In this project, you’ll use your knowledge of CSS visual rules to create rule sets and improve the appearance of a photography portfolio site!

If you get stuck during this project or would like to see an experienced developer work through it, click “**Get Help**“ to see a **project walkthrough video**.

### Tasks

**8/9 Complete**

Mark the tasks as complete by checking them off

**1.**

Look over **index.html** to review the different HTML elements you have to work with, then navigate to **style.css**.

Start by making the header section stand out a bit more. Select the .header element, and make its background color CornflowerBlue by using the background-color property.

Hint

The background-color property’’s syntax looks like:

.class-name {

background-color: CornflowerBlue;

}

**2.**

Now change how the text is aligned in the top .header section.

In your .header rule set, align the text in the center using the text-align property.

Hint

The text-align property can align text to the left, right, or in the center.

**3.**

Next, use CSS to make the paragraph below Olivia’s name have a larger text size.

In **style.css**, select the .about-me element, and set its font-size property to 20px.

Hint

The font-size property follows this syntax:

selector {

font-size: 15px;

}

**4.**

The .about-me paragraph looks a little dark against the light blue background, maybe it would look nice if it blended more with the background.

Within the .about-me selector, use the opacity property to make it 50% transparent.

Hint

To make an element 50% transparent, use a value of 0.5 for the opacity property.

**5.**

In the Projects section, make the section titles bold.

Select the .title elements, and add a font-weight property to make their text bold.

Hint

To bold an element’s text, follow this syntax:

.class-name {

font-weight: bold;

}

**6.**

Change the main title color so that it matches the background color more nicely. Set the color for h1 elements to Azure.

Hint

The syntax to set text color for an element is:

selector {

color: ColorName;

}

**7.**

Instead of the page being in the default Times font, change the font of the entire page.

Select the body element and make the font-family of the page Georgia.

Hint

You can select the body element with:

body {

}

Then, set the font-family property with:

font-family: Georgia;

**8.**

Finally, let’s make the background of the page more interesting.

Within the body selector, set the background-image property to this URL:

https://s3.amazonaws.com/codecademy-content/courses/learn-css-selectors-visual-rules/hypnotize\_bg.png

Hint

When using an image with background-image, you need to put it inside url() like this:

selector {

background-image: url('https://example.com/images/image.png');

}

**9.**

Great work! Feel free to keep coding and edit the visual rules to personalize the site’s appearance!

Cheat sheet:  
<https://discuss.codecademy.com/t/list-of-challenge-projects/505231>