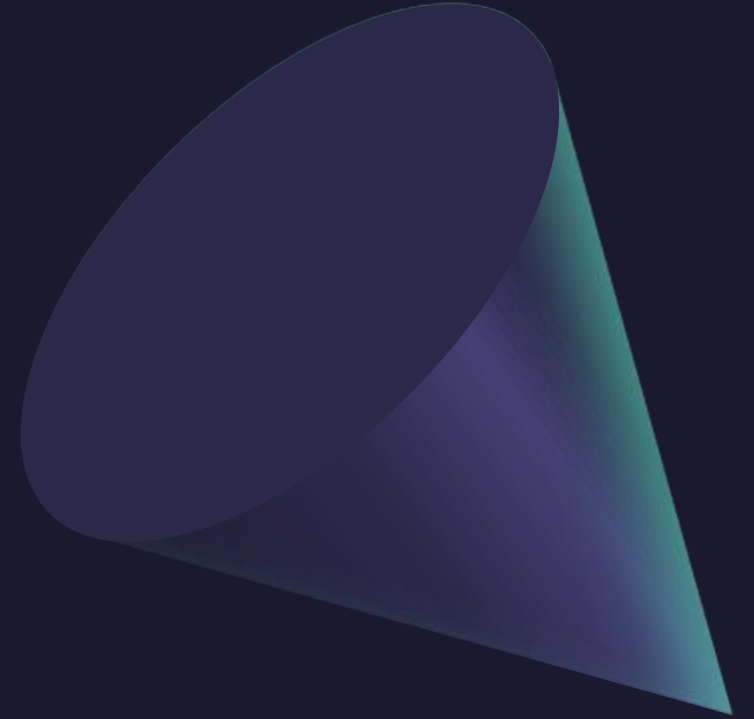




# CSS

Software Systems Development

IIIT Hyderabad



# Content

Meaning

Anatomy

Selectors

Combining  
Selectors

Conflict  
Resolution

The Box  
Model

Media  
Queries

Responsive  
Design

# Cascading Stylesheet



## Cascading

The selection of the prioritized rule is given by global rule to local or inline rule.

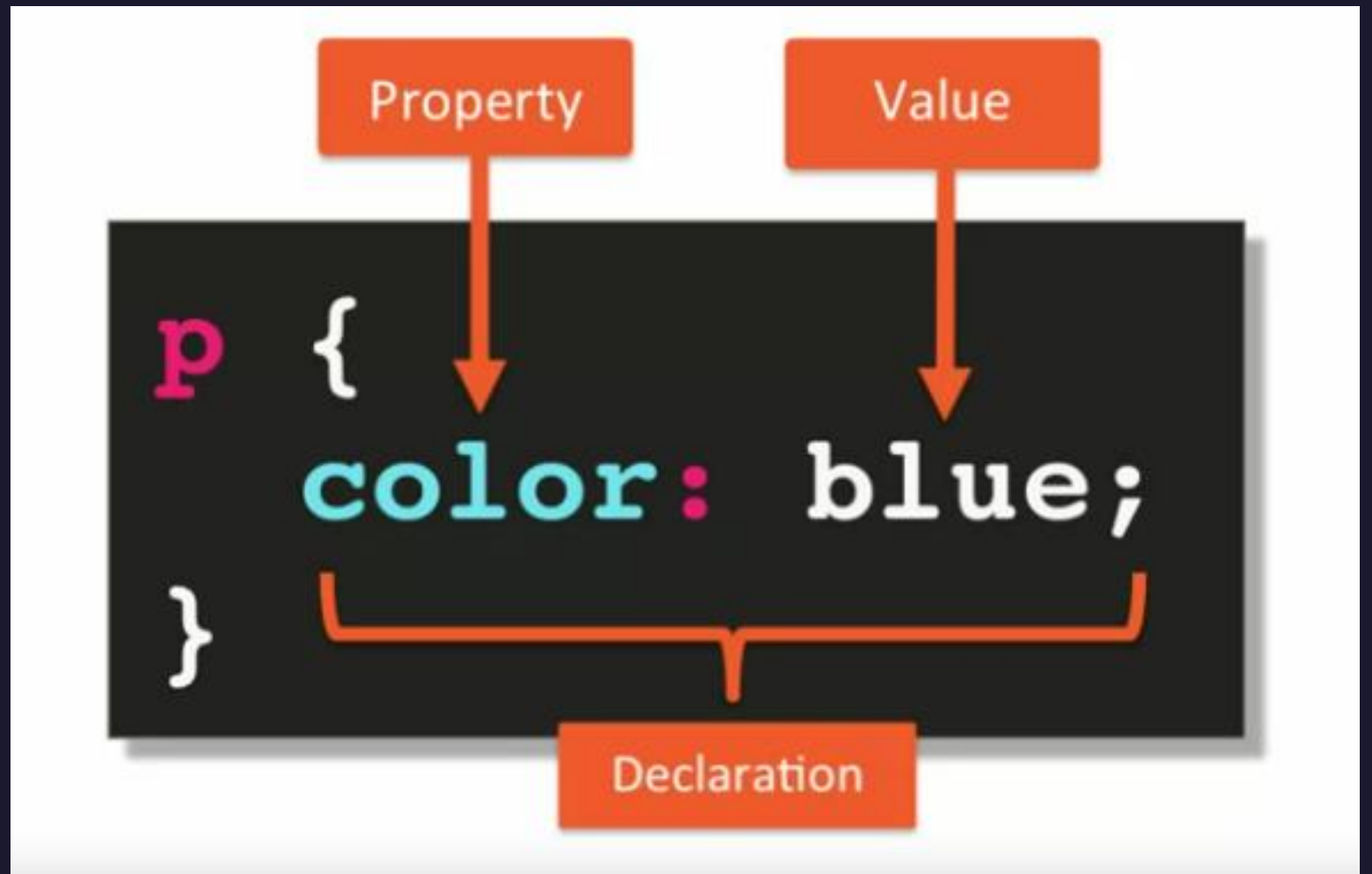
This has a cascading effect from global to local



## Stylesheet

Collection of CSS rules applied together

# Anatomy of a CSS rule



# Anatomy of a CSS rule

```
p {  
  color: blue;  
  font-size: 20px;  
  width: 200px;  
}  
  
h1 {  
  color: green;  
  font-size: 36px;  
  text-align: center;  
}  
...
```

Stylesheet

```
<link rel="stylesheet" href="style.css">
<style>

h2 {
  color: maroon;
}

</style>
</head>
<body>
<h1>Style Placement</h1>
<h2>Subheading 1</h2>
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Ex et simillique
cupiditate dignissimos delectus nulla
dolore, eos dolorum quidem excepturi,
facere non repellat accusamus, atque? Id
voluptate eos et accusantium.</p>
<h2>Subheading 2</h2>
<p style="text-align: center;">I am
centered!</p>
```

# Selectors

- CSS selectors are used to determine which HTML element, or set of elements, to apply the CSS declarations to.
- The browser uses its selector API to traverse the DOM or Document Object Model, and pick out the elements matching the selector.



# Element Selector



```
p {  
  color: blue;  
}
```

```
...  
<p>...</p>  
...  
<p>...</p>
```

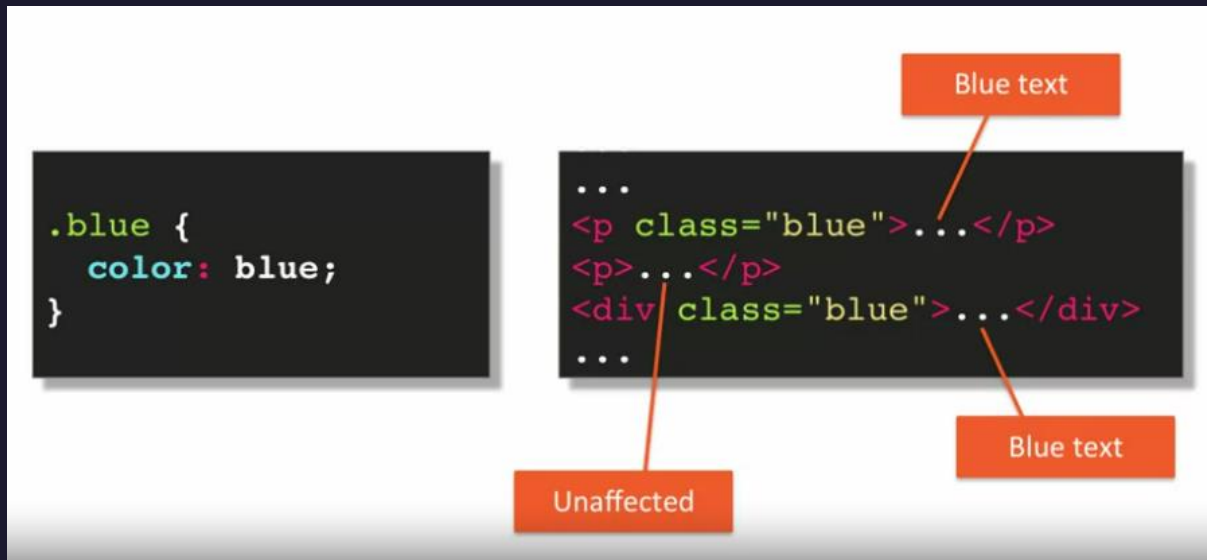
Blue text

Blue text

- The element selector is for specifying the element name.
- p for paragraph says that every paragraph in our HTML document should be colored blue.



# Class Selector



- The class selector is specified with a dot and the name of the class.
- We have a p, the paragraph tag and div tag both have the attribute `class="blue"` and therefore, their content or their text content will be colored blue.

# Id Selector

- We specify an id selector is by specifying the value of an id of an element within your HTML document, preceded by a pound sign.
- The name is a value of some id within our HTML document.
- We can also group multiple selectors for ex. Element OR Class

```
#name {  
  color: blue;  
}
```

Unaffected

```
...  
<p>...</p>  
<div id="name">...</div>  
...
```

Blue text

Separate selectors with commas

```
div, .blue {  
  color: blue;  
}
```

Blue text

```
...  
<p class="blue">...</p>  
<p>...</p>  
<div>...</div>  
...
```

Blue text

# Combining Selectors

```
p.big {  
    font-size: 20px;  
}
```

# Combining Selectors

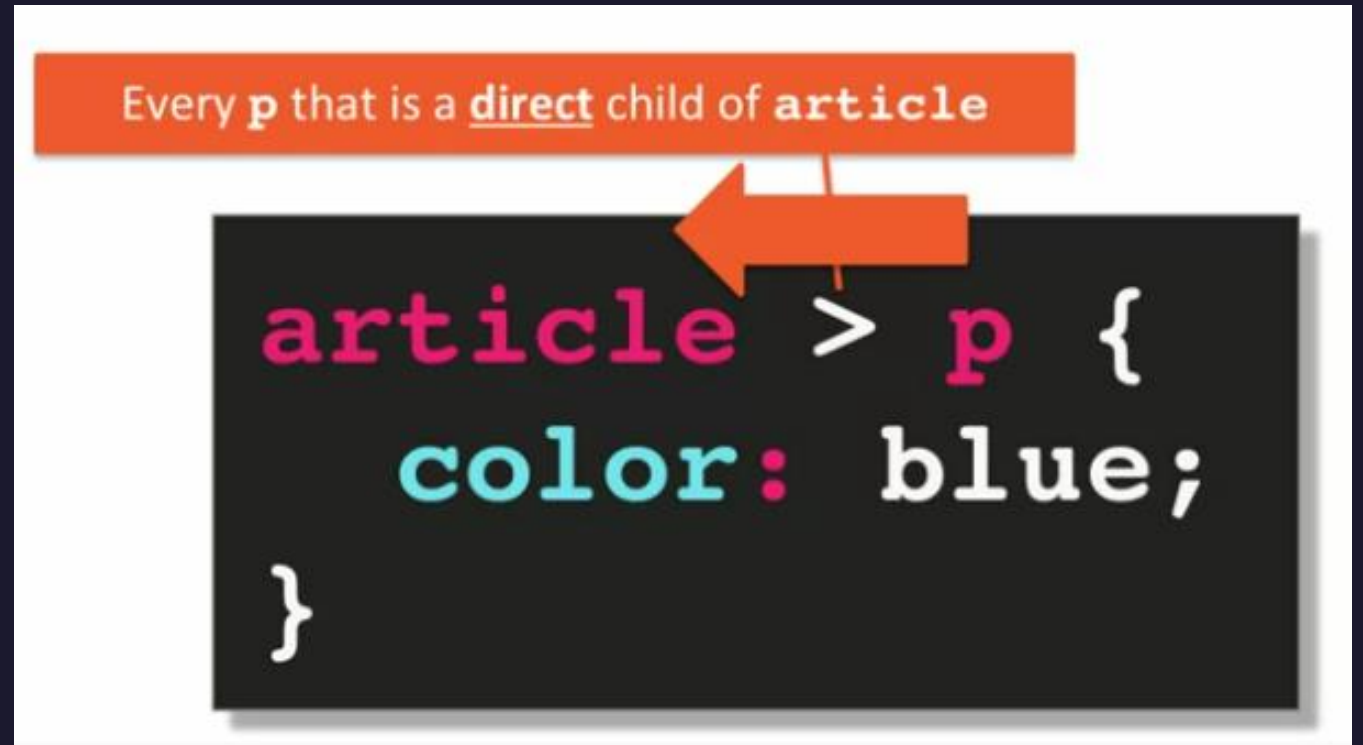
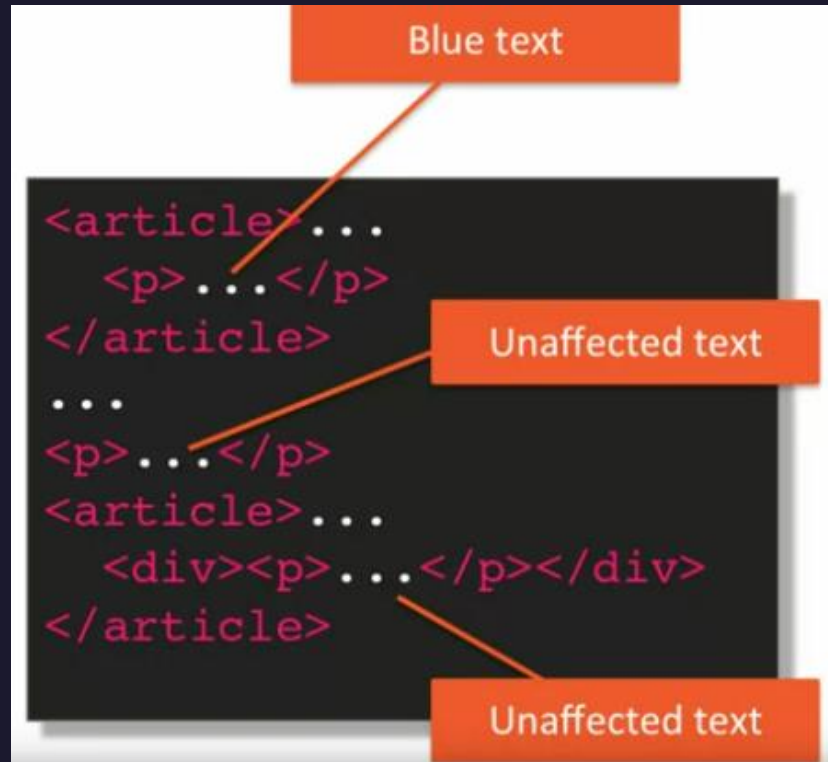
```
p.big {  
  font-size: 20px;  
}
```

```
...  
<p class="big">...</p>  
<div class="big">...</div>  
...
```

Text size 20px

Unaffected text

# Child Selector



# Descendent Selector

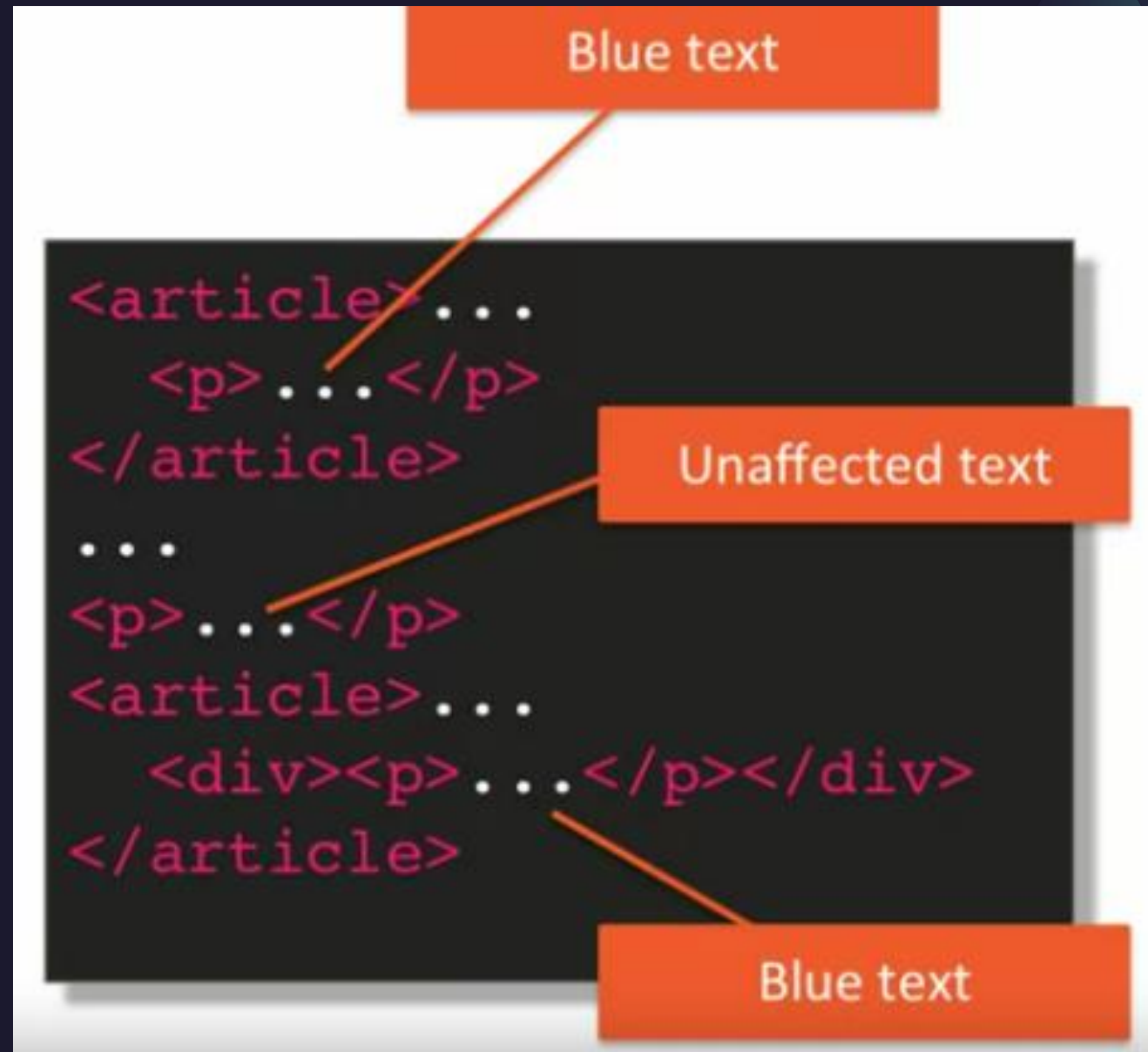
Every **p** that is inside (at any level) of **article**

```
article p {  
  color: blue;  
}
```

# Descendent Selector

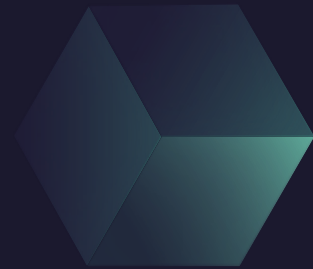
```
<article>...  
  <p>...</p>  
</article>  
  
...  
<p>...</p>  
<article>...  
  <div><p>...</p></div>  
</article>
```

# Descendent Selector





# Combining Selectors




```
.colored p {  
  color: blue;  
}
```

```
article > .colored {  
  color: blue;  
}
```




# Combining Selectors



```
.colored p {  
  color: blue;  
}
```

Every **p** that is inside (at any level) an element with `class="colored"`



```
article > .colored {  
  color: blue;  
}
```

Every element with `class="colored"` that is a direct child of **article** element

# Pseudo-class Selectors

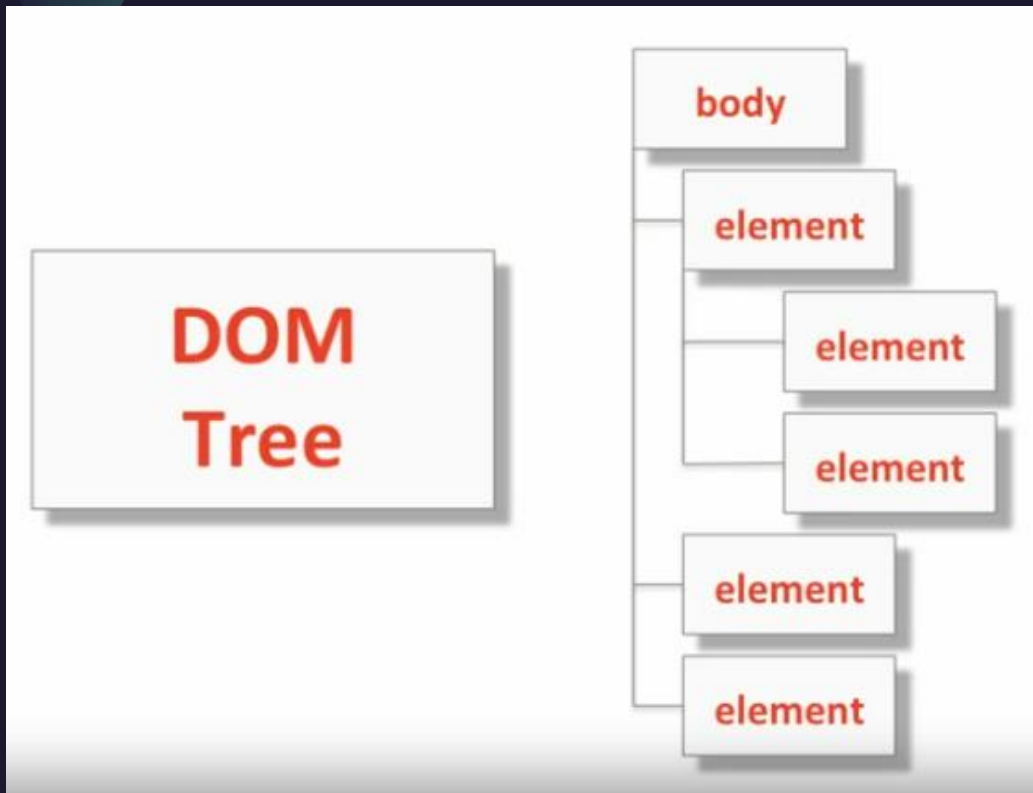


```
selector:pseudo-class {  
  ...  
}
```

- Pseudo-class selectors address targeting only the structures that can be targeted by simple combinations of regular selectors, or targeting the ability to style based on user interaction with the page.
- For example, we would want the styling of an element to change if the user hovers or moves their mouse over that element
- :link - any link state (<a>)
- :visited - link which is clicked
- :hover - hover over the element
- :active - mouse clicked and not released
- :nth-child - nth child can be targeted for style



# Conflict Resolution



- Origin Precedence (when in conflict) : last declaration wins.
- Merge Precedence (When not in conflict) : merges all rules specifying different target elements.
- Inheritance : If property is defined for parent, then all descendants will have that property.
- Specificity : Most specific selector combination wins.

# Conflict Resolution (Specificity)

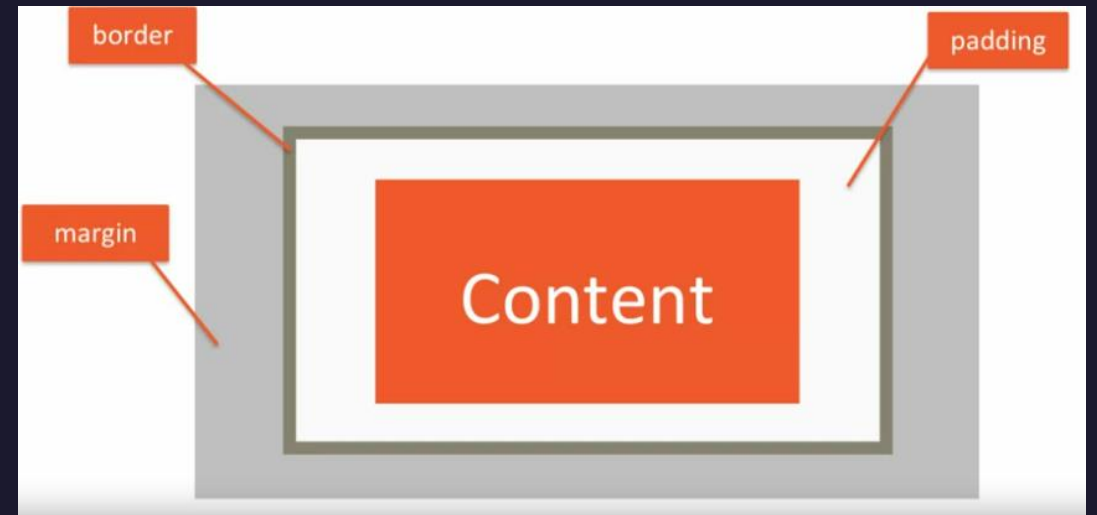
style="..."	ID	Class, pseudo-class, attribute	# of Elements
0	0	0	2

```
div p { color: green; }
```

<pre>div #myParag {   color: blue; }</pre>	?	<pre>div.big p {   color: green; }</pre>
0 1 0 1		0 0 1 2
✓		✗

# The Box model

```
#box {  
  background-color: blue;  
  padding: 10px 10px 10px 10px;  
  border: 5px solid black;  
  margin: 40px;  
}
```



# Media Queries

- It allow us to group styles together and target them to devices based on some criteria.
- For example, you can target a device by its width, its height, or orientation like landscape or portrait.

```
@media (max-width: 800px) {...}
```

```
@media (min-width: 800px) {...}
```

```
@media (orientation: portrait) {...}
```

```
@media screen {...}
```

```
@media print {...}
```

Media Feature (resolves to true or false)

```
@media (max-width: 767px) {  
  p {  
    color: blue;  
  }  
}
```

If TRUE,  
styles within  
curly braces  
apply.

Devices with width within a range

```
@media (min-width: 768px) and (max-width: 991px) {...}
```

Comma is equivalent to OR:

```
@media (max-width: 767px) , (min-width: 992px) {...}
```

# Responsive Design

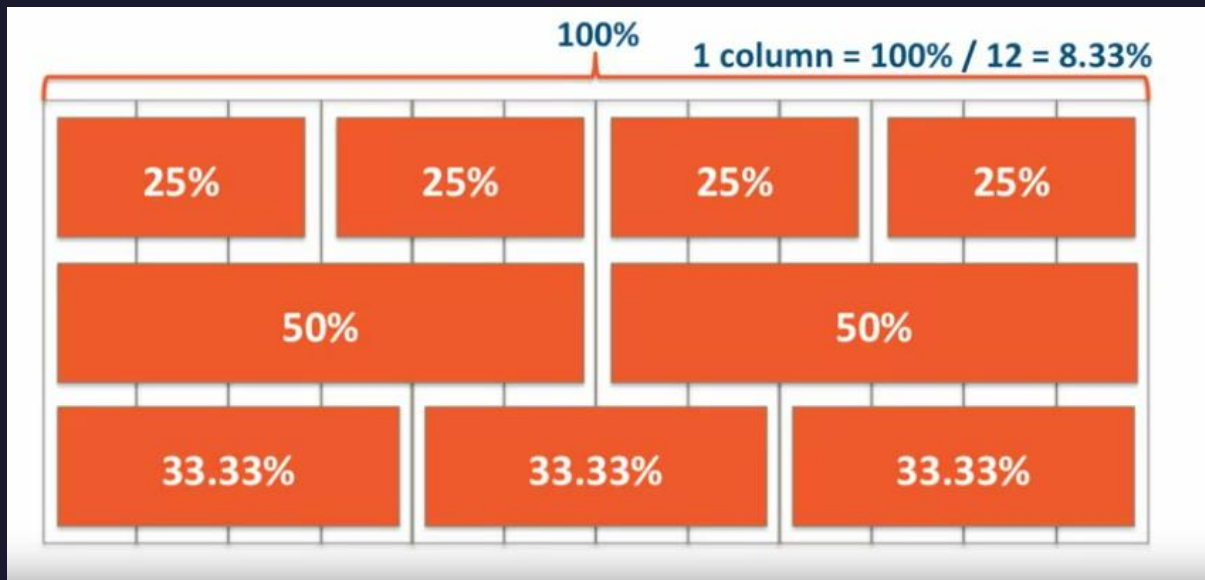
- Site designed to adapt its layout to the viewing environment by using fluid, proportion-based grids, flexible images and CSS3 media queries.
- Sites layout adapt to the size of the device.
- Visual delivery may change based on the importance of an element. For ex, "contact us" should be given more space in mobile than in desktop.



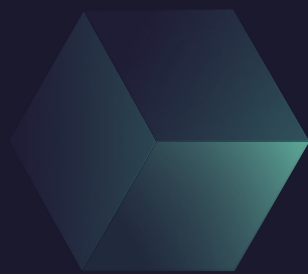
An alternative to responsive design



# 12-Column grid responsive layout



- The most common layout out there or responsive layout, is a 12-column grid responsive layout and this is what, a bootstrap, Twitter bootstrap uses.
- We could have content that is split into three columns out of the 12 (3-3-3-3) for a browser window of 100%.
- This layout defines how the elements will be displayed based on the device used.

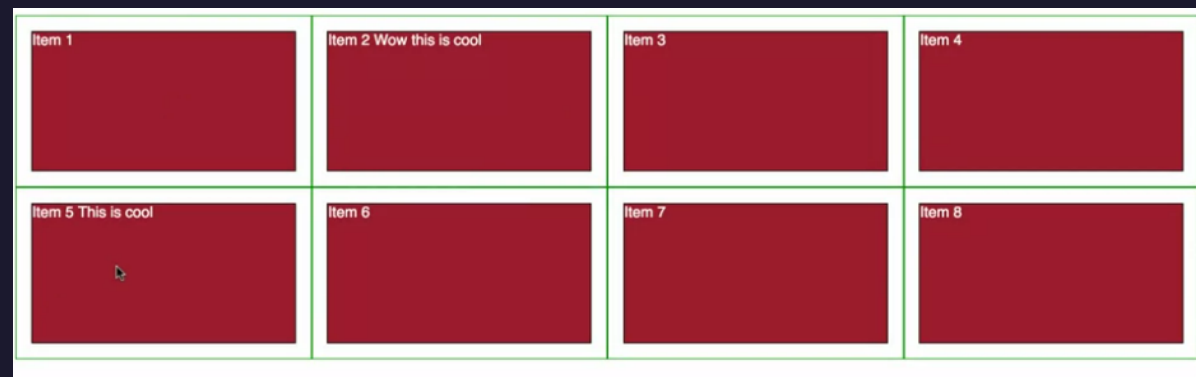


```
/****** Medium devices only *****/
@media (min-width: 950px) and (max-width: 1199px) {
  .col-md-1, .col-md-2, .col-md-3, .col-md-4, .col-md-5, .col-md-6,
  .col-md-7, .col-md-8, .col-md-9, .col-md-10, .col-md-11, .col-
  md-12 {
    float: left;
    border: 1px solid green;
  }
  .col-md-1 {
    width: 8.33%;
  }
  .col-md-2 {
    width: 16.66%;
  }
  .col-md-3 {
    width: 25%;
  }
  .col-md-4 {
    width: 33%;
  }
}
```

```
/****** Large devices only *****/
@media (min-width: 1200px) {
  .col-lg-1, .col-lg-2, .col-lg-3, .col-lg-4, .col-lg-5, .col-lg-6,
  .col-lg-7, .col-lg-8, .col-lg-9, .col-lg-10, .col-lg-11, .col-
  lg-12 {
    float: left;
    border: 1px solid green;
  }
  .col-lg-1 {
    width: 8.33%;
  }
  .col-lg-2 {
    width: 16.66%;
  }
  .col-lg-3 {
    width: 25%;
  }
  .col-lg-4 {
    width: 33%;
  }
  .col-lg-5 {
    width: 41.66%;
  }
}
```

```
<body>
<h1>Responsive Layout</h1>

<div class="row">
  <div class="col-lg-3 col-md-6"><p>Item 1</p></div>
  <div class="col-lg-3 col-md-6"><p>Item 2 Wow this is cool</p></div>
  <div class="col-lg-3 col-md-6"><p>Item 3</p></div>
  <div class="col-lg-3 col-md-6"><p>Item 4</p></div>
  <div class="col-lg-3 col-md-6"><p>Item 5 This is cool</p></div>
  <div class="col-lg-3 col-md-6"><p>Item 6</p></div>
  <div class="col-lg-3 col-md-6"><p>Item 7</p></div>
  <div class="col-lg-3 col-md-6"><p>Item 8</p></div>
</div>
```



Thank you

