1. What happens in the background.
2. Three pillars of writing good html and css.  
   Responsive design :   
   Writing maintainable and scalable code :   
   Web Performance :
3. Responsive web design:   
   one website that works beautifully on all screen sizes and all devices.

Here we need to think about responsive images, using correct units for font-sizes or element dimensions, even includes a desktop first or mobile first strategy.

1. Writing maintainable and scalable code:  
   clean, easy to understand, supports future growth and most importantly re-usable.  
   This also includes how to organize files, how to name classes, how to structure html.
2. Web Performance: less code , compress code, less http requests, use css preprocessor,mostly reduces the use of images by using the images which are really necessary for a website + compressing those images so that they consume less bandwidth for the user.
3. HOW CSS WORKS BEHIND THE SCENES.
4. What happens to our css code when we load a webpage.  
   - browser loads html  
   - browser parses html + loads the stylesheets and parses it which is a bit different than  
    how html is parsed.  
   - browser build the DOM like a family tree.
5. Two main steps in CSS parsing phase :   
   8.a : Conflicting css declarations are resolved in a process called cascade.  
   8.b : process final css values like converting a margin defined in percentage units to pixel   
    units.

After all of this is done final css is stored in a tree like structure called the CSS object model.

1. After the html and css are parsed and stored : these two together form the render tree.
2. Then the page is rendered. The browser uses the visual formatting model to render the page.  
     
    ---------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------
3. CSS parsing phase.  
   step 1: cascade : it is the process of combining different stylesheets and resolving conflicts between different css rules and declarations – when more than one rule applies to a css element.  
   CSS can come from different sources.  
   **Developer written CSS**: they are called author declarations.

**User declarations**: CSS coming from the user i.e in the browser.  
**Default browser declarations**: user agent CSS as it is set by the browser.

1. But how does CSS resolve conflicts.

It looks at the IMPORTANCE > *selector* SPECIFICITY > SOURCE ORDER of conflicting declarations in order to determine which one takes precedence.

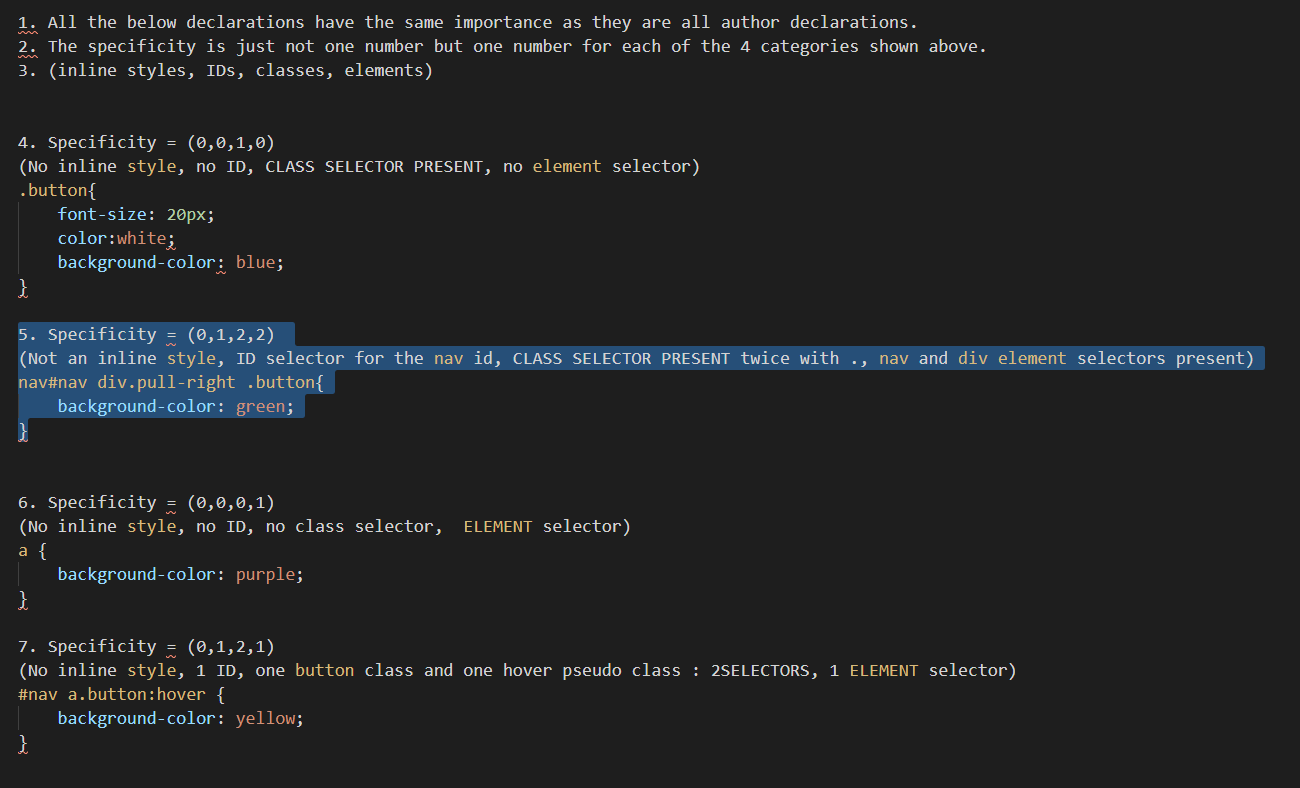
1. First the cascade starts by giving conflicting declarations different **importance** based on where they are declared / based on their source. The order is below

* Most important declarations are the user declarations marked with !important keyword.
* Author declarations marked with an !important keyword.
* Author declarations
* User declarations
* Browser declarations.

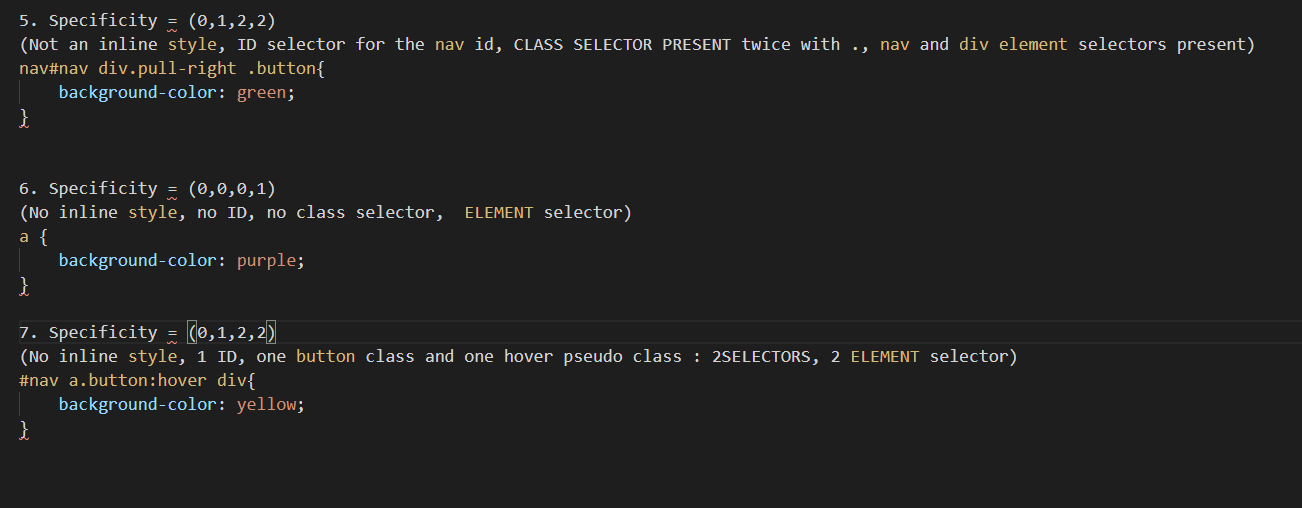
1. When we have conflicting declarations with the same importance we calculate the selector specificity based on the priorities shown below.

Now cascade calculates and compares the specificities of the declaration selectors.

* The specificity order is below.
* Inline styles : they have the highest specificity.
* IDs
* Classes , Pseudo classes, attribute selectors
* Elements, pseudo elements.

Example : How to calculate specificities.  


* Selector number 2 is the most specific selector of all.

1. Now let us say of if point 5 and 7 have the same specificity like below ..i.e if all the declarations have the same specificity.. then the last css declaration written in the code is the one that will apply.( SOURCE ORDER)  
     
    ---------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------