Website Design using CSS

Cascading Style Sheet

- CSS stands for **Cascading Style Sheet.** It gives an additional style to the HTML document. A cascading style sheet is a language that is designed to define the document formatting and look written in a markup language. Generally, CSS is applied with HTML documents to change various styles of user interfaces and web pages.

Uses of CSS

- We can add unique styles to our old documents of HTML.
- We can change the overall look and feel of our website by following some changes in the CSS code.
- A cascading style sheet can be used with JavaScript and HTML in most of the websites to develop user interfaces for a lot of mobile applications and user interfaces for various web applications.

Features of CSS

1. **Opportunity in Web designing:** If anyone wants to begin a career in web designing professionally, it is essential to have knowledge of CSS and HTML.

1. **Website Design:** With the use of CSS, we can control various styles, such as the text color, the font style, the spacing among paragraphs, column size and layout, background color and images, design of the layout, display variations for distinct screens and device sizes, and many other effects as well.

1. **Web Control:** CSS has controlling power on the documents of HTML, so it is easy to learn. It is integrated with the HTML and the XHTML markup languages.

1. **Other Languages:** Once we have knowledge of some basics of CSS and HTML, other associated technologies like Angular, PHP, and JavaScript are become clearer to understand.

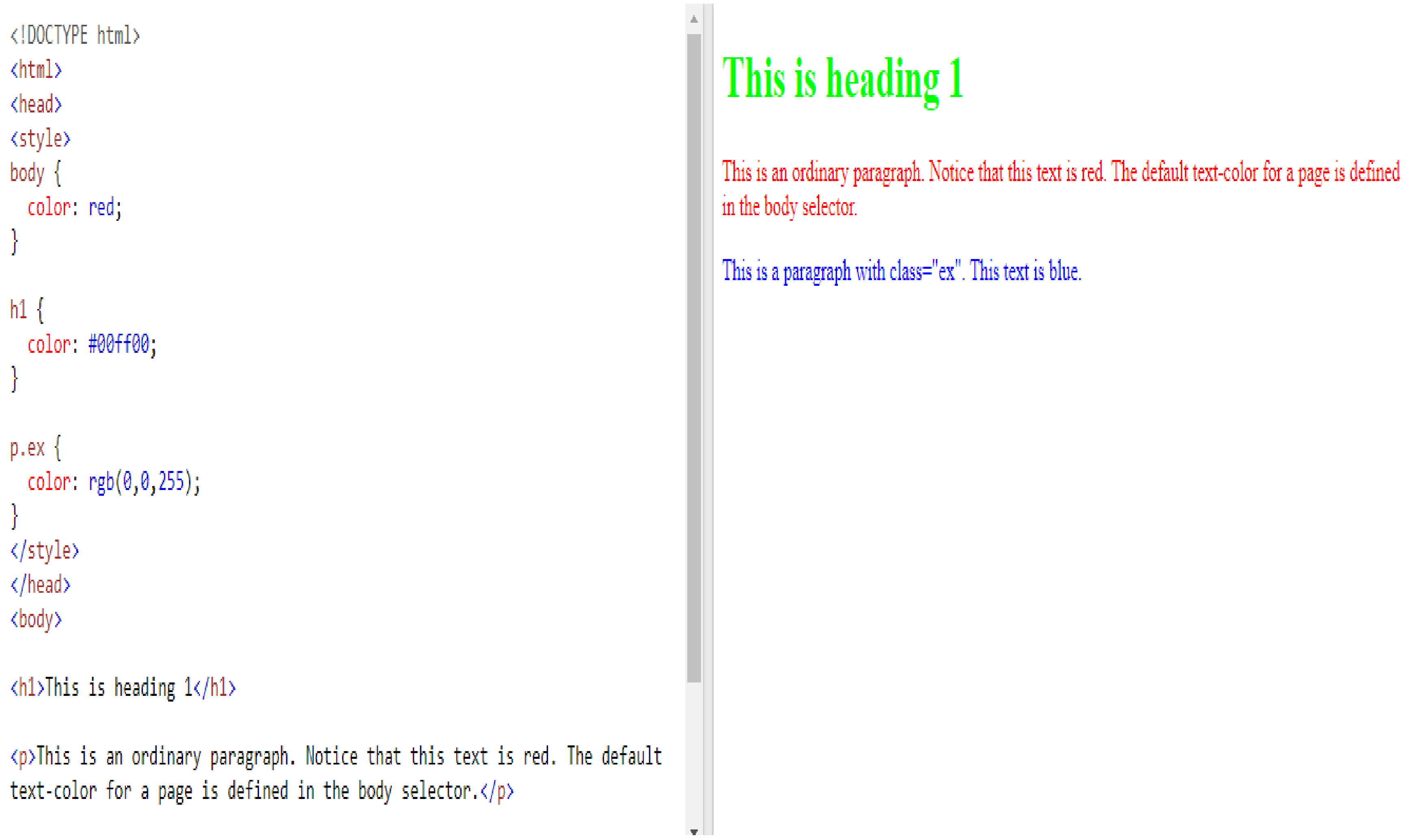
CSS Applications

- **Fast Page Loading:** We don’t need to mention the attributes of the HTML element every time if we use CSS. We need to specify one rule of CSS for an element and use it for every occurrence of that element. So, short code means high-speed download times.
- **Easy Maintenance:** To create a global alteration we need to alter the style. Every element will get automatically updated within all web pages.

Cont…

- **Superior HTML Styles:** HTML contains few extended attribute’s array than CSS, thus we can provide a much better view to our HTML page as compared to HTML attributes.
- **Save Time:** We can specify CSS once and reuse the same sheet within various HTML pages. We can describe a style for every HTML tag and apply this style to as a lot of web pages as we want.
- **Compatibility:** The cascading style sheet permits content to be upgraded for one or more device types. Distinct versions of the website could be granted for various handheld devices like cell phones and PDAs, and for printing.
- **Global Standards:** Now, the attributes of HTML are being recommended to apply CSS, and they are being deprecated. Thus it is better to begin the use of CSS in every HTML page for making them compatible for future browsers.

Changing of the text color



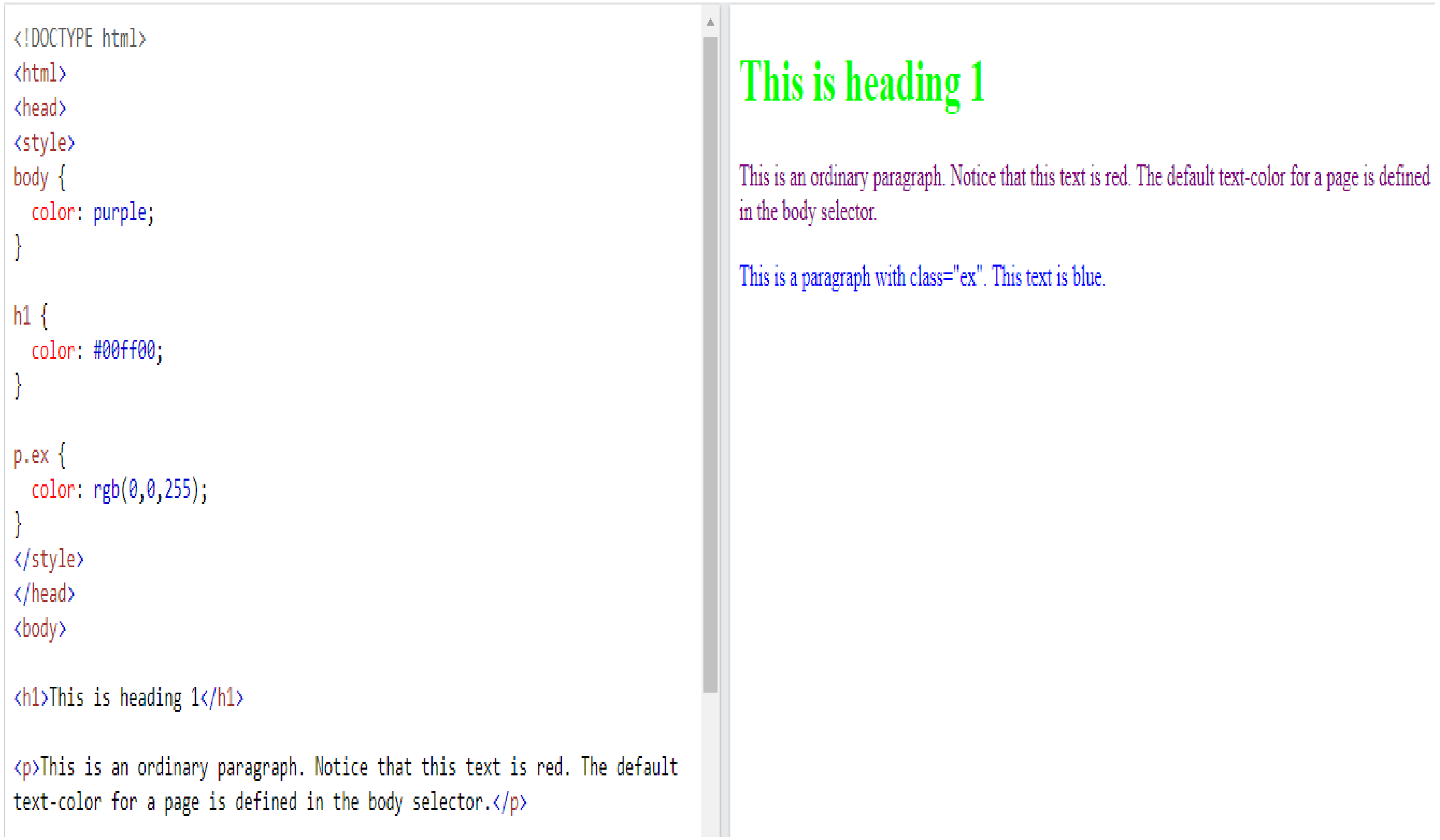
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  color: red;
}

h1 {
  color: #00ff00;
}

p.ex {
  color: rgb(0,0,255);
}
</style>
</head>
<body>

<h1>This is heading 1</h1>

<p>This is an ordinary paragraph. Notice that this text is red. The default
text-color for a page is defined in the body selector.</p>
```

This is heading 1

This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.

This is a paragraph with class="ex". This text is blue.

Cont…



```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  color: purple;
}

h1 {
  color: #00ff00;
}

p.ex {
  color: rgb(0,0,255);
}
</style>
</head>
<body>

<h1>This is heading 1</h1>

<p>This is an ordinary paragraph. Notice that this text is red. The default
text-color for a page is defined in the body selector.</p>
```

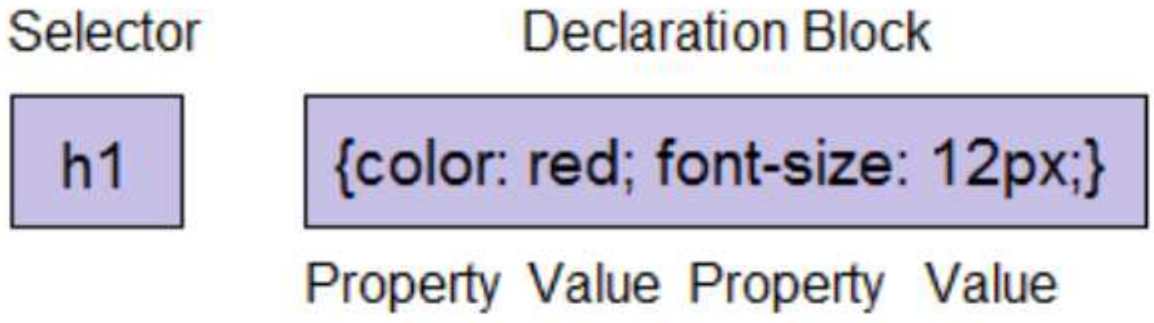
This is heading 1

This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.

This is a paragraph with class="ex". This text is blue.

CSS Syntax

- The CSS syntax contains the selector and the declaration block, which will be clearer with the following example:
- **Selector:** A CSS selector specifies the HTML tag we wish to style. A selector can be any element such as <**title**>, and <**h1**>.



Syntax

Declaration Block: This block can consist of more than one declaration that is isolated by the semicolon. Here, there are two types of declarations for the example as mentioned above:

Color: red;

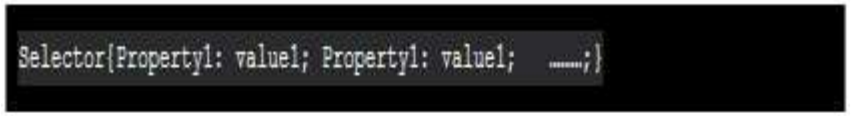
Font-size: 12px;

All the declarations include the property name and the value that are isolated by the colon.

Cont…

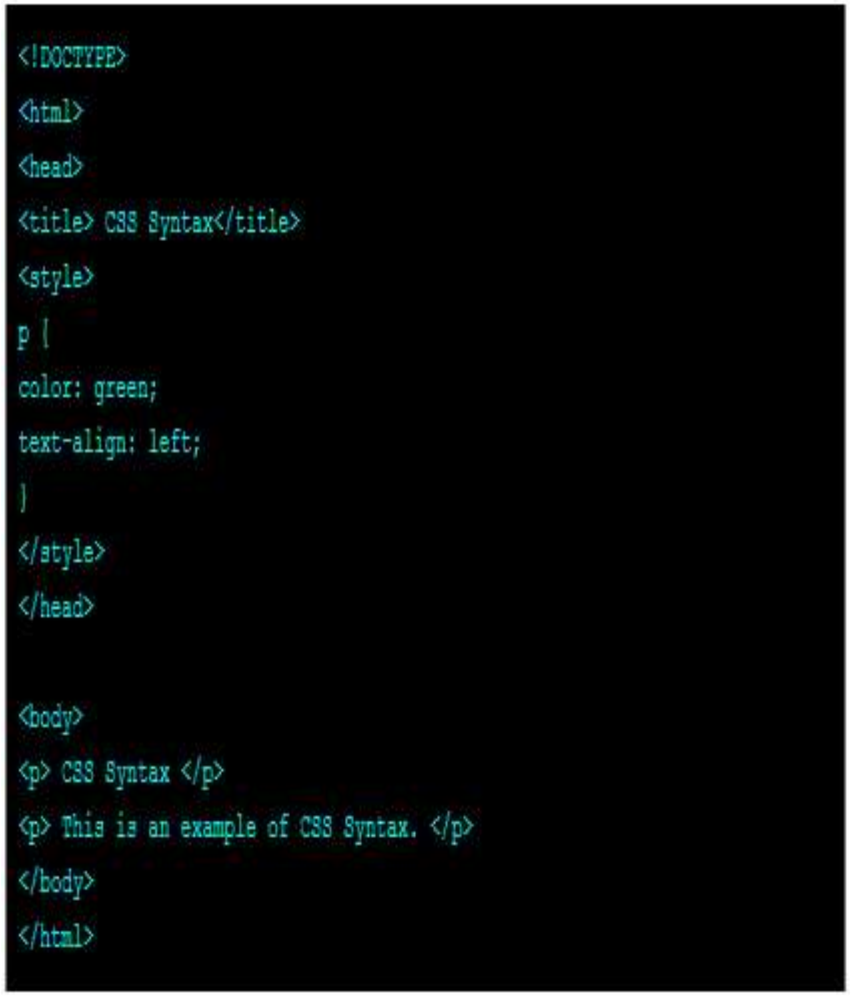
Property: It is a part of the HTML element’s attributes. A property could be any color or border.

Value: CSS properties are required to be assigned by some values. Thus, values are provided to the properties of CSS. Value “red” is given to the color property in the example, as mentioned above.

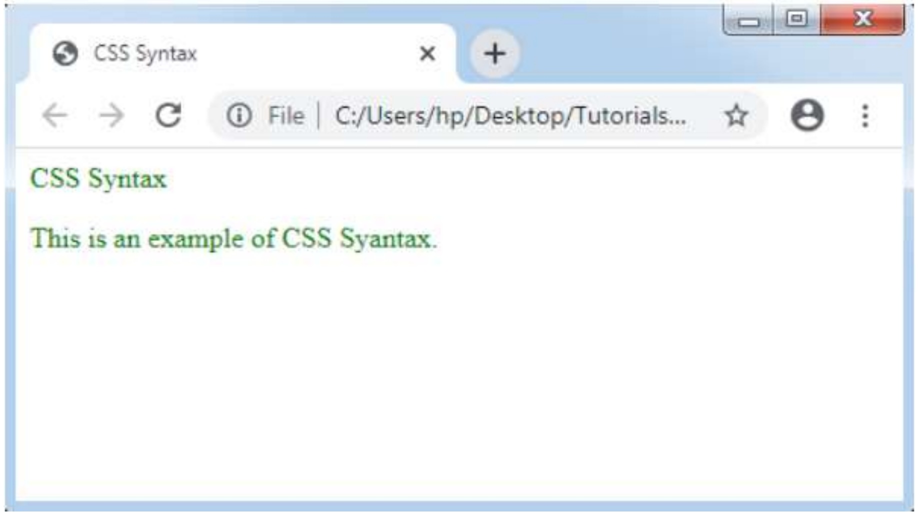


```
Selector{Property1: value1; Property1: value1;   ……;}
```

Example:



```
<!DOCTYPE>
<html>
<head>
<title> CSS Syntax</title>
<style>
p {
color: green;
text-align: left;
}
</style>
</head>

<body>
<p> CSS Syntax </p>
<p> This is an example of CSS Syntax. </p>
</body>
</html>
```



CSS Syntax

This is an example of CSS Syantax.

- **Explanation of Example:**
- P is the selector.
- Color is the property.
- The value of the property is green.
- Another property is text-align.
- The value of the property is left.

TEXT Features

Name of text feature	Purpose of text feature
Title	Quickly tells the reader what information they will learn about
Table of contents	Shows students the different chapter or section titles and where they are located
Index	Directs students where to go in the text to find specific information on a topic, word, or person
Glossary	Identifies important vocabulary words for students and gives their definitions
Headings or subtitles	Help the reader identify the main idea for that section of text

CSS Layout - The position Property

- The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

The position Property

- The position property specifies the type of positioning method used for an element.

There are five different position values:

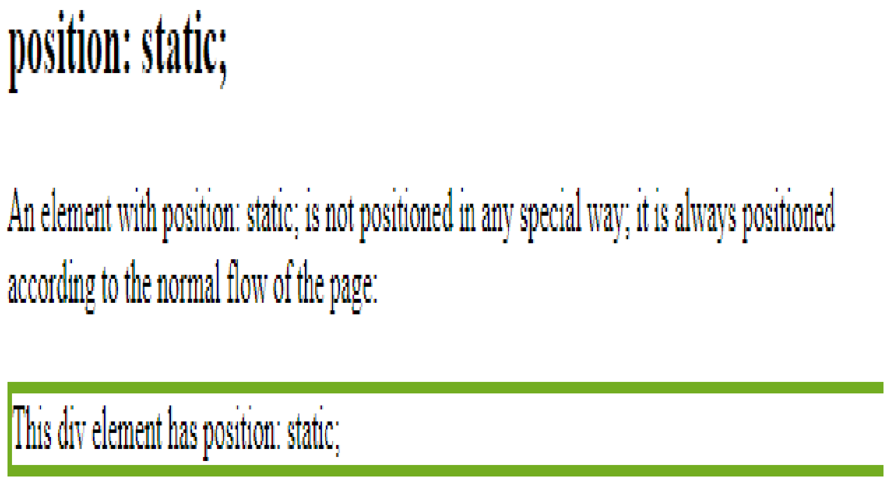
- static
- relative
- fixed
- absolute
- sticky

position: static;

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

Cont…

```
<!DOCTYPE html>
<html>
<head>
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: static;</h2>

<p>An element with position: static; is not positioned in any special way;
it is always positioned according to the normal flow of the page:</p>

<div class="static">
This div element has position: static;
</div>

</body>
</html>
```



position: static;

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

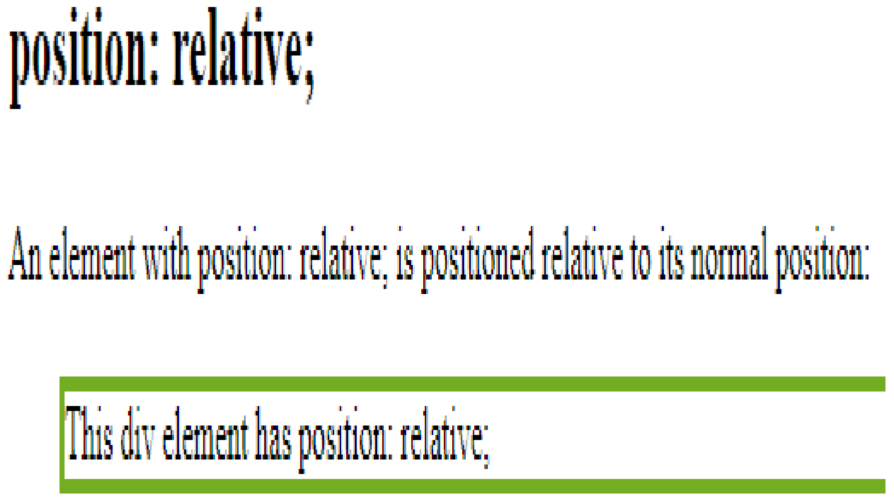
This div element has position: static;

position: relative;

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

Cont…

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: relative;</h2>

<p>An element with position: relative; is positioned relative to its normal
position:</p>

<div class="relative">
This div element has position: relative;
</div>

</body>
</html>
```



position: relative;

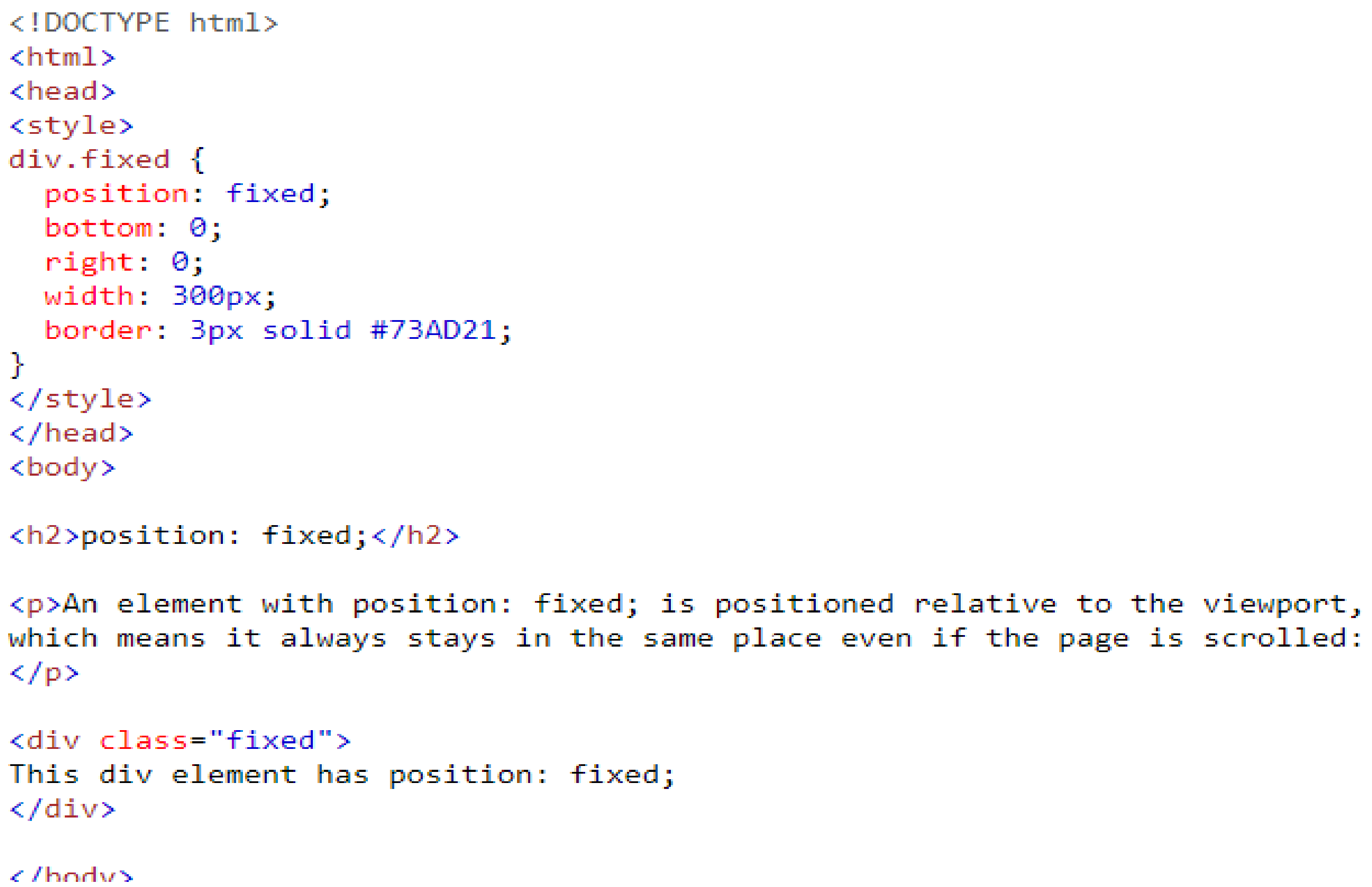
An element with position: relative; is positioned relative to its normal position:

This div element has position: relative;

position: fixed;

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.
- Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

Cont…



```
<!DOCTYPE html>
<html>
<head>
<style>
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: fixed;</h2>

<p>An element with position: fixed; is positioned relative to the viewport,
which means it always stays in the same place even if the page is scrolled:
</p>

<div class="fixed">
This div element has position: fixed;
</div>

</body>
```

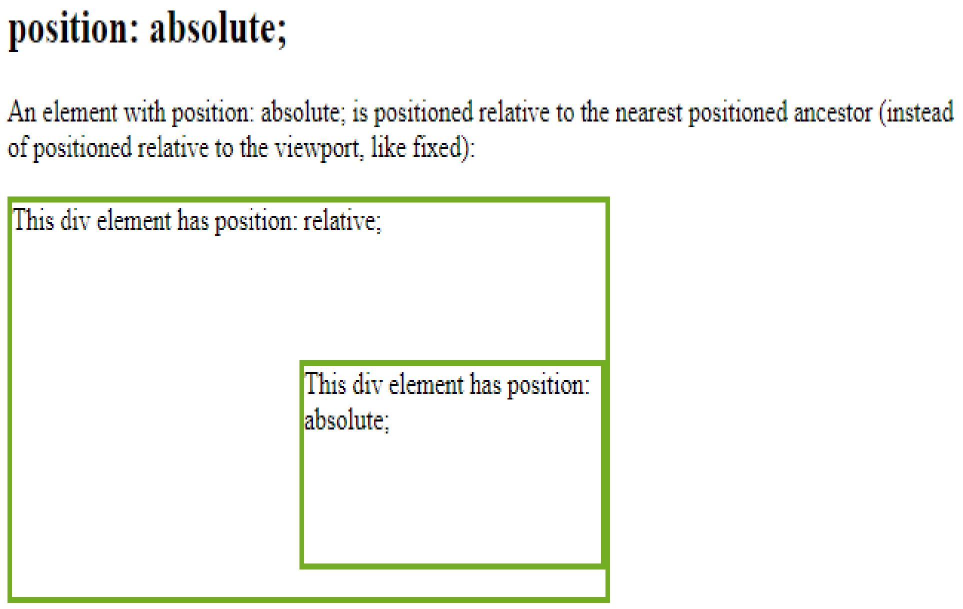
position: absolute;

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
- **Note:** Absolute positioned elements are removed from the normal flow, and can overlap elements.

Cont…

- <!DOCTYPE html>
- <html>
- <head>
- <style>
- div.relative {
-   position: relative;
-   width: 400px;
-   height: 200px;
-   border: 3px solid #73AD21;
- }

- div.absolute {
-   position: absolute;
-   top: 80px;
-   right: 0;
-   width: 200px;
-   height: 100px;
-   border: 3px solid #73AD21;
- }



position: absolute;

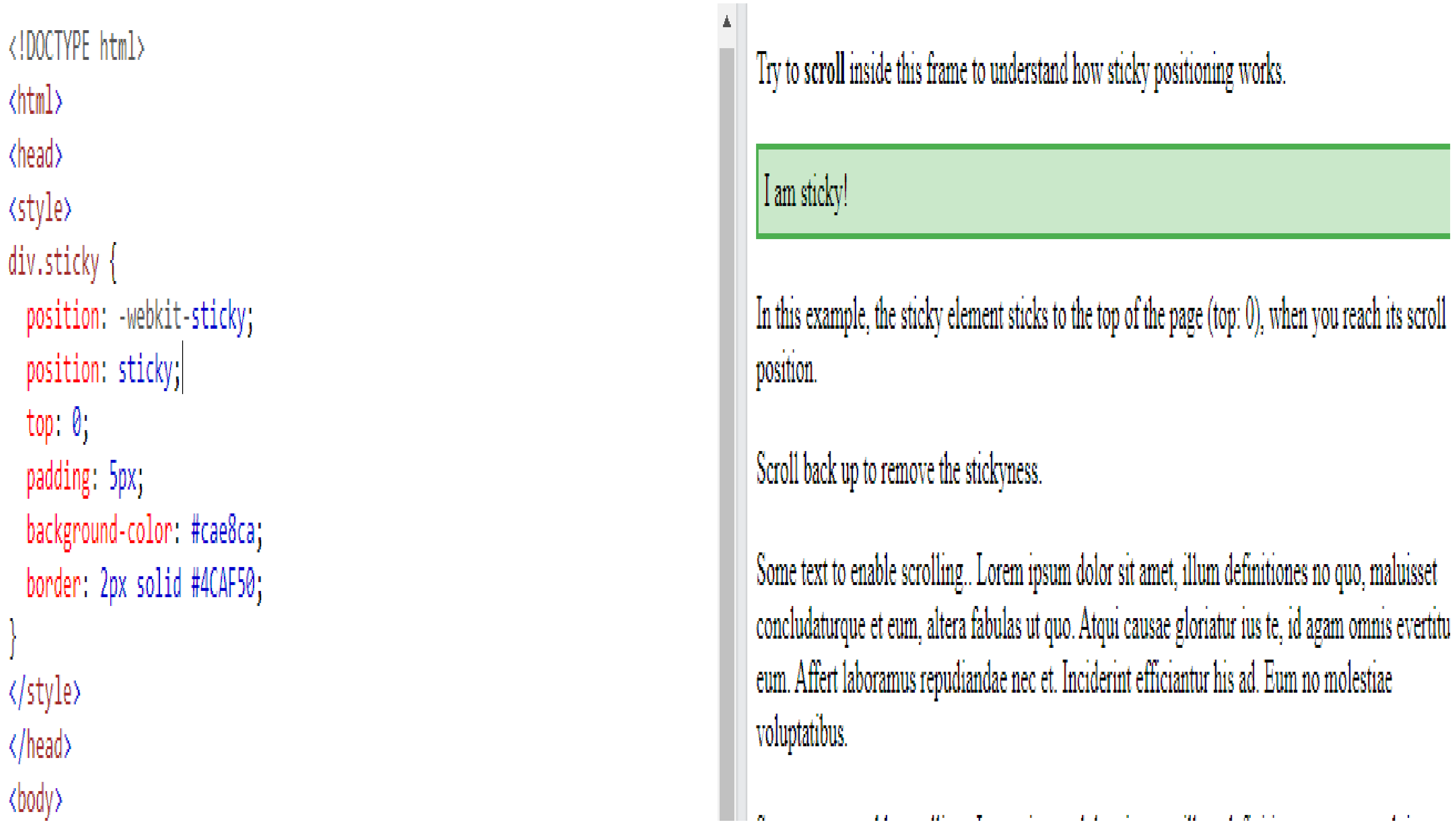
An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

This div element has position: relative;

This div element has position: absolute;

position: sticky;

- An element with position: sticky; is positioned based on the user's scroll position.
- A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).



```
<!DOCTYPE html>
<html>
<head>
<style>
div.sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #cae8ca;
  border: 2px solid #4CAF50;
}
</style>
</head>
<body>
```

Try to **scroll** inside this frame to understand how sticky positioning works.

I am sticky!

In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

Scroll back up to remove the stickyness.

Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitu eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Text on image

```html
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
}

.topright {
  position: absolute;
  top: 8px;
  right: 16px;
  font-size: 18px;
}

img {
  width: 100%;
  height: auto;
  opacity: 0.3;
}
</style>
</head>
<body>
```

```html
<h2>Image Text</h2>
<p>Add some text to an image in the top right corner:</p>

<div class="container">
  <img src="img_5terre_wide.jpg" alt="Cinque Terre" width="1000"
height="300">
  <div class="topright">Top Right</div>
</div>

</body>
</html>
```

Output

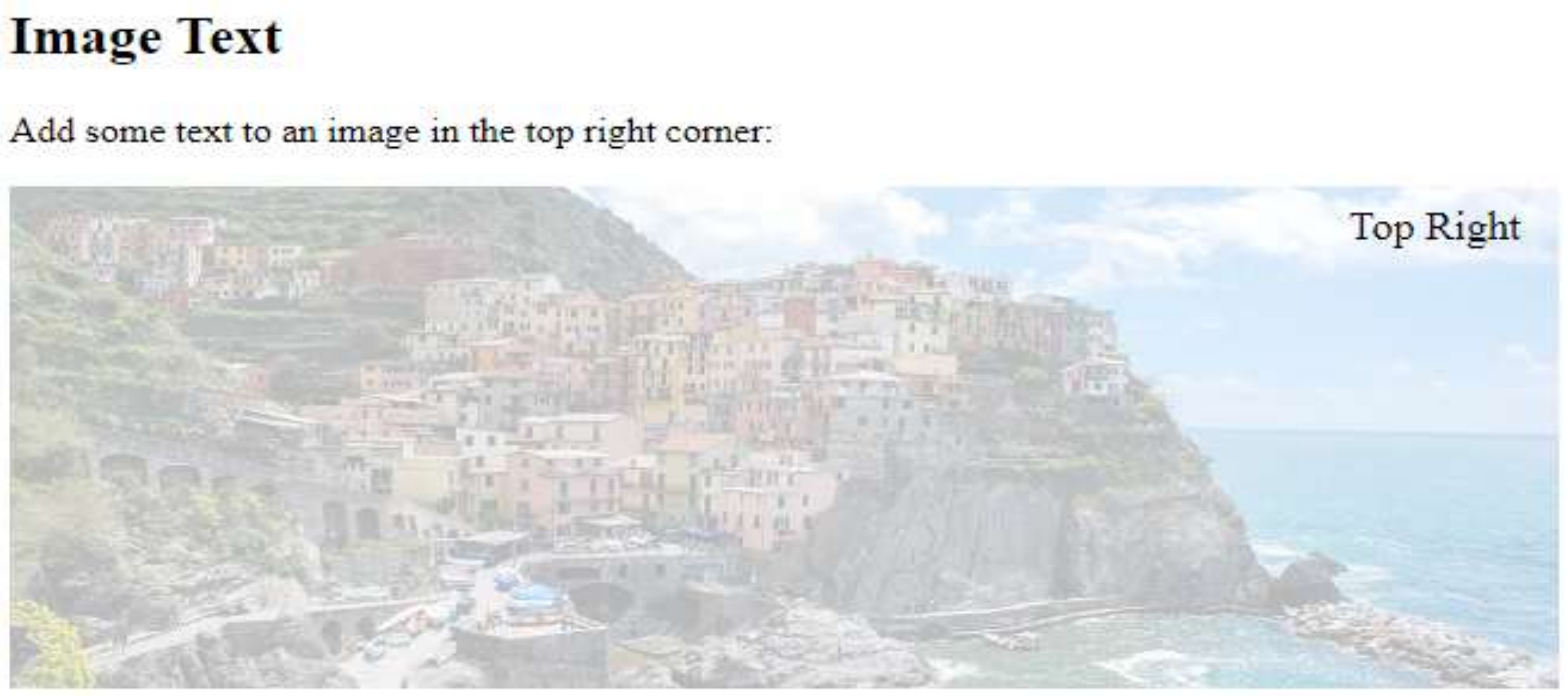


Image Text

Add some text to an image in the top right corner:

Top Right

Positioning Properties

Property	Description
bottom	Sets the bottom margin edge for a positioned box
clip	Clips an absolutely positioned element
left	Sets the left margin edge for a positioned box
position	Specifies the type of positioning for an element
right	Sets the right margin edge for a positioned box
top	Sets the top margin edge for a positioned box

What are CSS Animations?

- An animation lets an element gradually change from one style to another.
- You can change as many CSS properties you want, as many times as you want.
- To use CSS animation, you must first specify some keyframes for the animation.
- Keyframes hold what styles the element will have at certain times.

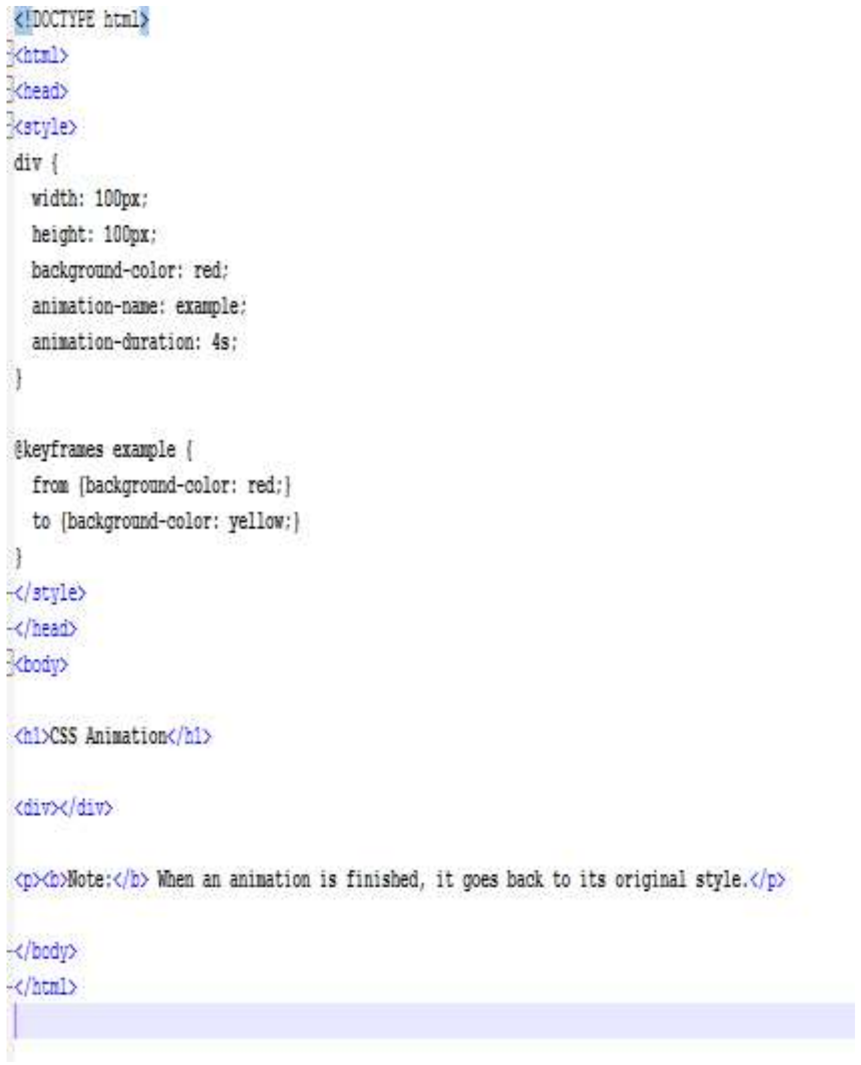
properties:

- @keyframes
- animation-name
- animation-duration
- animation-delay
- animation-iteration-count
- animation-direction
- animation-timing-function
- animation-fill-mode

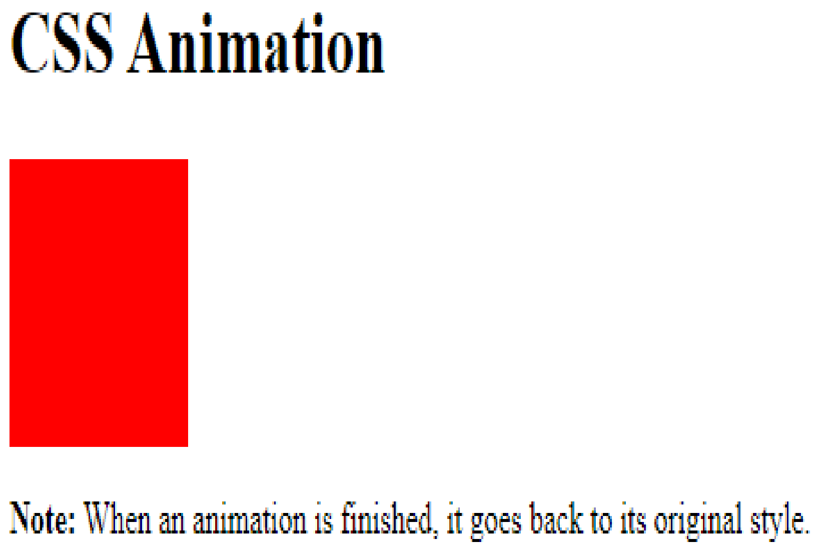
The @keyframes Rule

- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
- To get an animation to work, you must bind the animation to an element.
- The following example binds the "example" animation to the <div> element. The animation will last for 4 seconds, and it will gradually change the background-color of the <div> element from "red" to "yellow":

CSS Animation



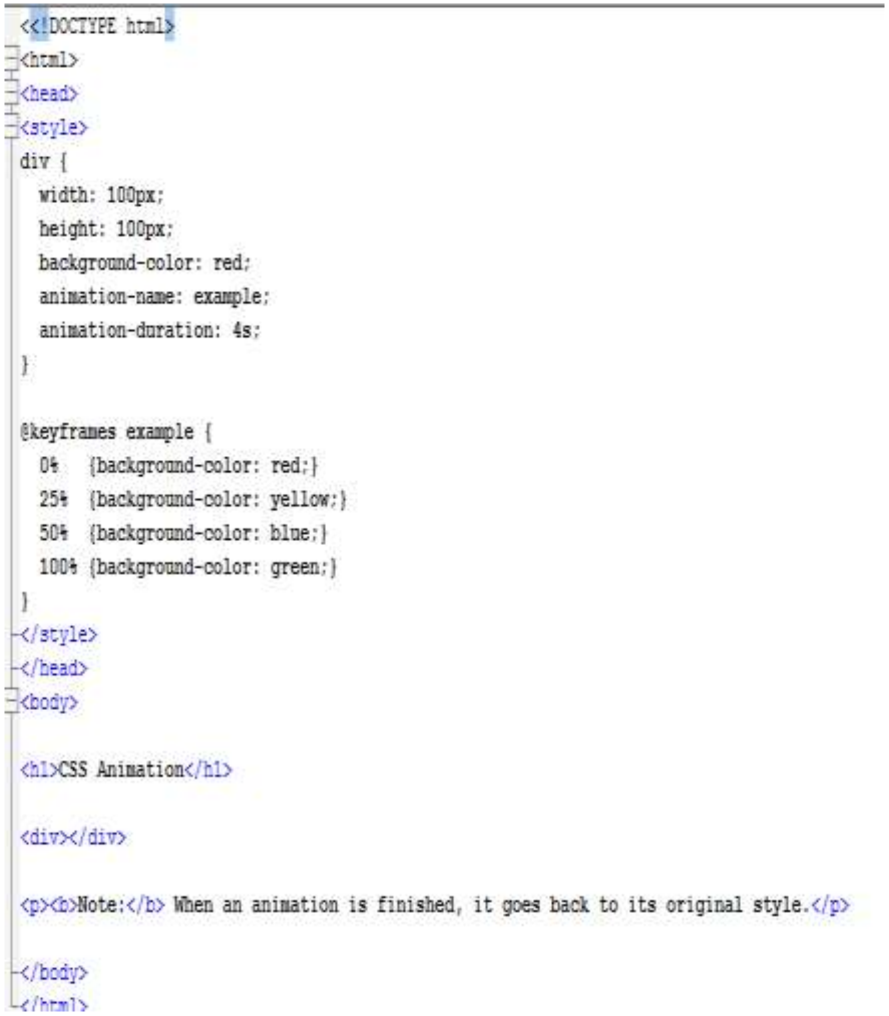
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<div></div>

<p><b>Note:</b> When an animation is finished, it goes back to its original style.</p>

</body>
</html>
```



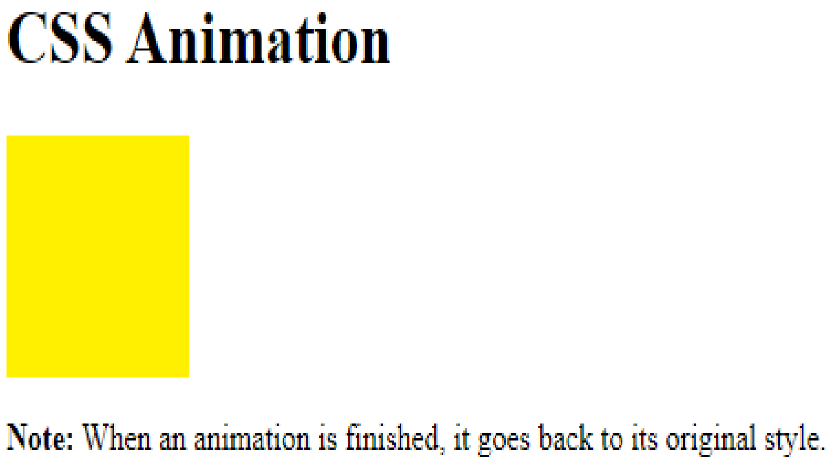
CSS Animation

Note: When an animation is finished, it goes back to its original style.

animation-duration



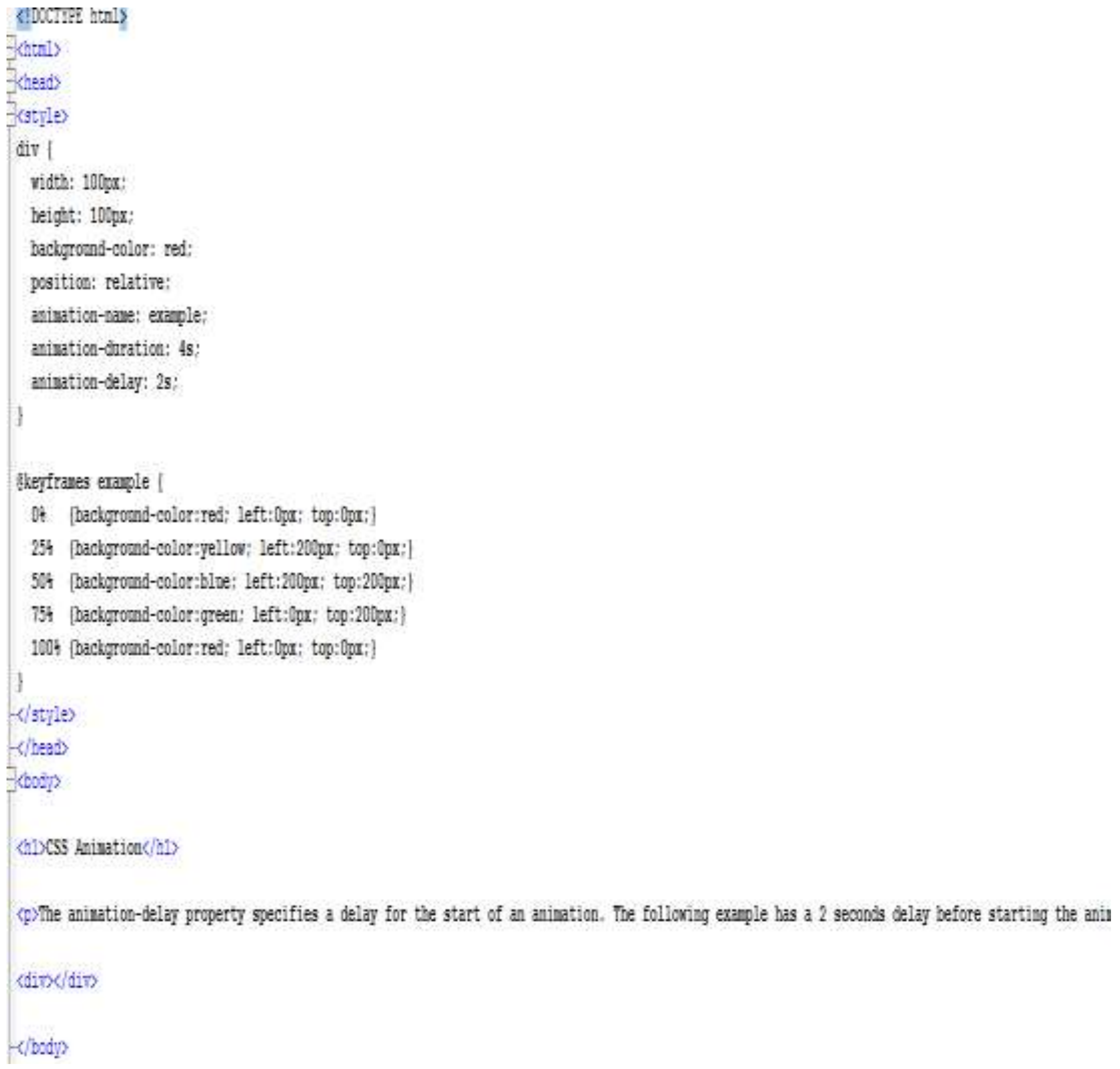
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<div></div>

<p><b>Note:</b> When an animation is finished, it goes back to its original style.</p>

</body>
</html>
```



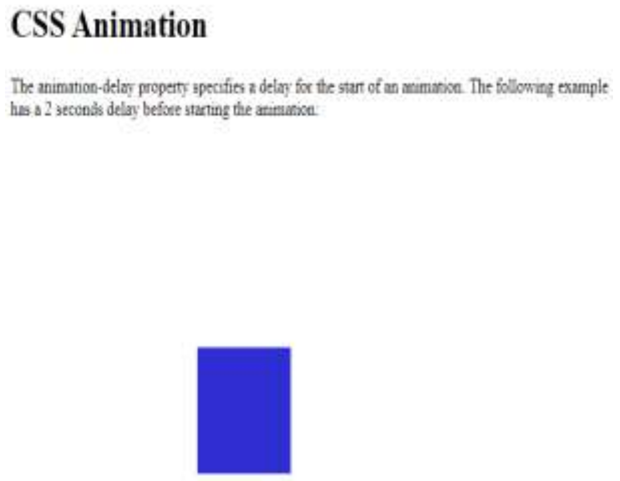
CSS Animation

Note: When an animation is finished, it goes back to its original style.

animation-delay



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
  animation-delay: 2s;
}

@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>

<h1>CSS Animation</h1>

<p>The animation-delay property specifies a delay for the start of an animation. The following example has a 2 seconds delay before starting the ani

<div></div>

</body>
```



CSS Animation

The animation-delay property specifies a delay for the start of an animation. The following example has a 2 seconds delay before starting the animation.

Run Animation in Reverse Direction or Alternate Cycles

- The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.
- The animation-direction property can have the following values:
- normal - The animation is played as normal (forwards). This is default
- reverse - The animation is played in reverse direction (backwards)
- alternate - The animation is played forwards first, then backwards
- alternate-reverse - The animation is played backwards first, then forwards

Example

-
  ```
  div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-direction: reverse;
  }
  ```

Specify the Speed Curve of the Animation

- The animation-timing-function property specifies the speed curve of the animation.
- The animation-timing-function property can have the following values:
- ease - Specifies an animation with a slow start, then fast, then end slowly (this is default)
- linear - Specifies an animation with the same speed from start to end
- ease-in - Specifies an animation with a slow start
- ease-out - Specifies an animation with a slow end
- ease-in-out - Specifies an animation with a slow start and end
- cubic-bezier(n,n,n,n) - Lets you define your own values in a cubic-bezier function

Example

- #div1 {animation-timing-function: linear;}
  #div2 {animation-timing-function: ease;}
  #div3 {animation-timing-function: ease-in;}
  #div4 {animation-timing-function: ease-out;}
  #div5 {animation-timing-function: ease-in-out;}

Specify the fill-mode For an Animation

- CSS animations do not affect an element before the first keyframe is played or after the last keyframe is played. The animation-fill-mode property can override this behavior.
- The animation-fill-mode property specifies a style for the target element when the animation is not playing (before it starts, after it ends, or both).
- The animation-fill-mode property can have the following values:
- none - Default value. Animation will not apply any styles to the element before or after it is executing
- forwards - The element will retain the style values that is set by the last keyframe (depends on animation-direction and animation-iteration-count)
- backwards - The element will get the style values that is set by the first keyframe (depends on animation-direction), and retain this during the animation-delay period
- both - The animation will follow the rules for both forwards and backwards, extending the animation properties in both directions

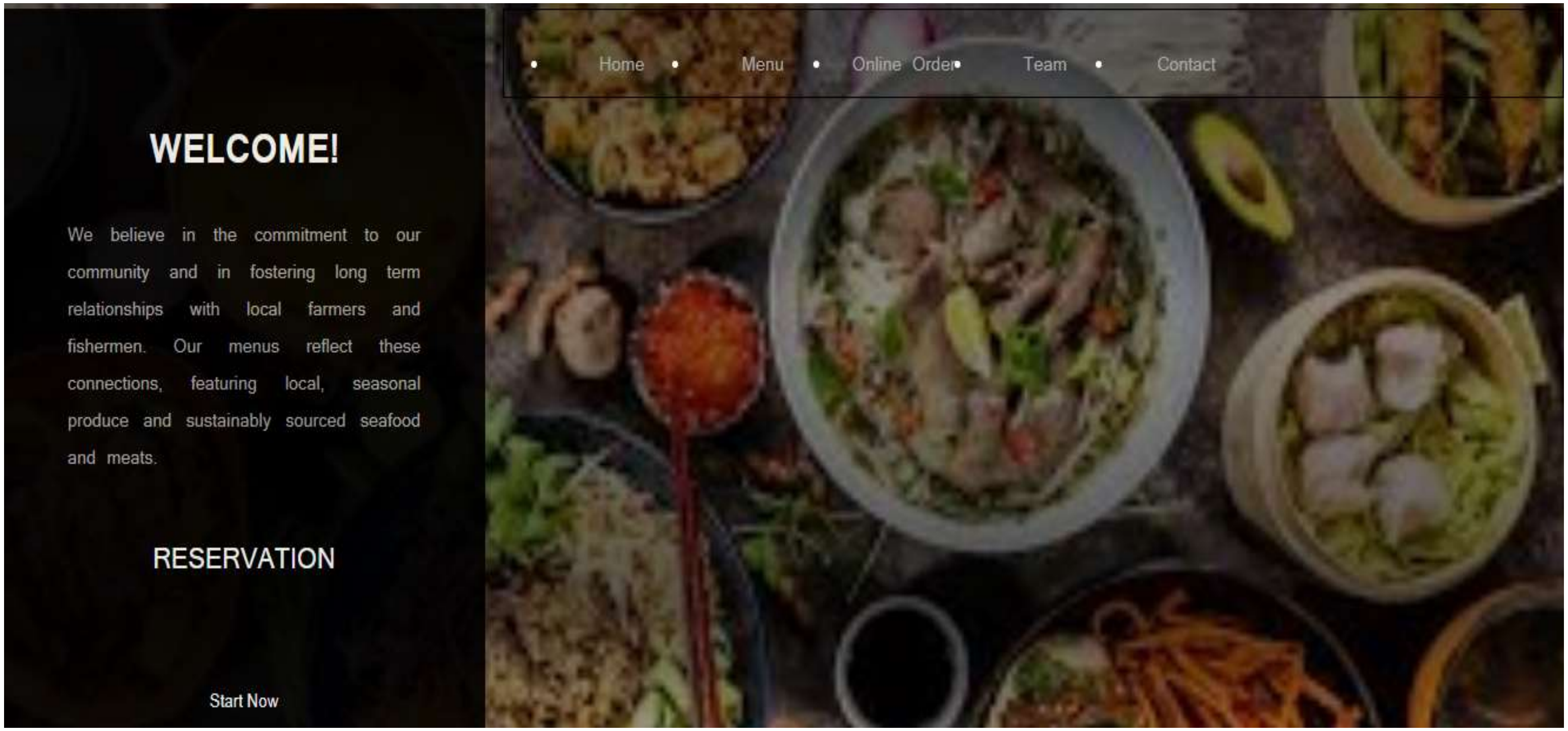
Animation Shorthand Property

```
div {
    animation-name: example;
    animation-duration: 5s;
    animation-timing-function: linear;
    animation-delay: 2s;
    animation-iteration-count: infinite;
    animation-direction: alternate;
  }
```

CSS Properties

Property	Description
@keyframes	Specifies the animation code
animation	A shorthand property for setting all the animation properties
animation-delay	Specifies a delay for the start of an animation
animation-direction	Specifies whether an animation should be played forwards, backwards or in alternate cycles
animation-duration	Specifies how long time an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
animation-iteration-count	Specifies the number of times an animation should be played
animation-name	Specifies the name of the @keyframes animation
animation-play-state	Specifies whether the animation is running or paused

Creating Website



- Home
- Menu
- Online Order
- Team
- Contact

WELCOME!

We believe in the commitment to our community and in fostering long term relationships with local farmers and fishermen. Our menus reflect these connections, featuring local, seasonal produce and sustainably sourced seafood and meats.

RESERVATION

Start Now

Questions

What is CSS?
What are the uses of CSS?
What are the features of CSS?
What are the application of CSS?
What is the syntax of CSS?
Explain selector in CSS?
What are different text features IN CSS?
Explain position property in CSS?
Explain CSS Animation properties?

References

1. https://www.tutorialspoint.com/css/css_syntax.htm
2. https://slideplayer.com/slide/14361379/
3. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Values_and_Units
4. https://www.digitalocean.com/community/tutorials/how-to-lay-out-text-with-css
5. https://www.w3schools.com/