Caltech | Center for Technology & Management Education

Full Stack Java Developer

simplilearn

CSS

Selectors

Learning Objectives

By the end of this lesson, you will be able to:

- Learn to write CSS syntax
- Understand the selector and tags to apply styles
- Differentiate between the types of selectors
- Learn CSS Pseudo classes and CSS Pseudo elements

Learning Objectives

By the end of this lesson, you will be able to:

- Get familiar with the concept of animation
- List the types of animation with their syntaxes
- Understand CSS media queries and their types

A Day in the Life of a Full Stack Developer

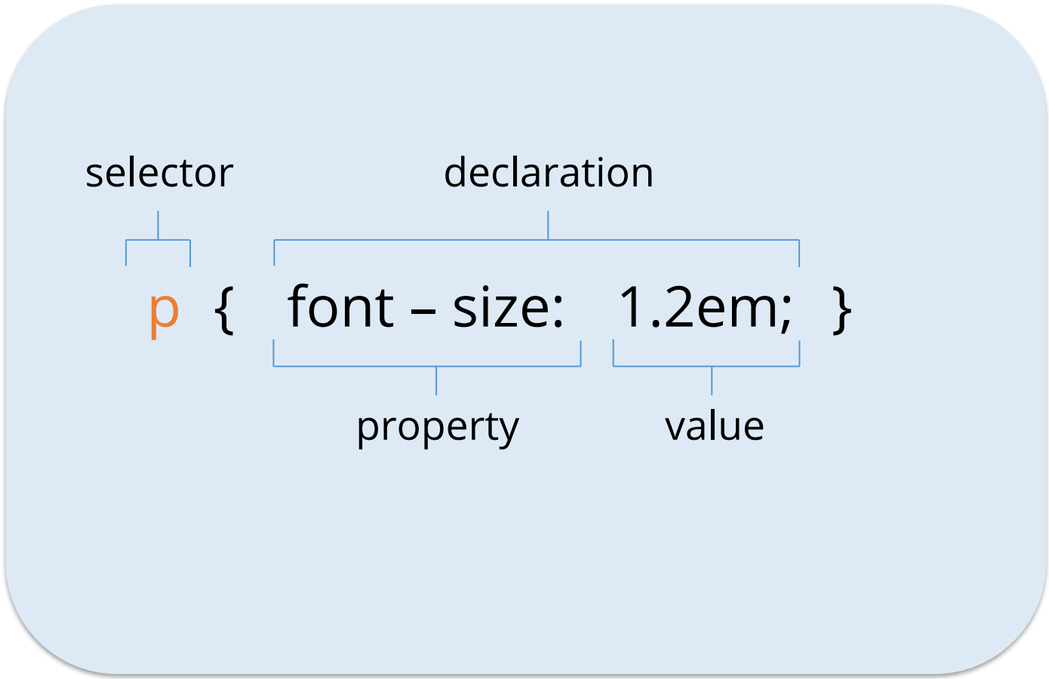
You are hired as a web developer in an organization and have been assigned an ongoing website development project. In this project, your task is to enhance the features of the website, like adding animations to the modules, like hover, highlight, delay, etc., based on the requirement.

To do so, examine the website and understand the use of media queries, animation elements, and more, to enhance the given website.

CSS Syntax

CSS Syntax

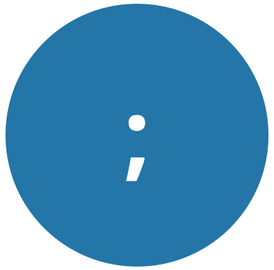
Stylistic rules of a CSS file are applied to the elements in the content.



CSS Syntax

A selector has to identify the HTML element to be styled.

Declarations are separated by a semi-colon.



{ Name: Value ; Name: Value }

Separates name and value

Separates each attribute statement

CSS Syntax

The CSS syntax is written as:

```
selector {
property: value
}
```

CSS Syntax: Example

CSS code:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
color: green.
text-align left.
font-size:20px.
}
</style>
</head>
<body>
<p>Thanks for reaching us! </p>
<p>These are basic syntax of CSS</p>
<p>This is a paragraph</p>
</body>
</html>
```

CSS selector

Selectors

Selectors

The selector instructs the browser on which HTML elements to apply the CSS property values.

A selector is an HTML tag that tells the browser where to apply the style.

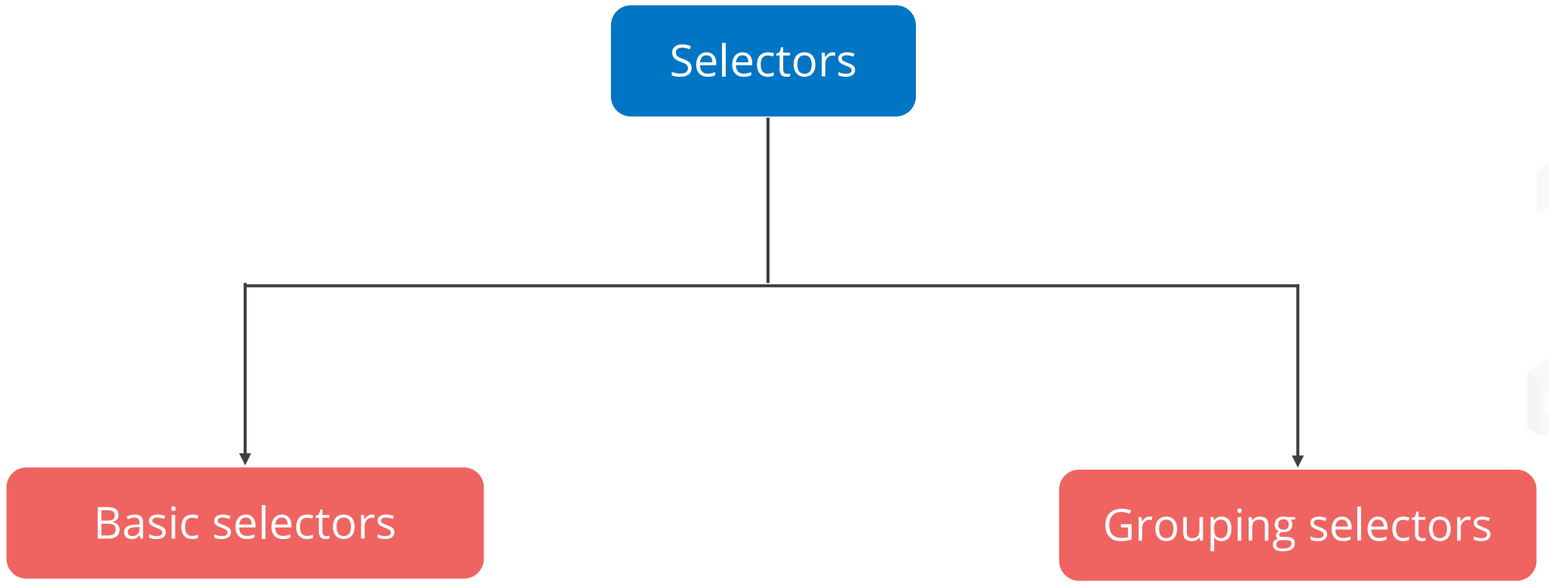
Selectors

The CSS syntax is written as:

```
selector {
property: value
}
```

Selectors: Classification

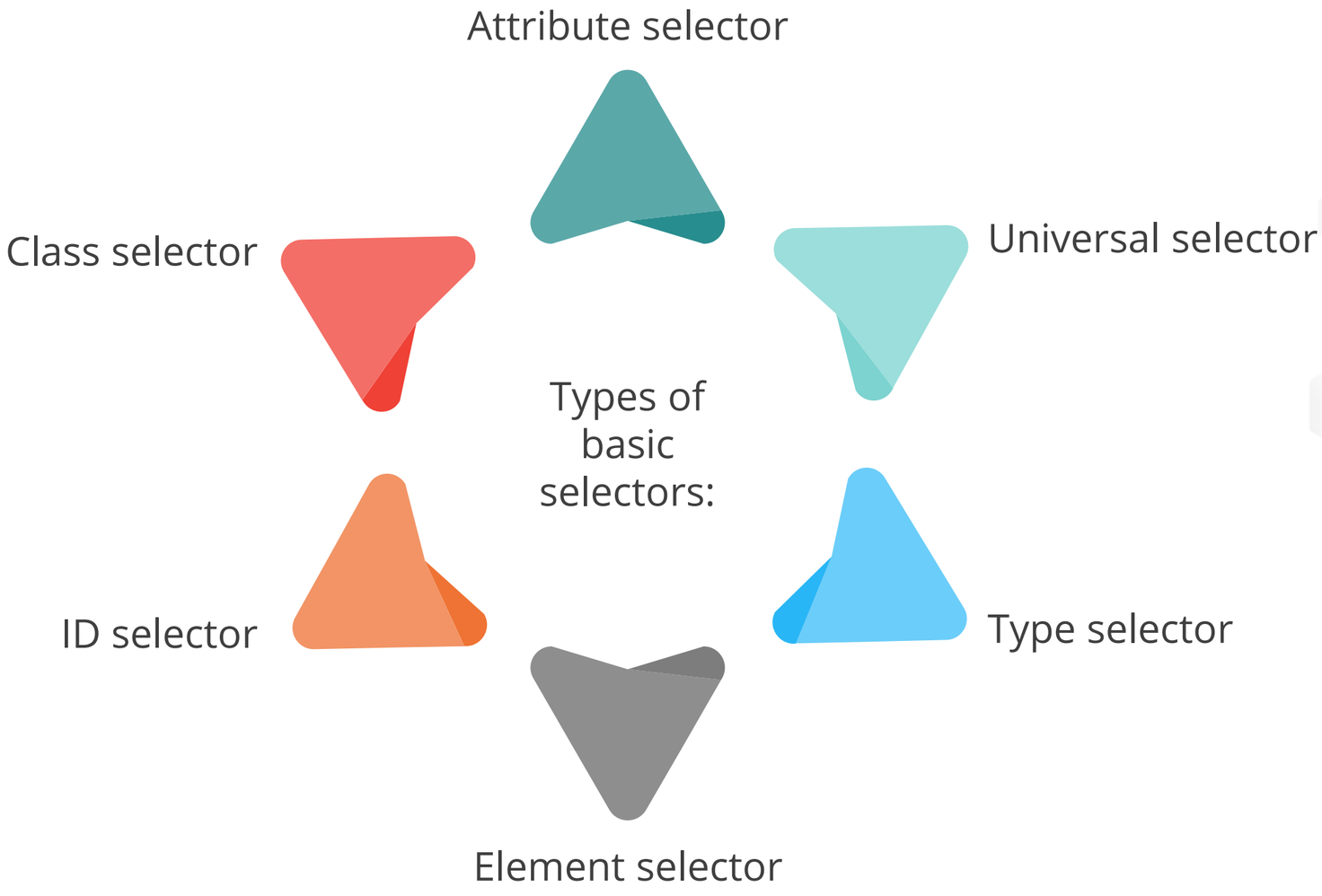
Selectors are classified into two categories:



Basic Selectors

Basic Selectors

Most commonly used selectors are:



Universal Selectors

It matches the name of any element type and applies to all the document's elements.

The syntax can be written as:

```
{
color: red
}
Or
{
}
Target all the elements in the document.
```

Type Selectors

It only matches the element name in the document.

The syntax can be written as:

```
h1, h2{
color: blue
}
Targets all the h1 and h2 in the document.
```

Element Selectors

Selects all elements with the same name as the provided element.

The syntax can be written as:

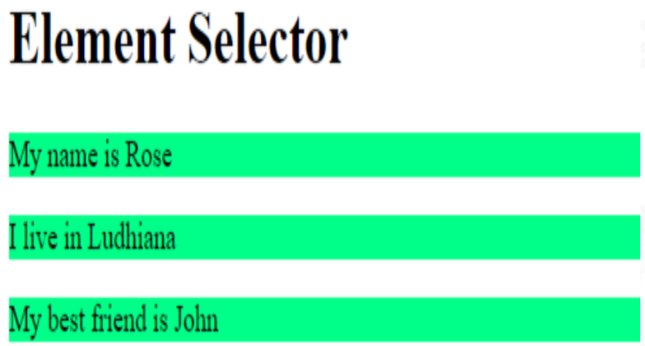
```
element {
CSS declarations.
}
```

Element Selector: Example

CSS code:

```
<!DOCTYPE html>
<html>
<head>
<style>
P {
Background-color: rgb (0,255,136).
}
</style>
<title>Element Selector Example</title>
</head>
<body>
<h1>Element Selector</h1>
<div>
<p id=”name”>My name is Rose</p>
<p id=” city”>I live in Ludhiana</p>
</div>
<p>My best friend is John</p>
</body>
</html>
```

Output:



ID Selector

An element's ID attribute is used to match it with an ID selector.

Each element can have only one ID.

The syntax:

```
#Id_value {
/* style properties*/
}
```

ID Selector: Example

Style CSS file:

```
#Custom {
color: #deb887.
font-size: 20px.
text-align left.
}
```

ID Selector: Example

HTML file:

```
<!DOCTYPE html>
<html>
<head>
<title>ID Selectors</title>
<link rel=" stylesheet" type="text/css" href="
style.css"/>
</head>
<body>
<p id=" color">This is colorful</p>
<p>This is colorless</p>
</body>
</html>
```

Output:

This is colorful

This is colorless

Class Selectors

Elements with a specific class attribute must be selected by the dot class selector.

The syntax is:

```
.class_value {
/* style properties*/
}
Or
t1{
}
Or
p.t1{
}
```

Class Selector: Example

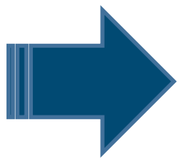
Style CSS file:

```
. custom {
color: #75a6e7.
font-size:20px.
text-align left.
}
```

HTML file:

```
<!DOCTYPE html>
<html>
<head>
<title>Class Selectors</title>
<link rel=" stylesheet" type="text/css" href="
style.css"/>
</head>
<body>
<p class=" custom">This is customized</p>
<p>This is not customized</p>
</body>
</html>
```

Output



This is customized

This is not customized

Attribute Selectors

It matches all the input elements that have a type of attribute with a value of the text.

The syntax is:

```
Input [type=”text”]
{
color: black.
}
```

Grouping Selectors

Grouping Selectors

With group selectors, multiple items can be selected and styled at once.

The syntax is:

```
element, element {
/*declarations*/
}
```


Comma (,)

Grouping Selectors: Example

CSS code:

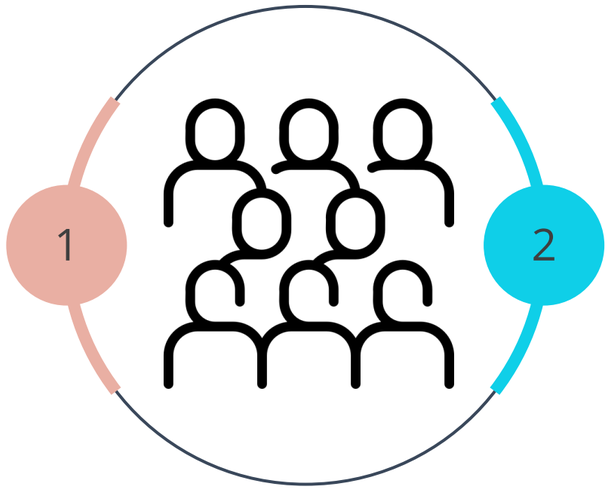
```
<!DOCTYPE html>
<html>
<head>
<style>
p, h1 {
text-align center.
Background-color: black.
Color: orange.
Width:50%.
}
</style>
<title>Grouping Selectors</title>
</head>
<body>
<p>This is a paragraph</p>
<h1>This is the heading</h1>
</body>
</html>
```

Output:

Types of Grouping Selectors

Group selectors are classified as:

Descendant selector



Child selector

Descendant Selectors

It matches an element that is a descendant of another element.

The syntax is:

```
The syntax is written as:
p a {
}
```

Descendant Selectors

It matches an element that is a descendant of another element.

The syntax is:

```
li> a {
}
Or
body > p {
color: black.
}
```

CSS Pseudo Classes

Pseudo Classes

A keyword that defines a special state for an element

The syntax is:

```
selector: pseudo-class {
property:value
}
```


CSS classes are used with pseudo-classes.

```
Selector.class: pseudo-class
{property:value}
```

Link

It assists in including a special style to the unvisited link.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style type=”text/css”>
a: link {
color: blue.
}
</style>
</head>
<body>
<h3>Link class example</h3>
<a href=””>Link</a>
</body>
</html>
```

Visited

It assists in adding a special style to the visited link.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style type=”text/css”>
a: visited {
color: green.
}
</style>
</head>
<body>
<h3>Visited class example</h3>
<a href=””>Click the link</a>
</body>
</html>
```

Hover

It is used to add a special style to an element when one moves the cursor over it.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style type=”text/css”>
a: hover {
color: orange.
}
</style>
</head>
<body>
<h3>Hover class example</h3>
<a href=””>Hover here</a>
</body>
</html>
```

Active Class

It helps to add a special style to the only active element.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style type=”text/css”>
a: active {
color: pink.
}
</style>
</head>
<body>
<h3>Active class example</h3>
<a href=””>Click Here</a>
</body>
</html>
```

Focus Class

It helps to add a special style to an element while the element has focused on it.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style type=”text/css”>
a: focus {
color: purple.
}
</style>
</head>
<body>
<h3>Focus class example</h3>
<a href=””>Click Here</a>
</body>
</html>
```

First Child

It helps to add a special style to an element that is the first child of another element.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style type=”text/css”>
div > p: first child {
text-indent:20px.
}
</style>
</head>
<body>
<div>
<p>The first paragraph in the div. It will be indented. </p>
<p>The second paragraph is not in the div. So, it will not be indented. </p>
</div>
<p>But this will not match the paragraph in this HTML. </p>
<div>
<h3>First child example</h3>
<p>This paragraph inside the div tag will not be affected</p>
</div>
</body>
</html>
```

Lang

It is used to add a special style to the elements.

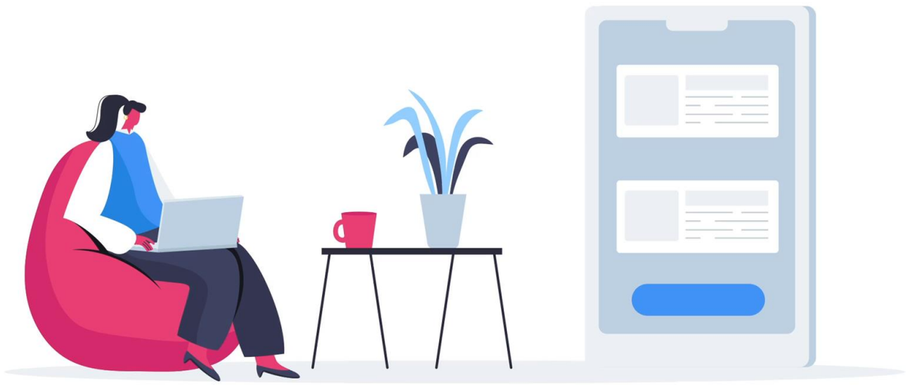
Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style type=”text/css”>
: lang(en) {quotes: '”'”'  ” ' ” ” ' ” }
: lang(fr) {quotes: “<<” “>>” “<” “>”}
</style>
</head>
<body>
<h3>Lang class example</h3>
<p>........<q lang=”fr”>A quote in the
paragraph</q>........<p>
</body>
</html>
```

CSS Pseudo Elements

CSS Pseudo Elements

It is employed to apply special effects to a few selectors.



The syntax is:

```
selector:pseudo-element {property: value}
```

```
selector:pseudo-element {property: value}
```

First Line

It is used to add special styles to the first line of the text in a selector.

Example: CSS code

```
<html>
   <head>
      <style type = "text/css">
         p:first-line { text-decoration: underline; }
         p.noline:first-line { text-decoration: none; }
      </style>
   </head>
   <body>
      <p class = "noline">
         Example for Firstline element
      </p>

      <p>
This example shows how to use: first-line pseudo element to give effects to
the first line of any HTML element.
      </p>
   </body>
</html>
```

First Letter

It is used to add a unique style to the first letter of the text in a selector.

Example: CSS code

```
<html>
   <head>
      <style type = "text/css">
         p:first-letter { font-size: 5em; }
         p.normal:first-letter { font-size: 10px; }
      </style>
   </head>

   <body>
      <p class = "normal">
         The first character of this statement is normal with a font size of 10 px;
      </p>

      <p>
This example shows how to use: first-letter pseudo element to design the first
character of an HTML element.

      </p>
   </body>
</html>
```

Before

It is used to insert content before an element.

Example: CSS code

```
<html>
   <head>
      <style type = "text/css">
         p:before {
            content: url(/images/bullet.gif)
         }
      </style>
   </head>

   <body>
      <p> This can be preceded by a bullet.</p>
      <p> This can be preceded by a bullet.</p>
      <p> This can be preceded by a bullet.</p>
   </body>
</html>
```

After

It is used to insert content after an element.

Example: CSS code

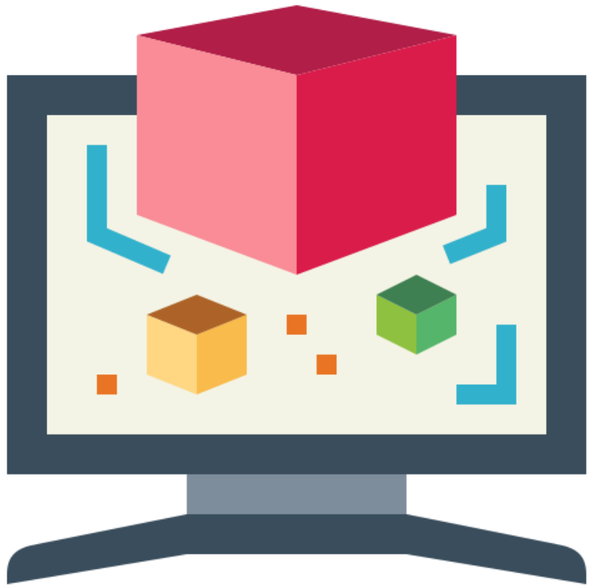
```
<html>
   <head>
      <style type = "text/css">
         p:after {
            content: url(/images/bullet.gif)
          }
      </style>
   </head>

   <body>
      <p> This line can be succeeded by a bullet.</p>
      <p> This line can be succeeded by a bullet.</p>
      <p> This line can be succeeded by a bullet.</p>
   </body>
</html>
```

DIV and SPAN Elements

CSS Animations

The animation property of CSS can be used to many other CSS properties, including background color, color, height, width, images, videos, audio, and more.



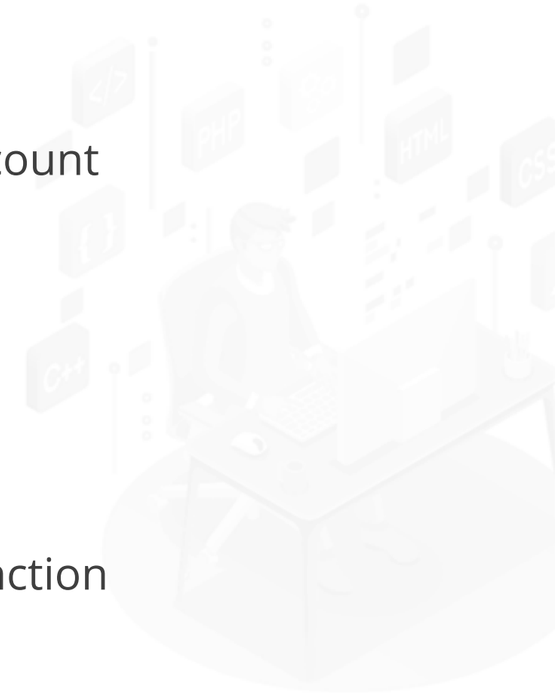
The keyframe stores the element's style or animation at specific points in time.

CSS Animations

With CSS, one can animate HTML elements without any JavaScript.

Some of the animation properties are:

1. @keyframes
2. animation-name
3. animation-duration
4. animation-delay
5. animation-iteration-count
6. animation-direction
7. animation-timing-function
8. animation-fill-mode



@keyframes

With keyframes, the animation transitions to the newest style from the current style.

It is important to bind the animation to an element.

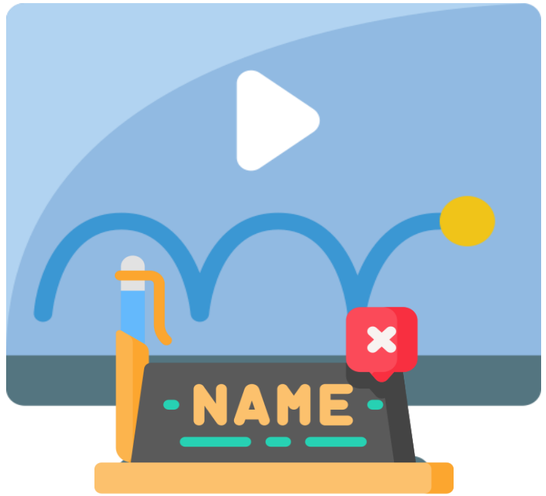
Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
width:200px.
height:200px.
background-color: orange.
animation-name: anim.
animation-duration:5s.
}
@Keyframes anim {
from {background-color: green;}
to {background-color: purple;}
}
```

```
</style>
</head>
<body>
<h2>CSS Animation</h2>
<div></div>
<p>Adding keyframes are must for the
CSS animation. The keyframe decides the
style or animation of the element at
certain times. </p>
</body>
</html>
```

Animation-name

It decides on which name or value the animation should be applied.



Animation-delay

This property indicates how long the animation should take to complete.

If there's no animation-duration property specified, no animation will occur without applying the duration property.

By default, its value is zero seconds.

Animation-delay

It is used to specify a delay for the start of the animation.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width:100px.
  height:100px.
 background-color: red.
position: relative.
animation-name:test.
animation-duration:5s.
animation-delay:2s.
}
@Keyframes example {
  0% {background-color: red; left:0px; top:0px;}
25% {background-color: blue; left:100px; top:0px;}
50% {background-color: orange; left:100px; top:100px;}
75% (background-color: green; left:0px; top:100px;}
100% {background-color: grey; left:0px; top:0px;}
}
```

```
</style>
</head>
<body>
<h3>Animation
properties</h3>
<p>
 </p>
<div></div>
</body>
</html>
```

Animation-iteration-count

It specifies the number of times the animation should run.

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width:100px.
  height:100px.
 background-color: red.
position: relative.
animation-name:test.
animation-duration:5s.
animation-iteration-count:3.
}
@Keyframes example {
  0% {background-color: red; left:0px; top:0px;}
25% {background-color: blue; left:100px; top:0px;}
50% {background-color: orange; left:100px; top:100px;}
75% (background-color: green; left:0px; top:100px;}
100% {background-color: grey; left:0px; top:0px;}
}
```

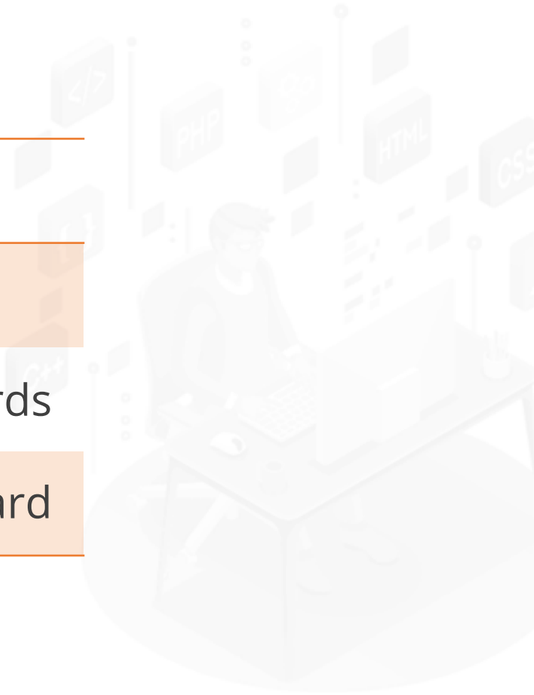
```
</style>
</head>
<body>
<h3>Animation iteration
count</h3>
<p>
 </p>
<div></div>
</body>
</html>
```

Animation-direction

It is used to specify whether an animation should play forward, backward, or in alternate cycles.

It can have the following values:

normal	Animation plays by default
reverse	Animation plays in the reverse direction
alternate	Animation plays forward first and then backwards
alternate-reverse	Animation plays backwards first and then forward



Animation-direction: Example

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
 width:100px.
  height:100px.
 background-color: green.
position: relative.
animation-name:test.
animation-duration:4s.
animation-direction: reverse.}
@Keyframes example {
  0% {background-color: red; left:0px; top:0px;}
25% {background-color: blue; left:100px; top:0px;}
50% {background-color: orange; left:100px; top:100px;}
75% (background-color: green; left:0px; top:100px;}
100% {background-color: grey; left:0px; top:0px;}
}
```

```
</style>
</head>
<body>
<h3>Animation direction</h3>
<p>
 </p>
<div></div>
</body>
</html>
```

Animation-timing-function

It is used to specify whether an animation should play forward, backward, or in alternate cycles.

It can have the following values:

ease	Specifies the flow of the animation
linear	Specifies the animation with the same speed
ease-in	Specifies the animation with a slow start
ease-out	Specifies the animation with the slow end
ease-in-out	Specifies the animation with the slow start and slow end
cubic-Bezier (n, n, n, n)	Defines values in the cubic-bezier function

Animation-timing-function: Example

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
width:400px.
height:200px.
background-color: green.
font-weight: bold.
position: relative.
animation: anim 5s infinite.
}
#div1{animation-timing-function: ease;}
#div2{animation-timing-function: linear;}
#div3{animation-timing-function: ease-in;}
#div4{animation-timing-function: ease-out;}
#div5{animation-timing-function: ease-in-out;}
@Keyframes anim {
from {left:0px;}
to{left:400px;}
}
```

```
</style>
</head>
</body>
<h3>Animation timing function
</h3>
<p>
 </p>
<div id=" div1">ease</div>
<div id=" div2">linear</div>
<div id=" div3">ease-in</div>
<div id=" div4">ease-out</div>
<div id=" div5">ease-in-out</div>
</body>
</html>
```

Animation-fill-mode

It is used to specify a style for the target element when the animation stopped.

It can have the following values:

none	Styles are not applied to the element before and after the execution.
forwards	The element keeps the original style values set by the last keyframes.
backwards	The element fetches the style values that are set by the first keyframe forwards.
both	The animation follows the rules for both forward and backward

Animation-fill-mode: Example

Example: CSS code

Let’s see an example:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
width:100px.
height:100px.
background: green.
position: relative.
animation-name: anim.
animation-duration:3s.
animation-fill-mode: forwards.
}
@Keyframes anim {
from {top:0px;}
to {top:100px; background-color: red;}
}
```

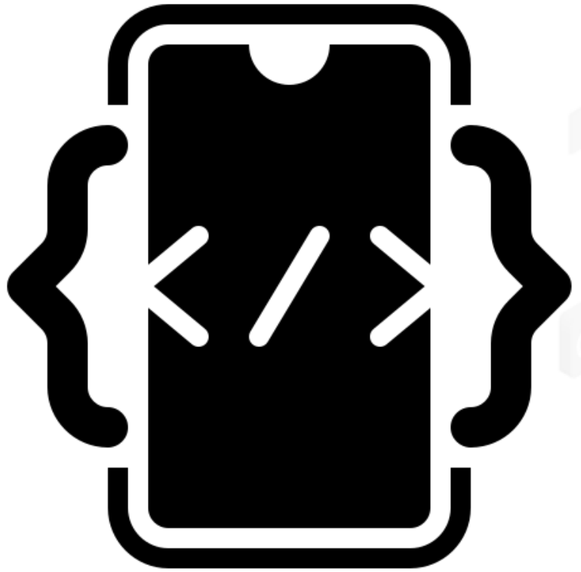
```
</style>
</head>
<body>
<h3>Animation fill mode</h3>
<p>
 </p>
<div></div>
</body>
</html>
```

TECHNOLOGY

CSS Media Queries

Media Queries

A media query contains media type and one or more expressions that resolve to either true or false.



It allows us to apply a special CSS for mobiles or adjust the layout for the print in this.

Media Queries

Media queries are used to check:

- Height and width of the viewport
- Height and width of the device
- Resolution
- Orientation

The syntax is:

```
@Media not | only mediatype and
(expressions) {
CSS-Code.
}
```

There are different stylesheets for different media.

```
@Media not | only mediatype and (expressions) {
CSS-Code. <link rel=" stylesheet" media=" mediatype and
|not|only (expressions) "href=" style.css">
}
```

Media Queries: Types

Some of the CSS media types are:

Value	Description
all	Used for all media type device
print	Used for printers
screen	Used for desktops, laptops, tablets, mobiles, and more
speech	Used for the screen readers to read the page loud

Media Query with min-width: Example

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
Background-color: green.
}
@Media screen and (min-width:480px) {
body {
background-color: red.
}
}
</style>
</head>
<body>
<h3>Media query Min width</h3>
<p>
</p>
</body>
</html>
```

Media Query with max-width: Example

Example: CSS code

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
Background-color: red.
}
@Media screen and (max-width:992px) {
body {
background-color: brown.
}
}
</style>
</head>
<body>
<h3>Media query Max width</h3>
<p>
 </p>
</body>
</html>
```

Key Takeaways

- The Selector instructs the browser on which HTML elements to apply the CSS property values.
- Universal selectors match the name of any element type and applies to all the document's elements.
- Element selectors select all elements with the same name as the provided element.
- With Group Selectors, multiple items can be selected and styled at once.
- With keyframes, the animation transitions to the newest style from the current style.
- A media query contains media type and one or more expressions that resolve to either true or false.

TECHNOLOGY

Thank You

simplilearn