2. Natours Project — Setup and First Steps (Part 1)

3. Building the Header - Part 1

* Every element in html has a default margin and padding that’s why we do reset by globally removing them.
  + Universal selector: \* {margin: 0; padding: 0; box-sizing: border-box}
  + border-box removes the padding and border from the total height from an element.
* Class name on each and every element in html.
* Font is inherited in CSS if we put it in the body.
* For making shape we use clip-path it’s a modern CSS property.
  + clip-path: polygon(100% 0, 100% 100%, 0 50%); side triangle.
  + Point works clock wise.

4. Building the Header - Part 2

* inline element can be separated by display: block. It has a line break.
* transform property is very useful for modern looking effects.
* transform: translateY(); always related to the initial state.
* There are two types of animations-
  1. Transition property.
  2. Keyframe.
* Transition needs to be in the initial state.

5. Creating Cool CSS Animations

* Sometimes we have shaky animation. To solve this problem, we use-
  + backface-visibility: hidden;

7. Building a Complex Animated Button - Part 2

* Sudo element ::after has to have a content, let it be an empty one.
* Sudo element is basically treated as the child of the parent element.
* transition is given to the initial state of an element.

3. How CSS Works A Look Behind the Scenes

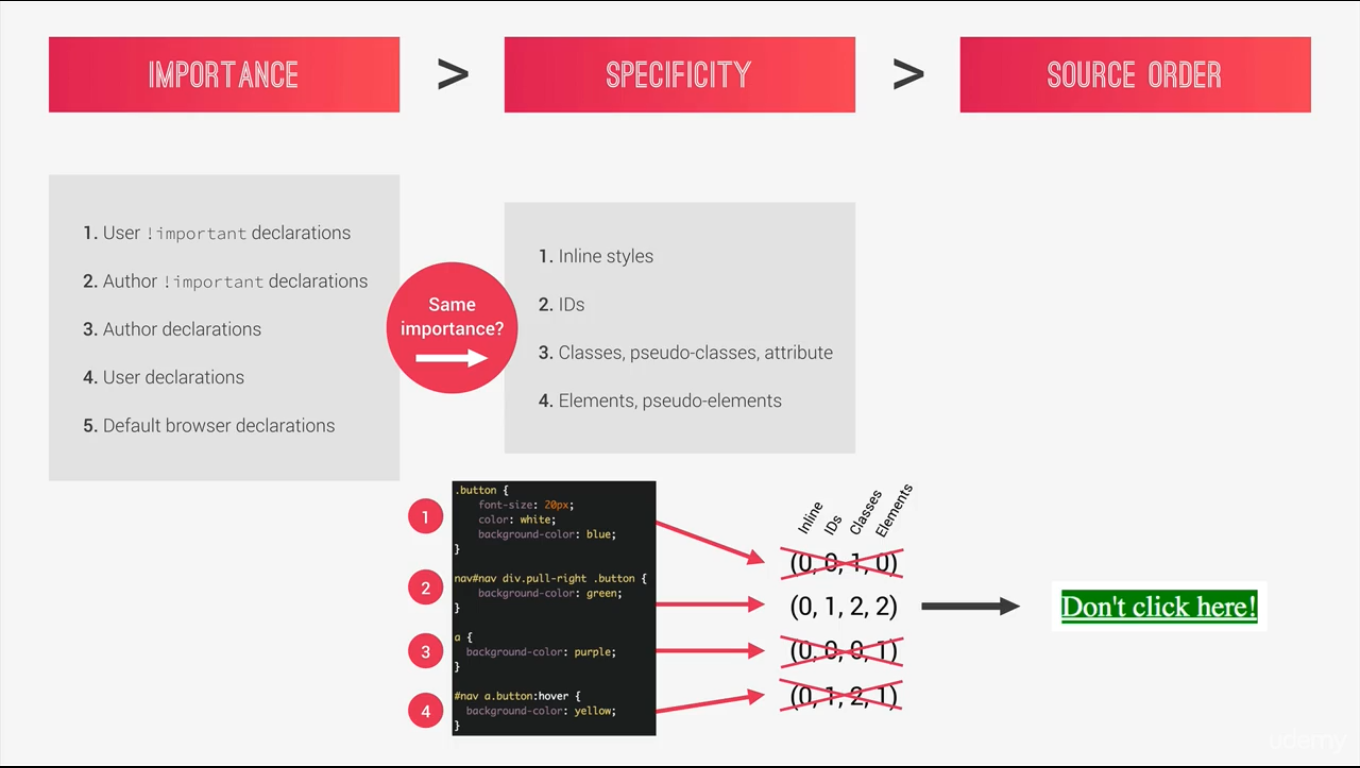
2. Three Pillars of Writing Good HTML and CSS (Never Forget Them!)

1. Responsive Design.
2. Maintainable and Scalable code.
3. Web performance. (less image)

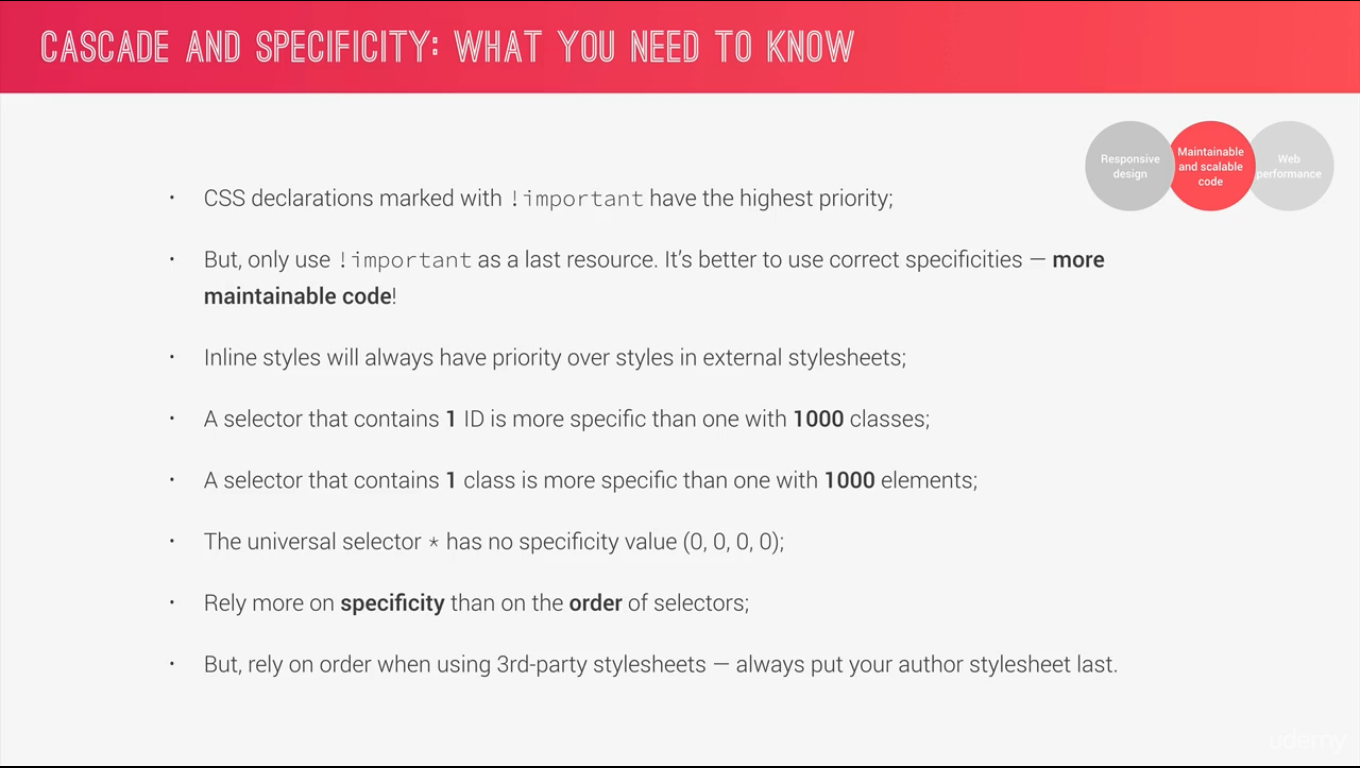
* In parsing CSS phase, the percentage is calculated to px, according to every screen size big or small. It is not calculated beforehand.

4. How CSS is Parsed, Part 1 The Cascade and Specificity





* It is calculated serial-wise. If inline has a bigger value then that is the highest specificity and rest of it is disregarded.
* If specificity is tied then the last declaration in the code will override all other declaration and will be applied.



5. Specificity in Practice

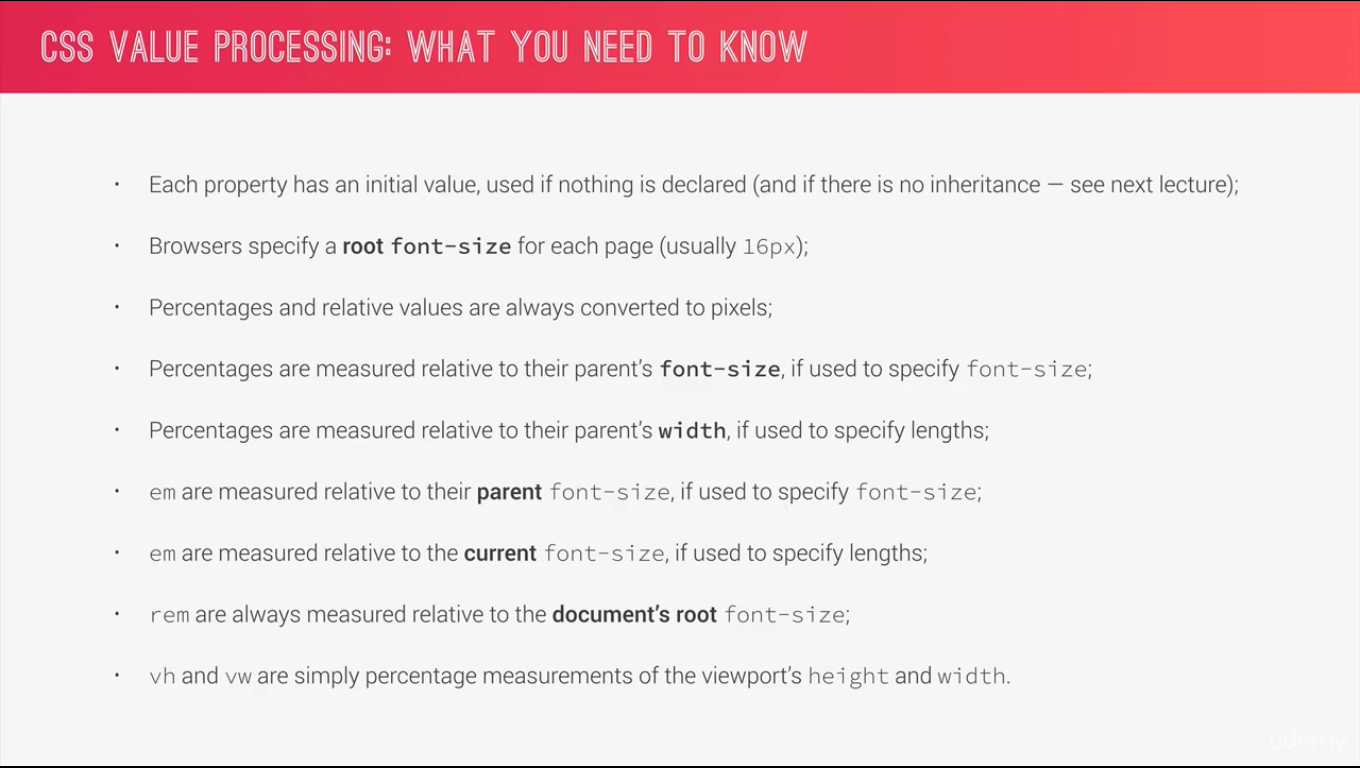
* hover sudo class is also counted for specificity. hover works as having more specificity over initial state in other words it overrides the element.
* hover acts as a normal class.

6. How CSS is Parsed, Part 2 Value Processing

* All the unit is converted to px.
* Padding is used if we declare it or not and it has some inheritance dependency (more on this latter!)

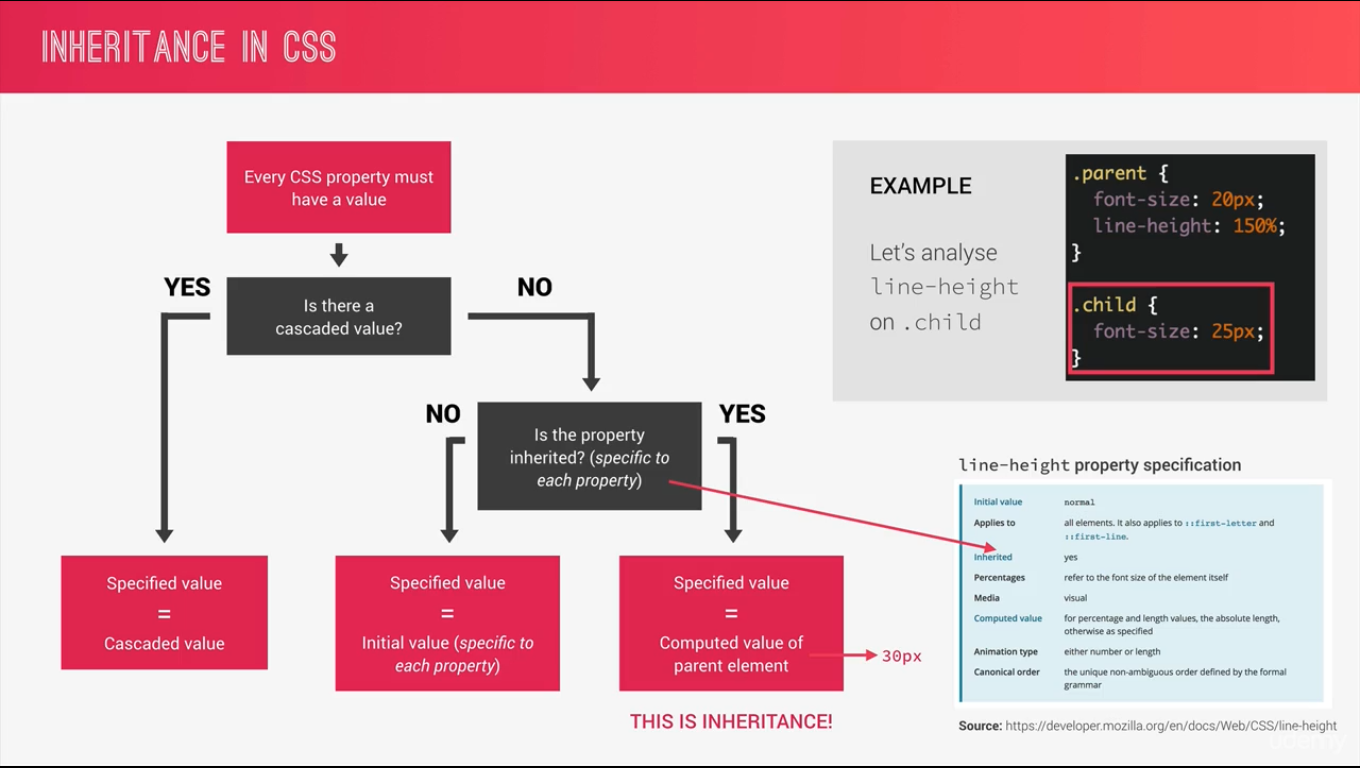


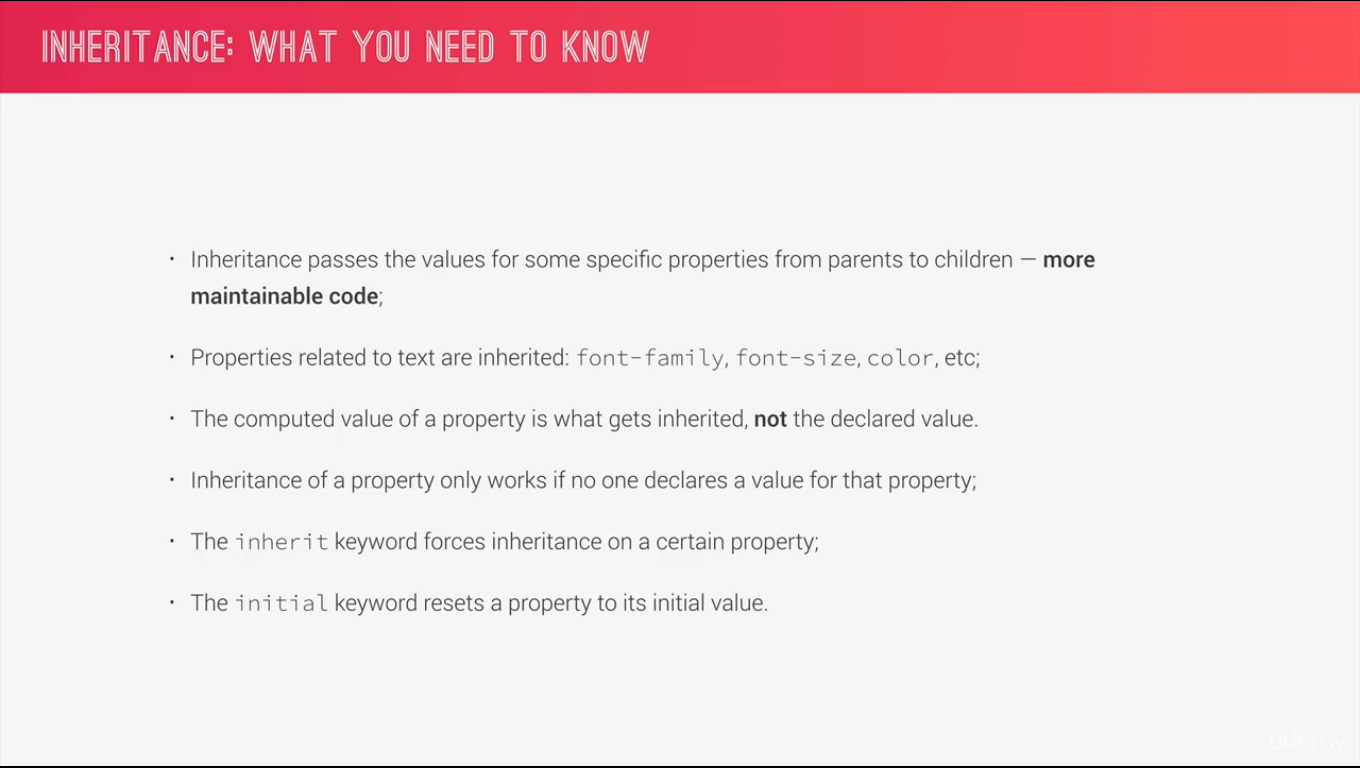
* Parents width is calculated is the reference for percentage-based calculations.
* em uses the current or parent element as a reference while rem uses the root font-size as the reference.
* For length it is different. When we use em for length is calculates the parent element as reference
* rem is used same for both font and length.
* Why use em and rem because if we change the font size then it will automatically change the length. It is a great technique for robust design.
* vh = Viewport height, vw = Viewport width.



7. How CSS is Parsed, Part 3 Inheritance

* For the example that is used in the video. If we use % inside a parent element then the child element will inherit based on the value of parent element. Image below:





8. Converting px to rem An Effective Workflow

* Awesome technique for not writing hundred lines of code in media query is to use rem for font size in the global settings.
* In html tag we give the global font-size and it is 10px for easier calculation.
* Using rem is the best solution rather than em because it is very difficult to maintain as it has dependency with parent child.
* It’s bad practice to use px as font-size. In the global settings if we use px as font-size then any user who has default font-size increased