CAA2

Web Standards and Languages

Víctor Sánchez Fuentes

12.11.2023

Techniques for software application development

First part

Question 1:

CSS cascade: The CSS Cascade is the way browsers resolve competing CSS declarations. Every time a CSS declaration (or rule) is written, it will enter the CSS Cascade, which will determine whether or not it will end up as the final style. The further down the cascade a declaration falls, the less likely it will end up as the final style. The tiers that differentiate these levels of preference are:

1. IMPORTANCE:

   The first tier of the Cascade looks at the type of rule we're looking at. Whether it's highlighted by !important and their origin.

   There are four basic types of rules:

   a. transition

      Rules that apply to an active transition take the utmost importance.

   b. !important

      When !important is added to the end of our declaration, it jumps to this level of the Cascade. Ideally, you reserve this level as a last resort resource, when there's a need to override styles from third-party libraries.

   c. animation

      Rules that apply to an active animation jump up a level in the Cascade.

   d. normal

      This level is where the bulk of rules live.

   Ignoring layers, they can be summarized in the following picture:

Order (low to high)	Origin	Importance
1	user-agent (browser)	normal
2	user	normal
3	author (developer)	normal
4	CSS @keyframe animations	
5	author (developer)	`!important`
6	user	`!important`
7	user-agent (browser)	`!important`
8	CSS transitions	

2. Specificity

   The second tier of the Cascade looks at the Specificity of a rule, the weight of a CSS selector is calculated through an algorithm to determine which rule from competing CSS declarations gets applied to an element.

   There are five levels of selectors:

   a. Inline: Styles declared within a style HTML property are the most specific
   b. Layer: Layers "win" by being defined later.
   c. Id: We can target elements based on their id, using the syntax #id
   d. class | attribute | pseudo-class: We can target elements based on their class, using the syntax .class.
      i. This level also includes attribute selectors that target HTML attributes, like [checked] and [href="https://wattenberger.com"]-
      ii. This level also includes pseudo-selectors, like :hover and :first-of-type
   e. type | pseudo-element: We can target elements based on their tag type, using the syntax type.
      i. This level also includes pseudo-elements, like :before and :selection

3. ORDER:

And lastly, the final tier of the Cascade, which looks at the order that the rules were defined in. Last rule defined will be the one applied when conflict arises.

- p a.ref
    - P = element selector = 1 point
    - A.ref = it's an element selector (anchor) with a class selector. 1 point for element and one for class selector

      0-1-2. There is no ID selector. One class selector. Two element selectors.

- #main a:focus
    - #main = id selector = 1 point.
    - A:focus = element selector pointed at a certain state, so it is a pseudo-class,
    - selector = 1 point for element, one point for class.

      1:1:1

- aside .menu li:last-child
    - Aside = element selector pointing at the aside = 1 point element.
    - .menu = class selector = 1 point class.
    - Li:last-child = element selector and pseudo-class selector = 1 point element and 1 point class.

      0-2-2

- section h2 + i
    - Section: element selector = 1 point element.
    - H2 = element selector = 1 point element.
    - I = element selector = 1 point element

      0-0-3.

      Worthy of note is that combinators, such as +, >, ~, " ", and ||, may make a

selector more specific in what is selected but they don't add any value to the specificity weight.

- #news > img[src|="logo"]
  - #news = id selector = 1 point id.
  - img[src|="logo"] = element selector with attribute selector = 1 point element and 1 point class.

    1-1-1

    Worthy of note is that combinators, such as +, >, ~, " ", and ||, may make a selector more specific in what is selected but they don't add any value to the specificity weight.

Question 2. Disabilities table

Visual impairments	Screen readers (such as JAWS, NVDA or VoiceOver)
Hearing impairments	Textual alternatives of different sorts: • Videos should be manually captioned. • Text simplification for those with high levels of language deprivation.
Mobility impairments	Make controls accessible by keyboard.
Cognitive impairments • Depression • Schizophrenia • Dyslexia • Attention deficit hyperactivity disorder	• Delivering content in more than one way, such as by text-to-speech or by video. • Easily understood content, such as text written using plain-language standards. • Focusing attention on important content. • Minimizing distractions, such as unnecessary content or advertisements. • Consistent webpage layout and

	navigation. • Familiar elements, such as underlined links blue when not visited and purple when visited. • Dividing processes into logical, essential steps with progress indicators. • Website authentication as easy as possible without compromising security. • Making forms easy to complete, such as with clear error messages and simple error recovery.

SECOND PART

HTML

index.html

- **“title” tag** is shown to users during search and it needs to be a piece of text that summarizes the content in order to capture the user’s attention. In this case, a piece of the h1 and h2’s text have been used to summarize the page’s content.
- The **favicon.png** is linked in the head with a link tag instead of using an img tag as is used in the header or body.
- The **header** of the web page is a set of introductory content. The **image** logo indicating the organization and a small text indicating what the content of the page is is included in it.
- The header’s text is the h1. It summarizes the content of both pages so it causes no conflict.
- The class active link was created to differentiate the link that needs to “explode”

when hovered and the one that doesn't.

- **h2** indicates a subsection of the web page. Seeing the two pages as a spa, "what is hac te?" is the subsection in charge of explaining it trough the list, images and snippets of press releases.
- **H2-container** was created to manipulate the h2 and the introduction.
- The tag **h3** is used to differentiate each way of explaining hac-te, as each can be considered to have the same importance.
- **Each li element** is divided into two parts: the image that will serve as the bullet point and the text. This was done to make the manipulation of each easier. The class list-display is used later.
- Important to note that the img representing the bullet points are given a null value in alt as they are only decorative. This is a valid exception.
- With the press releases part of the content, three classes have been created to manipulate its elements through css and make the abstraction easier.
- **Blockquote** tag indicates that the enclosed text is an extended quotation. In this case, each one with its cite.
- To achieve the big quotation mark at the beginning a reversed quotation has been added with name code. They are later manipulated through css. They are in a span with its own class so that said manipulation is easier
- School-images and school-images-foot were created to make later css manipulation easier.
- The images are aligned through a span and each image is inside an anchor so that it acts as a hyperlink. The span acts as a container of the text for simplicity.
- In the footer, there are two classes to differentiate the alignment of text. Later this will be addressed through css.
- The <**br**> are break lines.

News.html

- Same overall structure has been used as in the index.html for simplicity. Difference is in the main.
- Similarly as before, h2 has been used to indicate that this is a subsection of the spa in charge of news.
- The class picture-container was created for abstraction and "automatize" the process of creating a container for the images.
- Figure has been used to include a footnote with the image.

CSS

common.css:

In this css file, the rules that can be applied to both html filed, due to sharing some elements, are kept.

- **@import** to import the content of other files or APIs.
- **Varela round and Arvo** font are linked thanks to the **google api**.
- The **:root CSS pseudo-class** matches the root element of a tree representing the document. It has the same effect as the html selector but with higher specificity. It was used precisely because of this to declare the different colors' new name properties.
- The **html** document is given a background color previously defined thanks to the **var** function.
- **Arvo** is applied as the font family throughout the **html** thanks to font family.
- **header has a background image** and only appears in the background. It doesn't have any semantic meaning so there's no need to make it an img and text can be put over it.
- No-repeat has been chosen so that it doesn't repeat constantly in the background.
- Background-size cover scales the image (while preserving its ratio) to the smallest possible size to fill the container (that is: both its height and width completely cover the container), leaving no empty space. Even if the proportions of the background differing from the element may cause the image to be cropped either vertically or horizontally, it was chosen so that no white spaces were left.
- Text shadow gives the text that shadow effect. 1px in the x axis/ 1px in the y axis in the length of the shadow, 2px is the blur radius of the shadow (how big and wide it is) and black is the color of the shadow.
- Text-align centers the text within the header.
- Nav's float property with left value makes sure that the nav is in the higher left part of its container.
- Width has a value in % to establish it by taking its parent's width and applying the percentage to it.
- Nav ul is a selector used to set the property list-style to none, which makes the elements of the list have no bullet point.
- **Next property** selects the list element with class active-link that is being hovered over. It adds changes compared to its regular state and make it animated with the transition-timing-function: ease-in-out. It makes it smoothly go into and out of the

hover state.

- Padding-left value in transition causes that “explosion” to the left effect and 1s makes it happen over a 1 second span.
- In the same property, width and background color have similar effects but cause it to change its width and the color of the background of the list element.
- The border has been given to the main element so that it is maintained consistent and it reaches the footer more easily. It is easier to play with the margin and padding with respect to it, too.
- Of the border-left property, 2px signals the width of the border, solid is the style and lastly is the color. This is a shorthand property.
- **Display-inline** will be used throughout the css files to signal the display style. In this case, the inline-block value displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values.
- The footer's display’s value is block to ensure that it starts on a new line, and takes up the whole width.
- All the different anchors’ possible states in the footer have been selected and given a white color to ensure it stays that way.
- Text-decoration: none; gets rid of the underline that an anchor usually has.
- **Each class in the footer** was given to differentiate the one with the text aligned to the right and the one with it aligned to the left.
- Both classes occupy the 50% of the footer’s width minus the space that their padding and margin takes.
- In the case of the right one, the width of the border was also extracted from the width. That is why the value subtracted to the 50% is higher.

index.css:

- For the class h2-container, in this case a background-image that is a linear gradient was created. This is a function that takes several possible values. In this case, the color the gradient starts with and the percentage of the way it starts transitioning (length wise, it starts transitioning at 10%). 50% is the midpoint of transition. It doesn’t become full on whitish until it reaches the “theoretical” 120% of the length, which is never seen and that’s why the blur is never quite white.
- To achieve the bullet points, the most important values are in the selectors that include the class list-display and the span. The rest are for localizing it.
- First, the vertical-align property sets the vertical alignment of a box. Middle value aligns the middle of the element with the baseline plus half the x-height of the

parent. This makes the boxes with list-display class center within the li.

- The display:table-cell will make it behave like the elements of a table so that they can be manipulated as such.
- Then the images inside the boxes are given the value relative to the position. This makes the element be located within the natural flow of the content but its position can be set through the properties bottom, left, right and top.
- In this case, the image was moved 7px above the bottom line so that it indented into the list.
- Lastly, the text itself was given a line-height of 25px so that there was some space for the image to indent into.
- **The press-releases class** was created to make it clearer where each element selected was located. Press-releases-title was given the inherit value in display so that it inherited it from the parent. Nothing of note is to be discussed here as most of the values and properties have been seen before.
- **The class quotes** are the most important part. The quotes were treated as their own element with decoration. The position: relative has been seen before. Top: 50px pushes the element down 50 pixels. No line-height was given to it so that its background didn't overflow outside of the box. Varela round and the font size are needed to match the screenshot.
- **School-images class** was created for the different images of the part that take part of the group.
- The most important part is the images themselves. The span has a max width and min-width of 24% to ensure that no image is bigger or smaller than that (smaller images might turn out distorted). As there are 4 images the space is distributed more or less evenly.
- Object-fit contain of the images makes the images scale to maintain their aspect ratio while fitting within the element's content box. As the content box has a set width, the images width is going to be kept consistent throughout.
- Min-height has a similar effect as min-width. In this case, it was used to ensure the images height is consistent throughout.

News.css

- .h2-news-container is similar to the one used in index. Html to keep it consistent.
- The insides of **picture container class** box have been centered through text-align.
- Both the picture-container and the picture-container h4 have been given a border radius. 15px radius is given to both the top corners and no radius is given to the bottom ones. The bigger the value, the bigger the border is.

- The images inside the image caption have been given an ellipse effect through the border radius property, too. 50% gives the effect.

Accessibility

Non-text Content is given an alt value so that screen readers can give a description of them.

Distinguishable

- Use of Color:

  Some elements have been changed to black or white to increase the contrast enough so that it's more readable for users with visual impairment.

-

REFERENCES

https://2019.wattenberger.com/blog/css-cascade

https://developer.mozilla.org/

W3schools.com