Sizes and Units:

px: defined and fixed size.

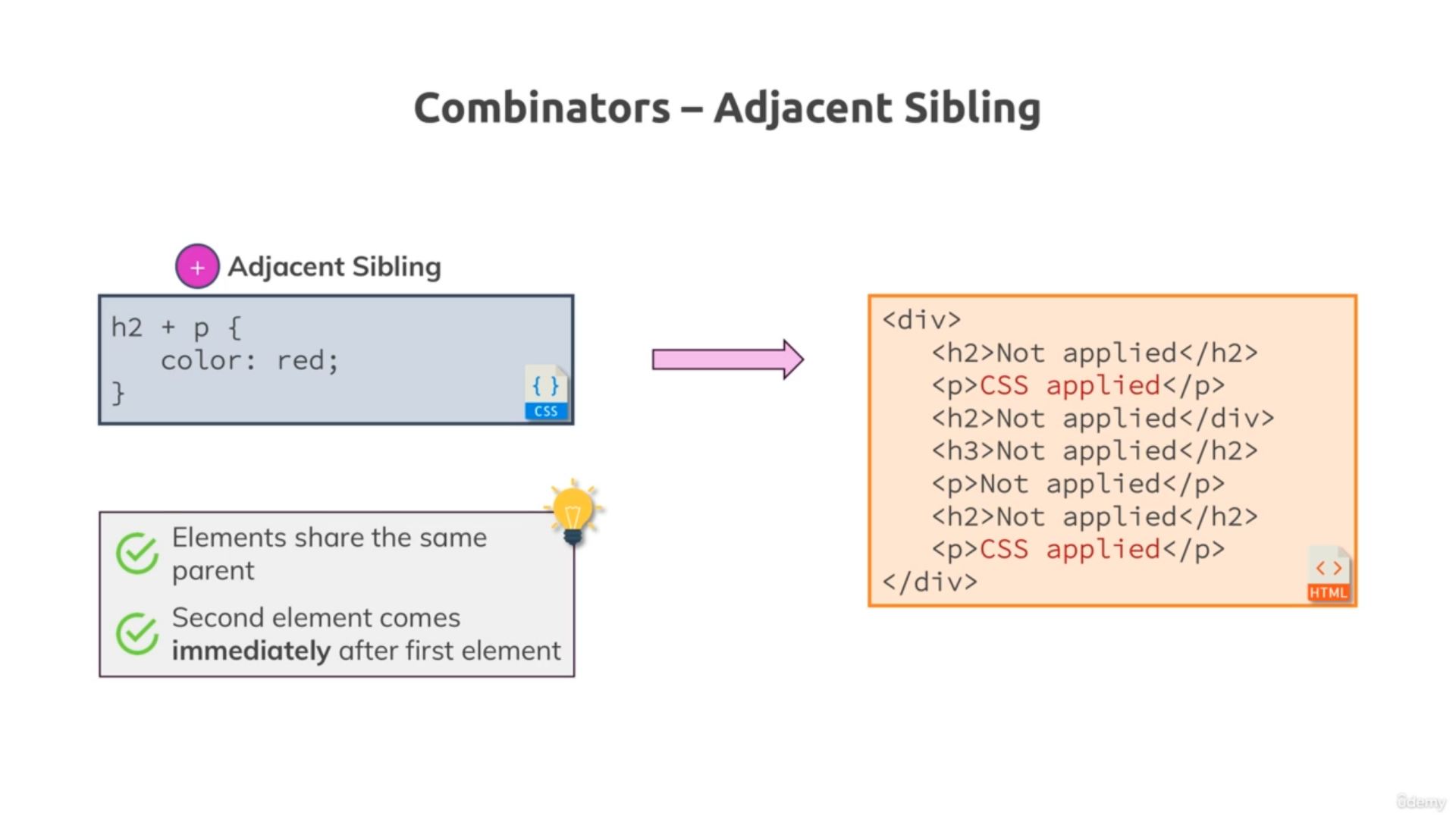
%: depends on its parent

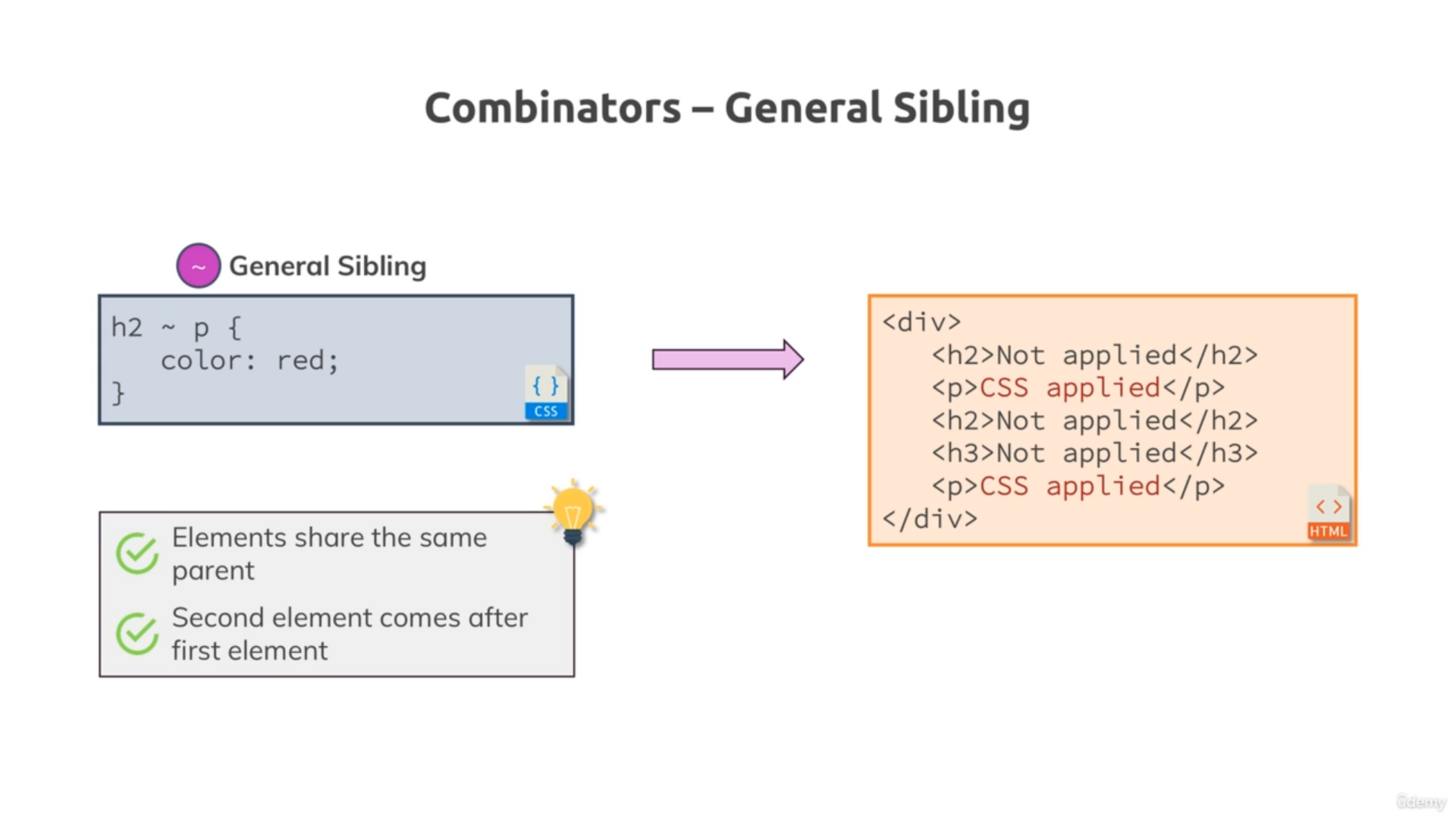
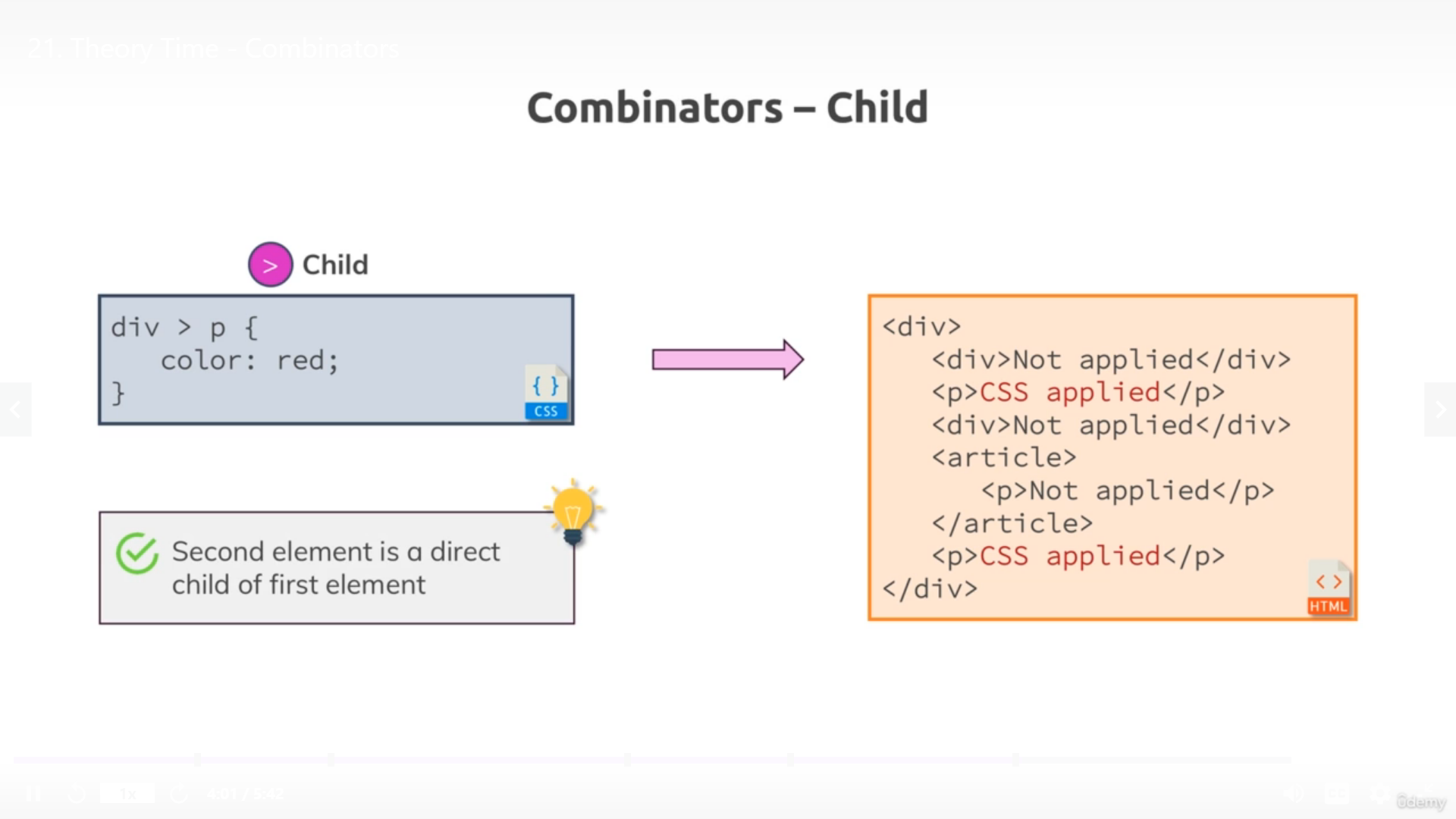
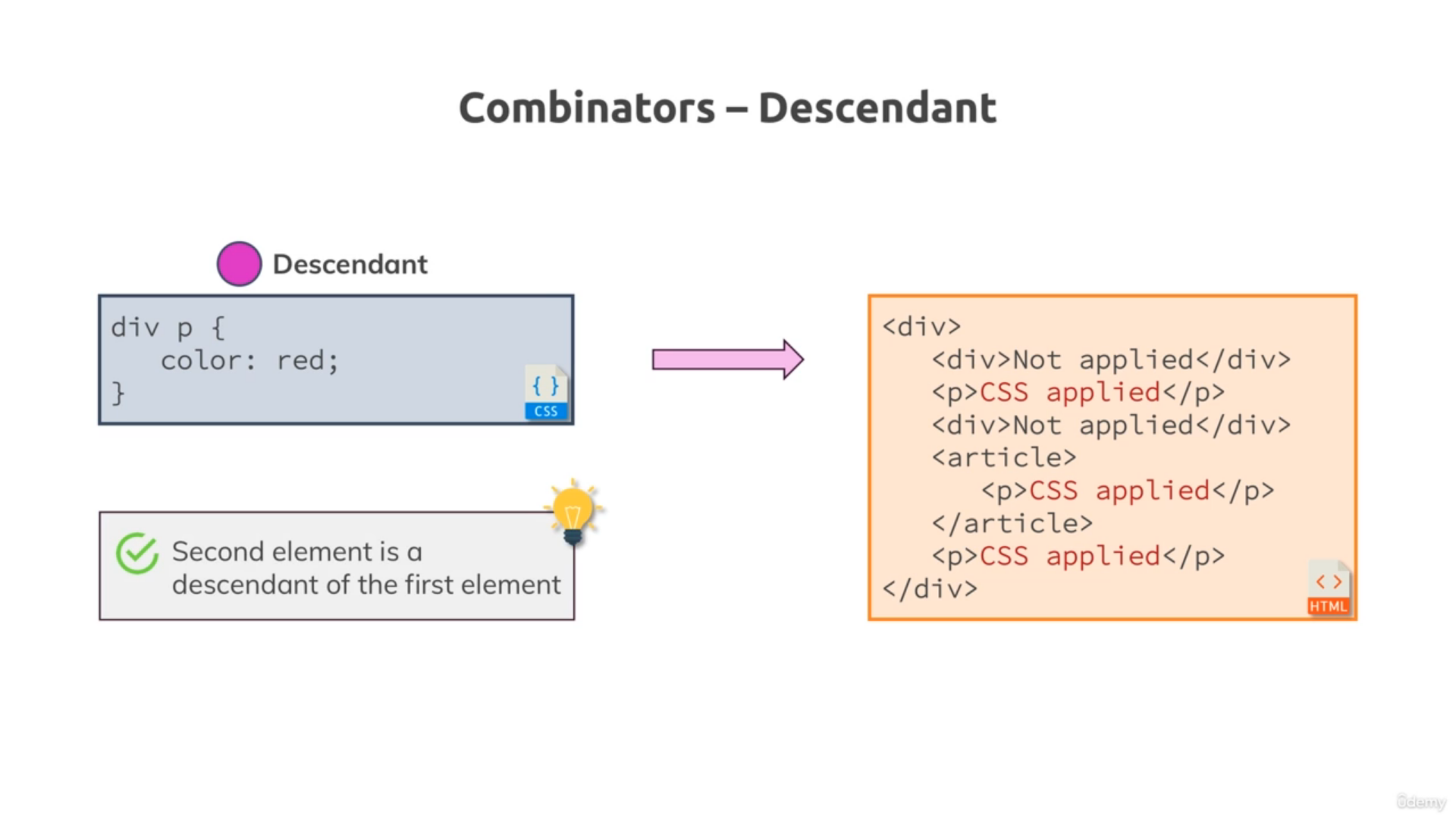
vh, vw : it totally depends on screenSize only, does not depend on parent, e.g 50vh-> 50% of ScreenHeight & 25vw-> 25% of screenWidth.

em: relative to parent font size.

rem: relative to root/browserDefault font size.

Combinators:

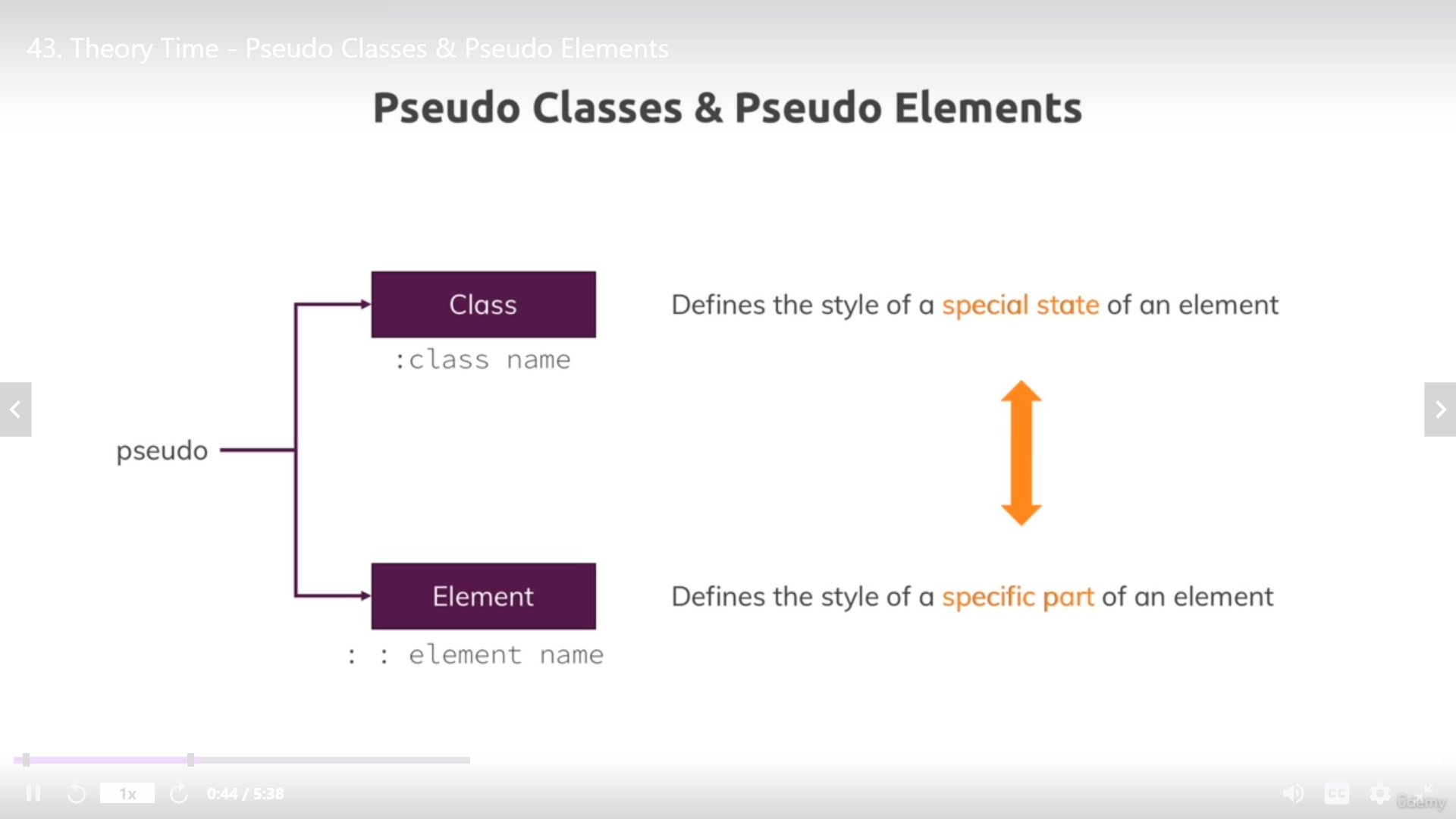


Box Model

Height/width : this will consider element content height/width property having default box-sizing: ‘content-box’ property

While Setting box-sizing: ‘border-box’ this will set height/width property of element considering total content+padding+border



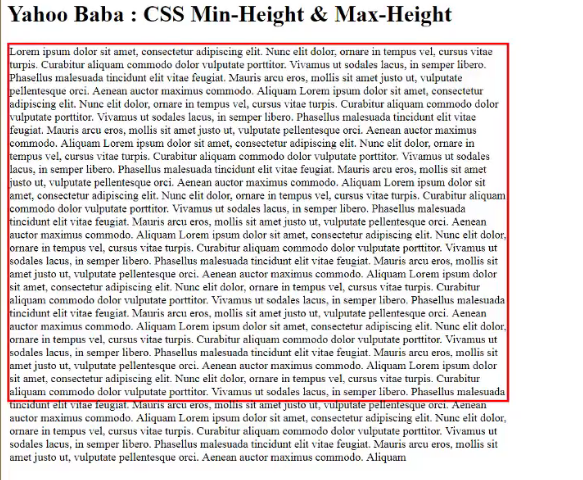
Pseudo element means: element part e.g Newspaper 1st later is capital and bold

Examples:

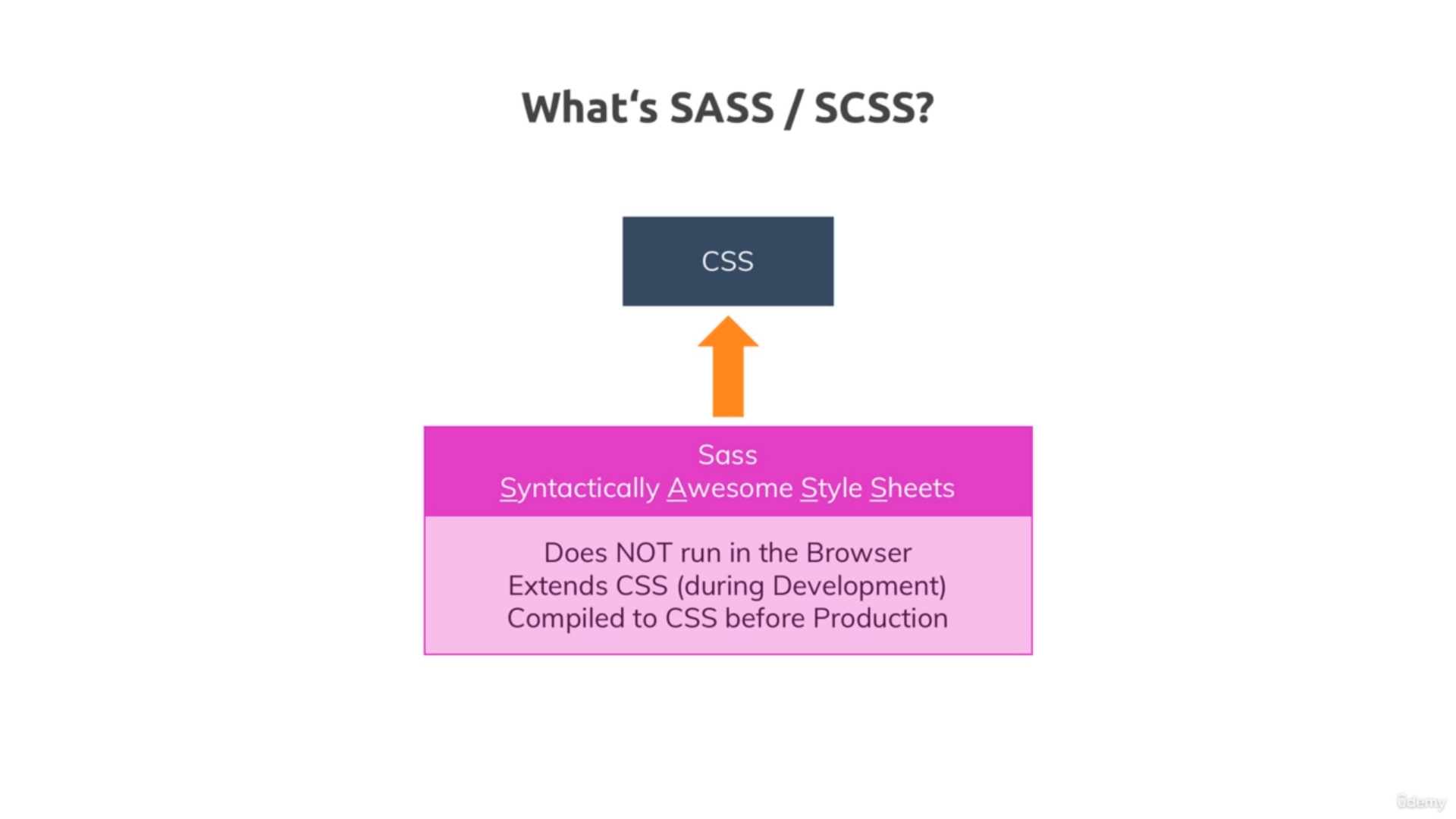
p::firstElement, p::after, p::before

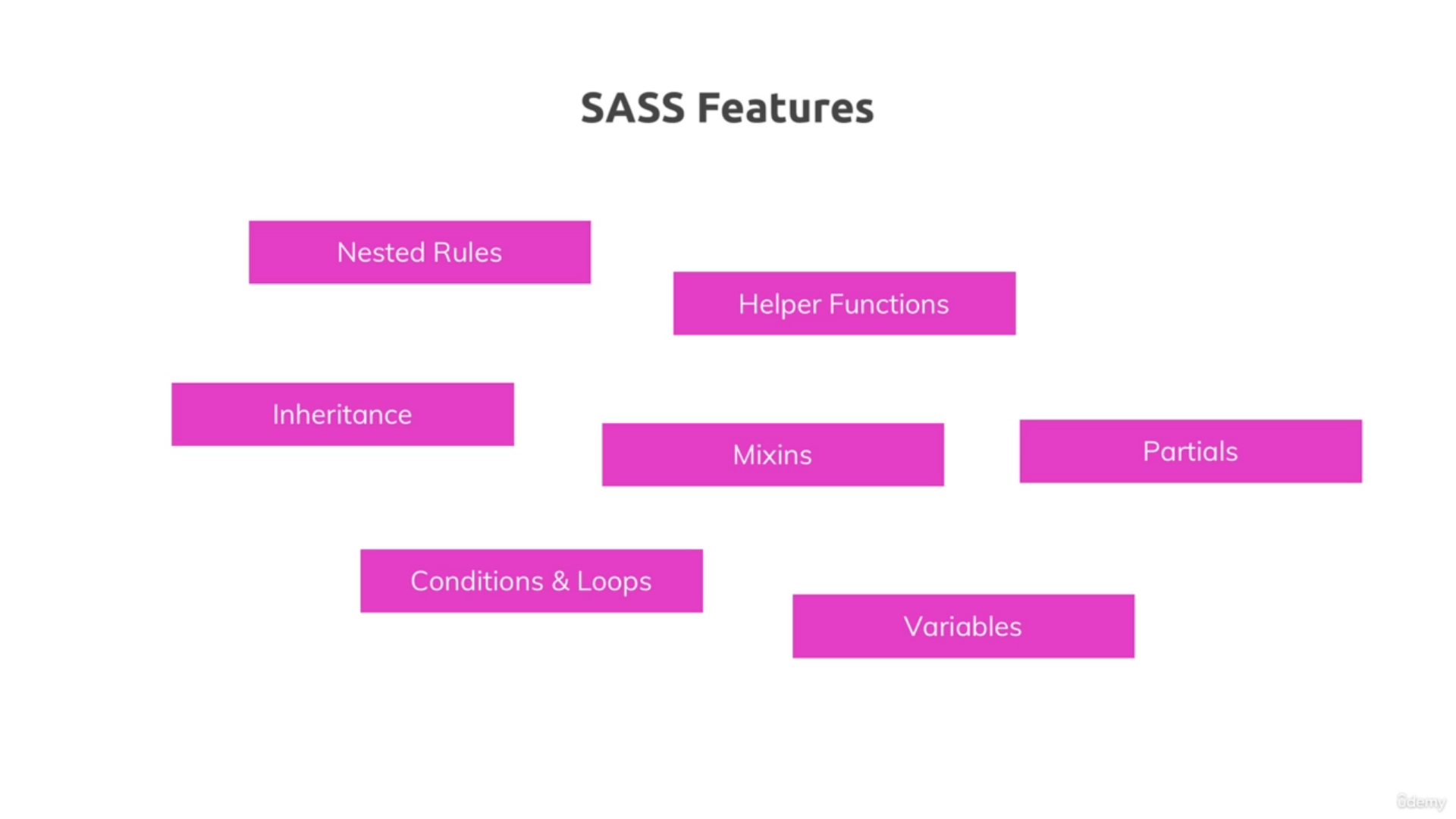
pseudo class examples:

hover, active(hold a click), focus(staying on element), link, visited.



SASS is the superset of CSS, It’s a CSS extension language.basically it helps developer to write css in more easier manner give ability to developer to use features like inheritance, Nesting, variables, functions etc.

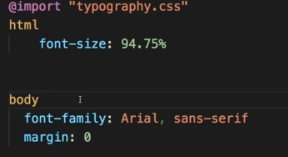




SASS Built on ruby so it requires ruby runtime, so need to Install Ruby on windows.

Difference between SASS and SCSS, both are same but just syntax difference is there.

e.g of SASS:

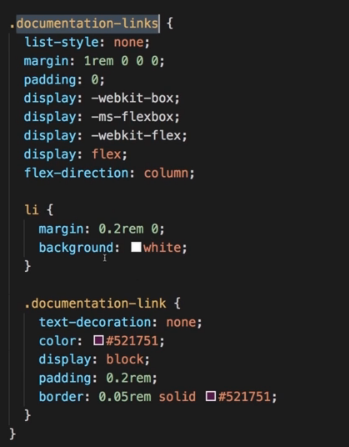


e.g of SCSS, it retain normal CSS syntax



Nesting in SCSS

In normal css its not work but in scss this nesting give more readability.



Now it need to convert back to css, because browser don’t understand Scss, so we use command in terminal “sass –watch main.scss:main.css”

Note it also create main.map.css

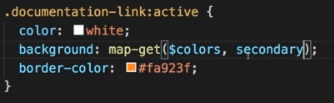
using variable in SCSS:



Properties are stored in variable like this and we can use this variable name anywhere required.



Here colors are stored in map variable, we can use it like:



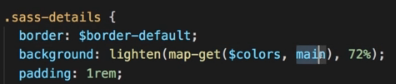
We can also do some arithmetic for e.g



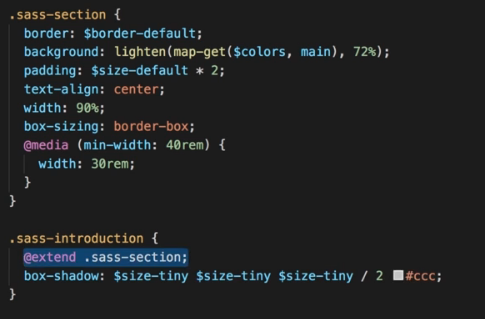


Functions in Scss:

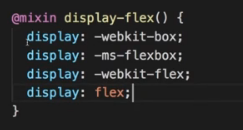
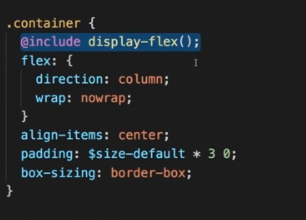
When we want 1 color depends on another color, then we can use functions in SCSS for e.g lighten(dependent color, % we want to lighten)



Inheritance in SASS/SCSS using @extend .className



Mixins: Mixins are reusable custom functions:

It similar to Inheritance we can use repeating code snippet inside class

Margin don’t work for top and bottom for inline element