

# CSS Basics

---

## CSS Basics



markup · presentation · style · rules  
· selectors · declarations ·  
properties · values



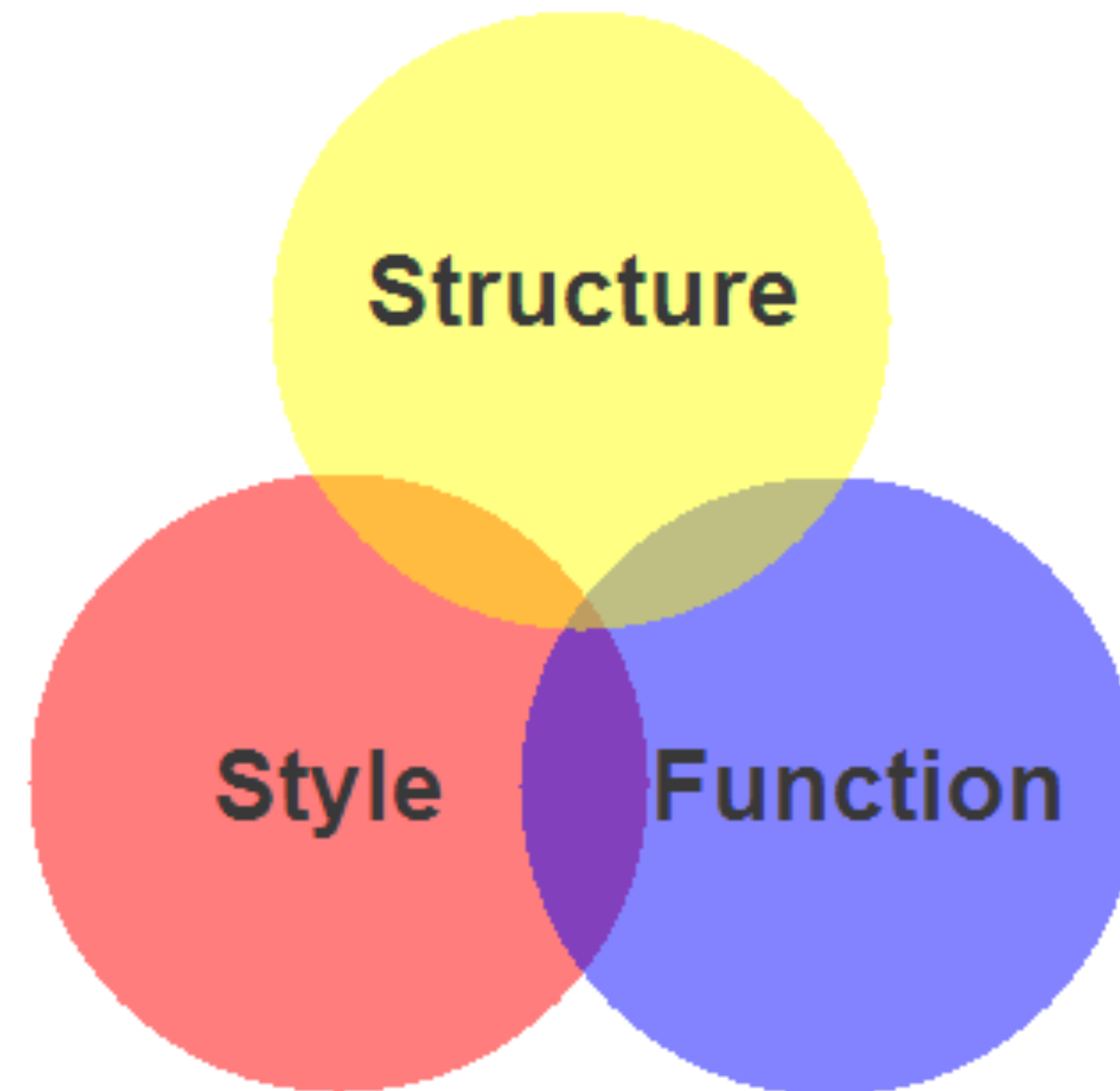




# Markup, Style, Function

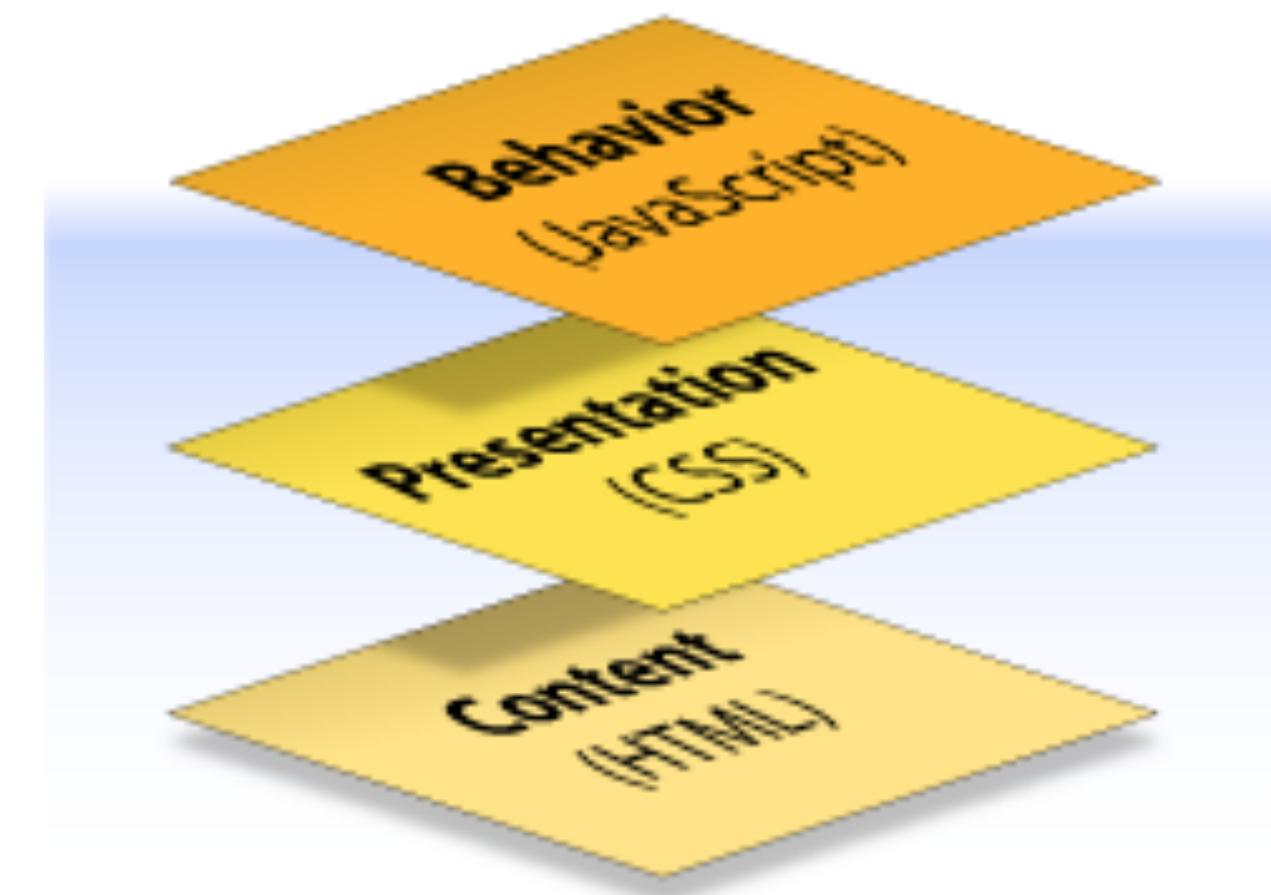
---

- Markup (XHTML, HTML)
  - Structure
  - Content
- Style (CSS)
  - Style
  - Presentation
  - Appearance
- Function (Javascript)
  - Actions
  - Manipulations



# Content, Presentation, Behavior

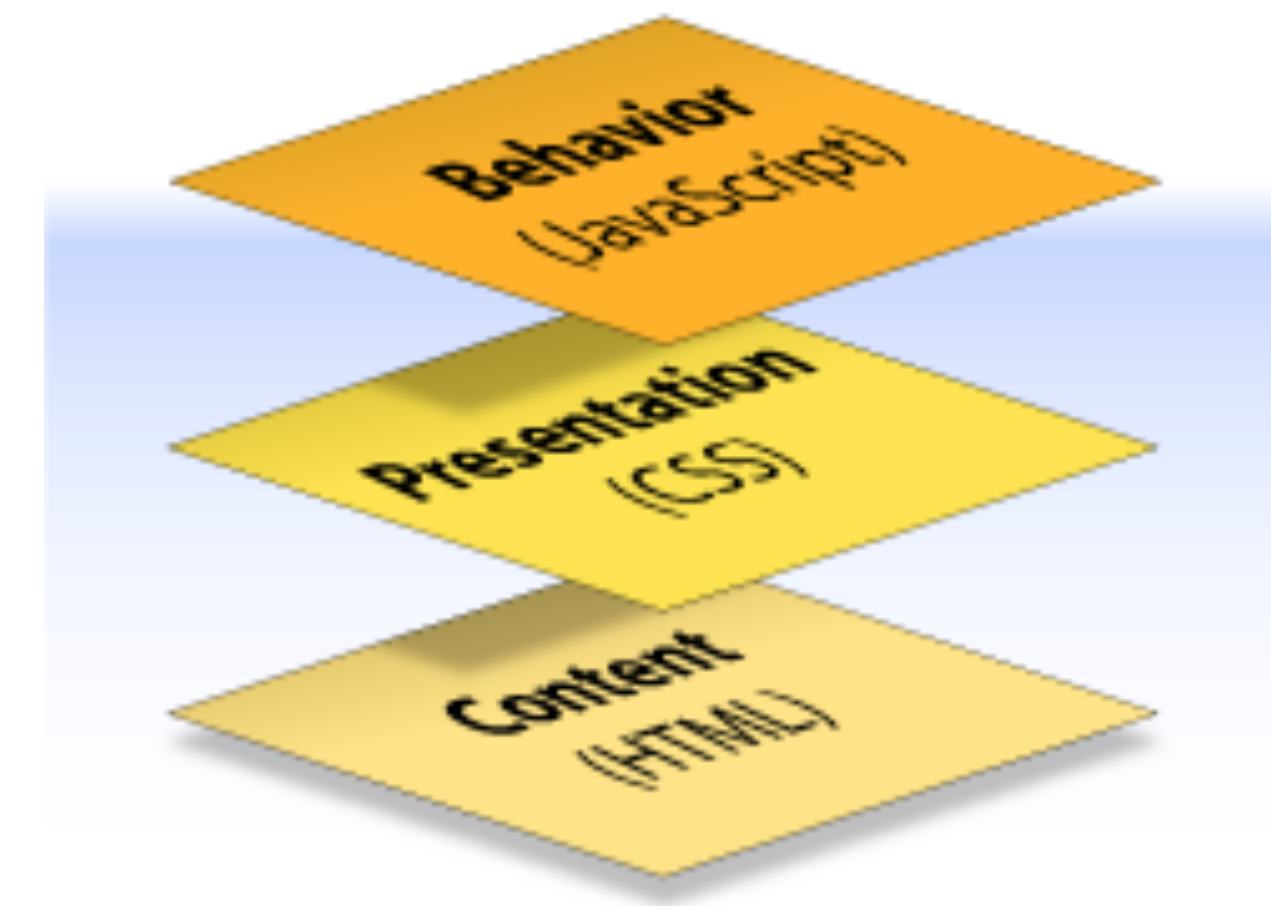
- **Content** comprises the information the author wishes to convey to his or her audience, and is embedded within HTML or XHTML markup that defines its structure and semantics.
- Most of the content on the Web today is text, but content can also be provided through images, animations, sound, video, and whatever else an author wants to publish



# Content, Presentation, Behavior

---

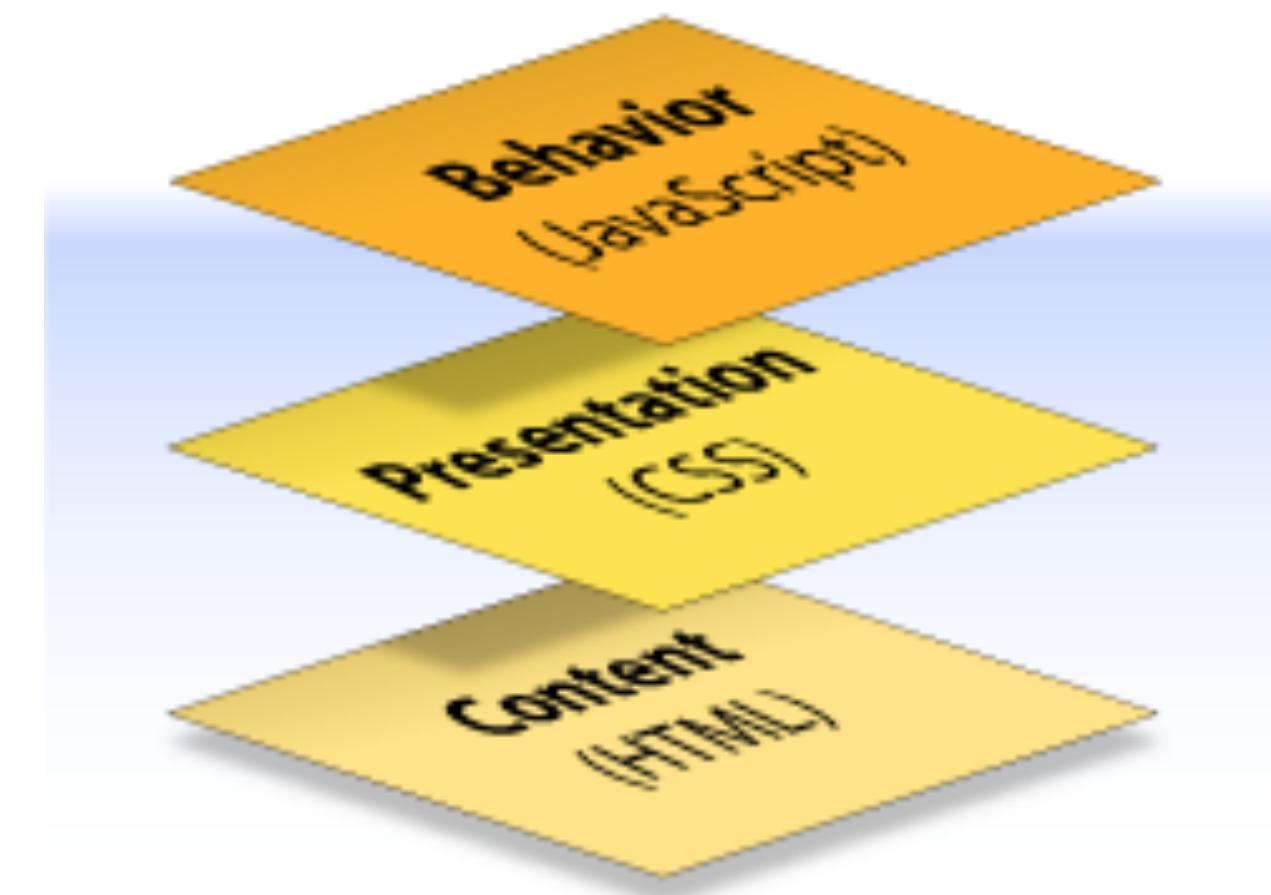
- **Presentation** defines how the content will appear to a human being who accesses the document in one way or another
- The conventional way to view a web page is with a regular web browser, of course, but that's only one of many possible access methods. For example, content can also be converted to synthetic speech for users who have impaired vision or reading difficulties



# Content, Presentation, Behaviour

---

- **Behaviour** layer involves real-time user interaction with the document.
- This task is normally handled by JavaScript.
- The interaction can be anything from a trivial validation that ensures a required field is filled in before an order form can be submitted, to sophisticated web applications that work much like ordinary desktop programs.



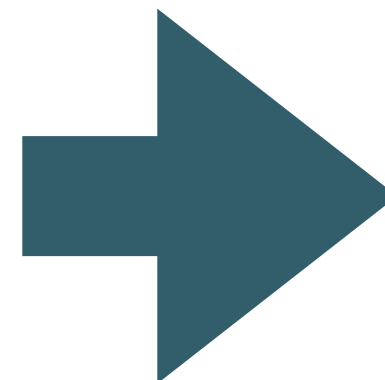


# Separation of Concerns

- It's possible to embed all three layers within the same document
  - `<em>` and `<b>` can be used to control the presentation of text, and `<hr>` will insert a visible rule element
- Sometimes called *Presentation Markup*, these types of elements embed presentation-layer information within the content layer, they negate any advantage we may have gained by keeping the layers separate.



Keeping them separate  
gives us one valuable  
advantage:



We can modify or replace  
any of the layers without  
having to change the  
others.

# What is CSS?



CSS is a language for specifying how documents are presented to users — how they are styled & laid out

Cascading Style Sheets is the recommended way to control the presentation layer in a web document.

The main advantage of CSS over presentational HTML markup is that the styling can be kept entirely separate from the content.

CSS also provides far better control over presentation than do presentational element types in HTML.



# Example

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```

style.css

```
h1 {
  color: blue;
  background-color: yellow;
  border: 1px solid black;
}

p {
  color: red;
}
```





```
h1 {  
  color: blue;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
p {  
  color: red;  
}
```

Two rules



```
h1 {  
  color: blue;  
  background-color: yellow;  
  border: 1px solid black;  
}
```

The first rule starts with an `h1` selector, which means that it will apply its property values to the `<h1>` element. It contains three properties and their values:

1. The first one sets the text color to blue.
2. The second sets the background color to yellow.
3. The third one puts a border around the header that is 1 pixel wide, solid (not dotted, or dashed, etc.), and colored black.



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

The second rule starts with a p selector, which means that it will apply its property values to the <p> element. It contains one declaration, which sets the text color to red.

```
h1 {  
  color: blue;  
  background-color: yellow;  
  border: 1px solid black;  
}  
  
p {  
  color: red;  
}
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>My CSS experiment</title>  
    <link rel="stylesheet" href="style.css">  
  </head>  
  <body>  
    <h1>Hello World!</h1>  
    <p>This is my first CSS example</p>  
  </body>  
</html>
```

Rules Applied to HTML document



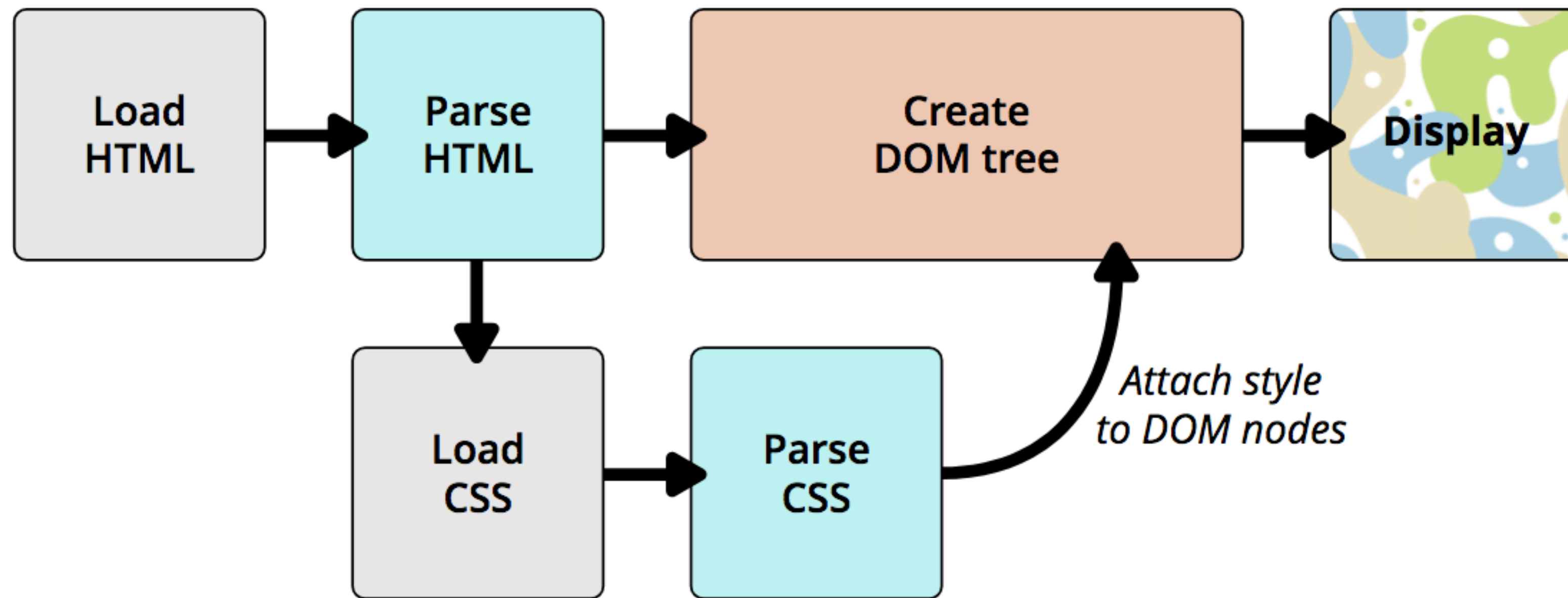
# Hello World!

This is my first CSS example

Browser Render

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```

# How CSS Does it Work?



1. The browser converts HTML and CSS into the DOM (Document Object Model). The DOM represents the document in the computer's memory. It combines the document's content with its style.
2. The browser displays the contents of the DOM.



# <link> element

---

Use the link element to "link in" external information.

The type of this information is "text/css". In other words, a CSS style sheet.

And the style sheet is located at this href (in this case we're using a relative link, but it could be a full-blown URL).

`<link type="text/css" rel="stylesheet" href="lounge.css" />`

The rel attribute specifies the relationship between the XHTML file and the thing you're linking to. We're linking to a style sheet, so we use the value "stylesheet".

<link> is an empty element.

# One Stylesheet - Multiple html pages

---

Index.html

```
<head>
  <title>App Store</title>
  <link type="text/css" rel="stylesheet" href="style.css" />
</head>
```

movies.html

```
<head>
  <title>Movies</title>
  <link type="text/css" rel="stylesheet" href="style.css" />
</head>
```

music.html

```
<head>
  <title>Music</title>
  <link type="text/css" rel="stylesheet" href="style.css" />
</head>
```



Style.css

Rules

Selectors

Declarations

Properties


Values



# CSS Rule

---

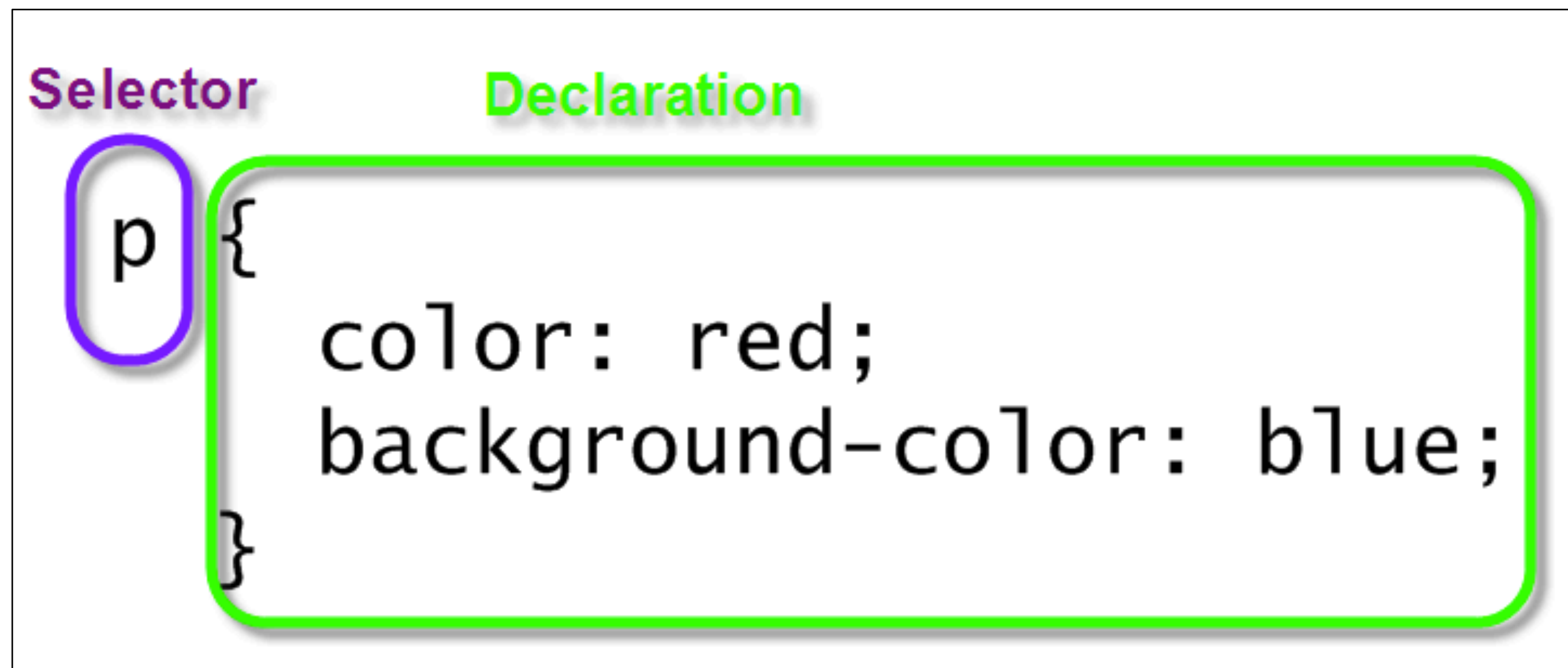
```
p {  
    color: red;  
    background-color: blue;  
}
```



CSS  
Rule

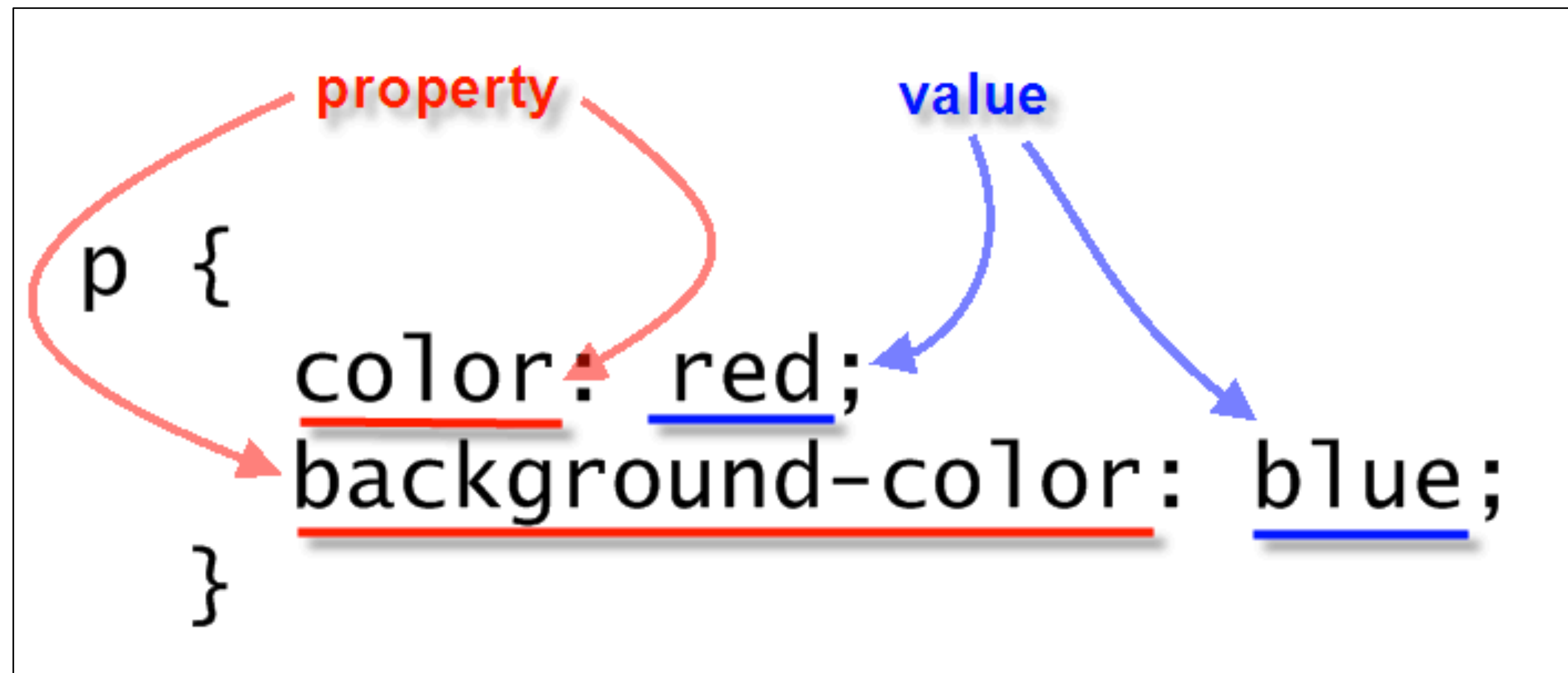
# Selector and Declarations

---



# Properties & Values

---





# A CSS Rule

The first thing you do is select the element you want to style, in this case the <p> element. Notice in CSS, you don't put <> around the name.

p {  
background-color: red;  
}

Place all the styles for the <p> element in between { } braces.

Then you specify the property you want to style, in this case the <p> element's background color.

There's a colon in between the property and its value.

And you're going to set the background-color to red.

At the end, put a semicolon.

We call the whole thing a RULE.

# More Properties...

---

- You can add as many properties and values as you like in each CSS rule.
- To put a border around your paragraphs:

```
p {  
    background-color: red;  
    border: 1px solid gray;  
}
```

The <p> element will have a border...

...that is 1 pixel thick, solid, and gray.

All you have to do is add another property and value.