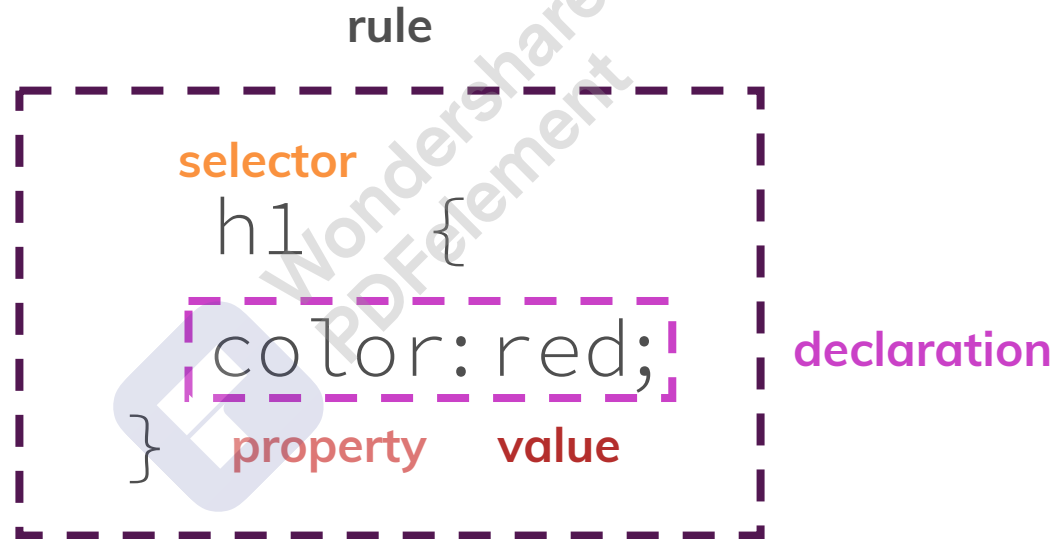


Understanding the CSS Syntax



More about Selectors



Elements

➡ Set equal style for these elements

```
<h1>Our header</h1>
<p>The Blog Post</p>
<div>More Info</div>
```

```
h1 {
  color: red;
}
```

Classes

➡ Set equal style for elements within the same class

```
<h1 class="blog-post">
Our header</h1>
<p class="blog-post">
The blog post</p>
<div class="blog-post">
More info</div>
```

```
.blog-post {
  color: red;
}
```

Universal

```
<h1>Our header</h1>
<p class="blog-post">The
blog post</p>
```

```
* {
  color: red;
}
```

Rarely use this one!

More about Selectors



IDs

- ➔ Set style to one specific element

```
<h1 id="main-title">Our  
header</h1>
```



```
#main-title {  
  color: red;  
}
```

Attributes

- ➔ Set equal styles to all elements with attribute(s)

```
<button disabled>  
  Click  
</button>
```

```
[disabled] {  
  color: red;  
}
```

Understanding Combinators

 Adjacent Sibling

```
div + p {  
}
```

 General Sibling

```
div ~ p {  
}
```

 Child

```
div > p {  
}
```

 Descendant

```
div p {  
}
```



Combinators – Adjacent Sibling

+ Adjacent Sibling

```
h2 + p {  
  color: red;  
}
```



```
<div>  
  <h2>Not applied</h2>  
  <p>CSS applied</p>  
  <h2>Not applied</h2>  
  <h3>Not applied</h3>  
  <p>Not applied</p>  
  <h2>Not applied</h2>  
  <p>CSS applied</p>  
</div>
```



- ✓ Elements share the same parent
- ✓ Second element comes **immediately** after first element



Combinators – General Sibling

~ General Sibling

```
h2 ~ p {  
  color: red;  
}
```



```
<div>  
  <h2>Not applied</h2>  
  <p>CSS applied</p>  
  <h2>Not applied</h2>  
  <h3>Not applied</h3>  
  <p>CSS applied</p>  
</div>
```



Element share the same parent



Second element comes after first element



Combinators – Child

> Child

```
div > p {  
  color: red;  
}
```



Second element is a direct child of first element




```
<div>  
  <div>Not applied</div>  
  <p>CSS applied</p>  
  <div>Not applied</div>  
  <article>  
    <p>Not applied</p>  
  </article>  
  <p>CSS applied</p>  
</div>
```




Combinators – Descendant

 Descendant

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




 Second element is a descendant of the first element



```
<div>  
  <div>Not applied</div>  
  <p>CSS applied</p>  
  <div>Not applied</div>  
  <article>  
    <p>CSS applied</p>  
  </article>  
  <p>CSS applied</p>  
</div>
```



Cascading Style Sheets & Specificity

Cascading

Multiple Rules can apply to the same Element

Specificity

Resolve conflicts arising from multiple Rules

Inline Styles

#ID selectors

.class, :pseudo-class and [attribute] selectors

<Tag> and ::pseudo-element selectors

Cascading Style Sheets & Specificity

Cascading

Multiple Rules can apply to the same Element

Specificity

Resolve conflicts arising from multiple Rules

Selector Hierarchy

Directly applied Styles win over Inheritance

`p {...} > div {...}` for
`<div><p>Hi!</p></div>`

More specific Selector wins over less specific one

`p.some-class {...} > p {...}` for
`<p class="some-class">Hi!</p>`

Inheritance

```
div {
  color: red;
}
```



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



Parent styles are inherited by child elements if not overwritten!



```
<div>
  <div>
    <h1>Inherited!</h1>
  </div>
  <p>Overwritten</p>
  <div>Inherited!</div>
  <article>
    <p>Overwritten</p>
  </article>
  <p>Overwritten</p>
</div>
```



Value Types

Values are tightly coupled to specific property!

Pre-defined Options	Colors	Length, Sizes & Numbers	Functions
<code>display: block;</code>	<code>background: red;</code>	<code>height: 100px;</code>	<code>background: url(...);</code>
<code>overflow: auto;</code>	<code>color: #fa923f;</code>	<code>width: 20%;</code>	<code>transform: scale(...);</code>
	<code>color: #ccc;</code>	<code>order: 1;</code>	

Possible Values can be found in CSS References
(e.g. MDN)!

Summary

CSS works with Rules

```
h1 {  
  color: red;  
}  
p {  
  color: red;  
}
```

Different Types of Selectors

```
h1 {...}  
.some-class {...}  
[disabled] {...}  
#some-id {...}  
* {...}
```

Selectors with Combinators

```
div + p {  
  color: red;  
}  
div ~ p {  
  color: red;  
}  
div > p {  
  color: red;  
}  
div p {  
  color: red;  
}
```

Properties & Values

- Long list of available Properties and Values
- Check MDN or comparable References
- Different Type of Values, depending on Properties

Inheritance & Specificity

- Parent styles are generally inherited
- Multiple rules can apply to one element
- Specificity resolves “multiple rules” conflicts
- Inheritance defaults can be changed