Challenge Questions: CSS

**Done:**View **To do:**Make forum posts: 1

Choose**between 1-3** of the following Challenge Questions to research on your own and answer. You can research all of the questions, but only submit answers to a max of three.

The answers are *not* in your lessons. These questions are intended to have you conduct additional research on your own to gain experience in finding answers for yourself. Don't look for answers in the lessons!

If some of your classmates have already submitted answers, try to choose a question/questions that have not already been answered. If they have all been answered, try to find additional information that has not yet been presented by your classmates.

Challenge Questions

**Question #1:**

* Aside from color names and hex color codes, what is another way to encode a CSS color value? Give an example.
  + You can use the RGB method. RGB stands for red, green, blue. To get the color you want you would specify a quantity (0 to 255) or a percent (0% to 100%) of each of the three colors. A quantity or percent of 0 for all three colors(red, green, blue) would produce black. A quantity of 255 or a 100% of all three colors would produce white. Listed below is how the code for these would look as well as the code for purple. All three examples demonstrate making a paragraph the associated color.
    - Code for black
      * P { color: rgb (0, 0, 0); }
    - Code for white
      * P { color: rgb (255, 255, 255); }
    - Code for purple
      * P { color: rgb (128, 80, 200); }

**Question #2:**

* Hex color codes typically have 6 characters that follow the #, with each set of 2 representing R (red), G (green), or B (blue), such as **#ffffff**. However, these two example hex codes are also valid:

#fff

#ffffff44

Why is this, and what do the two hex codes above mean?

* Both of these are the color white
* The 3 digit RGB notation (#rgb) is converted to 6 six digit form by replicating the digits (#rrggbb). This means #fff (white) replicated is #fffff.
* In 8 digit hex code the last two digits are the alpha channel. These last two digits determine how transparent or opaque the color will be. 00 would be fully transparent. FF would be fully opaque. Thus #ffffff44 is a fairly transparent white.

**Question #3**

* What does this declaration do to an element?

margin: 10px 20px 5px;

* Your top margin will be 10px, right & left margin will be 20px, and bottom margin will be 5pm

**Question #4**

* Two other common units for CSS lengths, aside from **px**, are **em** and **rem**. Briefly describe the meaning of each.

<https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units#lengths>

[**Lengths**](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units#lengths)

The numeric type you will come across most frequently is [<length>](https://developer.mozilla.org/en-US/docs/Web/CSS/length). For example, 10px (pixels) or 30em. There are two types of lengths used in CSS — relative and absolute. It's important to know the difference in order to understand how big things will become.

**Absolute length units**

The following are all **absolute** length units — they are not relative to anything else, and are generally considered to always be the same size.

| **Unit** | **Name** | **Equivalent to** |
| --- | --- | --- |
| cm | Centimeters | 1cm = 37.8px = 25.2/64in |
| mm | Millimeters | 1mm = 1/10th of 1cm |
| Q | Quarter-millimeters | 1Q = 1/40th of 1cm |
| in | Inches | 1in = 2.54cm = 96px |
| pc | Picas | 1pc = 1/6th of 1in |
| pt | Points | 1pt = 1/72nd of 1in |
| px | Pixels | 1px = 1/96th of 1in |

Most of these units are more useful when used for print, rather than screen output. For example, we don't typically use cm (centimeters) on screen. The only value that you will commonly use is px (pixels).

**Relative length units**

Relative length units are relative to something else, perhaps the size of the parent element's font, or the size of the viewport. The benefit of using relative units is that with some careful planning you can make it so the size of text or other elements scales relative to everything else on the page. Some of the most useful units for web development are listed in the table below.

| **Unit** | **Relative to** |
| --- | --- |
| em | Font size of the parent, in the case of typographical properties like [font-size](https://developer.mozilla.org/en-US/docs/Web/CSS/font-size), and font size of the element itself, in the case of other properties like [width](https://developer.mozilla.org/en-US/docs/Web/CSS/width). |
| ex | x-height of the element's font. |
| ch | The advance measure (width) of the glyph "0" of the element's font. |
| rem | Font size of the root element. |
| lh | Line height of the element. |
| rlh | Line height of the root element. When used on the [font-size](https://developer.mozilla.org/en-US/docs/Web/CSS/font-size) or [line-height](https://developer.mozilla.org/en-US/docs/Web/CSS/line-height) properties of the root element, it refers to the properties' initial value. |
| vw | 1% of the viewport's width. |
| vh | 1% of the viewport's height. |
| vmin | 1% of the viewport's smaller dimension. |
| vmax | 1% of the viewport's larger dimension. |
| vb | 1% of the size of the initial containing block in the direction of the root element's [block axis](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Logical_Properties#block_vs._inline). |
| vi | 1% of the size of the initial containing block in the direction of the root element's [inline axis](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Logical_Properties#block_vs._inline). |
| svw, svh | 1% of the [small viewport](https://developer.mozilla.org/en-US/docs/Web/CSS/length#relative_length_units_based_on_viewport)'s width and height, respectively. |
| lvw, lvh | 1% of the [large viewport](https://developer.mozilla.org/en-US/docs/Web/CSS/length#relative_length_units_based_on_viewport)'s width and height, respectively. |
| dvw, dvh | 1% of the [dynamic viewport](https://developer.mozilla.org/en-US/docs/Web/CSS/length#relative_length_units_based_on_viewport)'s width and height, respectively. |

**Question #5**

* How do you use the **list-style**property? Give an example to illustrate.

<https://developer.mozilla.org/en-US/docs/Web/CSS/list-style-type>

**Question #6**

* What is a pseudo-element selector? This is not the same as a pseudo-class selector. Give an example to illustrate.

**Question #7**

* What is the !important rule in CSS, and why should you generally avoid it?