SmartCliff

CAREER MOBILITY SOLUTIONS

We are on a mission to address the digital skills gap for 10 Million+ young professionals, train and empower them to forge a career path into future tech

Cascading Style Sheets

JUNE, 2023

Introduction

- **Cascading Style Sheets** (CSS) is a **style sheet language** used for describing the **look** and **formatting** of a document written in a **markup language**.
- Used to describe the **presentation of documents**
- Define sizes, spacing, fonts, colors, layout.
- Improve content **accessibility** and **flexibility**.
- It is a highly effective **HTML tool** that **provides easy control over layout** and **presentation** of website pages by **separating content from design**.
- Typical CSS file is a text file with an extension **.css** and contains a series of **commands** or **rules**.
- These rules tell the HTML **how to display**.

Introduction

- Developed to enable the separation of document content from document presentation
- Initial release in **1996** (CSS1).
- CSS2 published as a recommendation in May **1998** and builds on CSS1.
- CSS3 recommendation in June **1999** and builds on older versions CSS.

HTML without CSS

- “HTML without CSS is like a piece of candy without a pretty wrapper.”
- Without CSS, HTML elements typically flow from top to bottom of the page and position themselves to the left by default.
- With CSS help, we can create containers or DIVs to better organize content and make a **Web page visually appealing**.

HTML
(Structure)

HTML + CSS
(Presentation)

CSS

- CSS handles the **look and feel part of a web page.**
- Using CSS, you can **control**
  - **the color of the text**
  - **the style of fonts**
  - **the spacing between paragraphs**
  - **how columns are sized and laid out**
  - **what background images or colors are used**
  - **layout designs** and
  - **variations in the display for different devices and screen sizes**

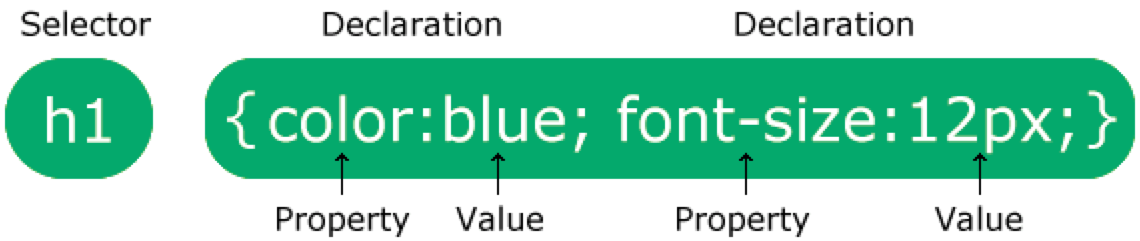
  as well as a variety of other effect

CSS Benefits

- Separates structure from presentation
- Provides advanced control of presentation
- Easy maintenance of multiple pages
- Faster page loading
- Better accessibility for disabled users
- Easy to learn

Style Sheets Syntax

- Style sheets consist of **rules, selectors, declarations, properties and values.**
- Selectors are separated by **commas.**
- Declarations are separated by **semicolons.**
- Properties and values are separated by **colons.**



```
h1 {color:blue; font-size:12px;}
```

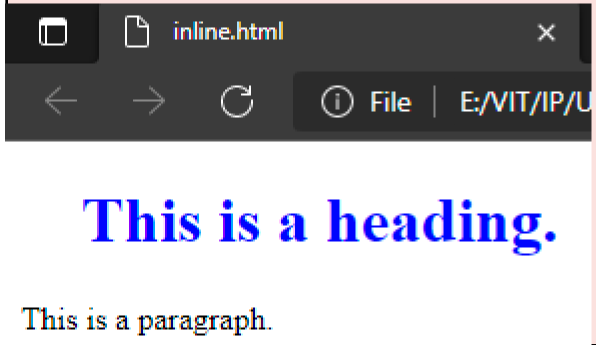
Attaching Style sheet

- HTML (content) and CSS (presentation) can be linked in **three ways**:
- **Inline**: the CSS rules in the **style attribute**
  - No selectors are needed
- **Embedded/Internal**: in the <head> in a <style> tag
- **External**: CSS rules in **separate file** (best)
- Usually a file with **.css** extension
- Linked via <link rel="stylesheet" href=…> tag or @import directive in embedded CSS block
- **Example**: @import url(sheet2.css);

Attaching Style sheet: Inline

- An inline style may be used to **apply** a unique style for a **single element**.
- An inline style loses many of the advantages of a style sheet (by mixing content with presentation). **Use this method sparingly!**
- To use inline styles, add the style **attribute** to the relevant tag.
- The style attribute can contain **any CSS property**.

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
<p>This is a paragraph.</p>
```



This is a heading.

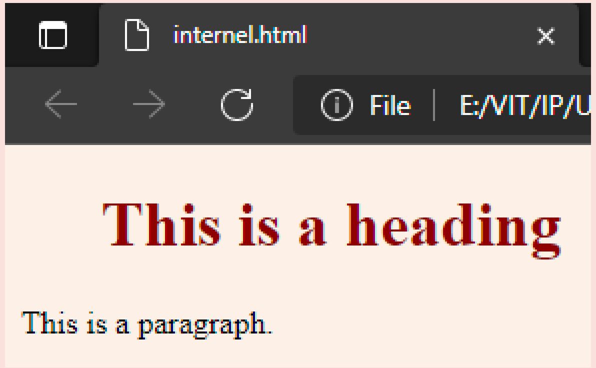
This is a paragraph.

Attaching Style sheet: Internal

- An **internal style** sheet may be used if **one single page** has a **unique style**.
- Internal styles are defined within the **<style> element**, **inside the head section** of an HTML page
- The <style> tag is used to define style information for an HTML document.
- Inside the <style> element you specify **how HTML elements** should render in a browser.

Attaching Style sheet: Internal

```
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```



This is a heading

This is a paragraph.

Attaching Style sheet: External

- With an **external style sheet**, you can change the look of an **entire website** by changing just **one file**.
- Each page must include a **reference** to the external style sheet file inside the **<link> element**. The <link> element goes inside the head section
- An external style sheet can be written in any text editor. The file should not contain any html tags.
- The style sheet file must be saved with a **.css extension**.
- Using external files is **highly recommended.**
- Simplifies the HTML document.
- **Improves page load speed** as the CSS file is cached.

Attaching Style sheet: External

- The **<link>** tag defines a **link** between a **document** and an **external resource**.
- The <link> tag is used to link to **external style sheets**.

rel

- Specifies the relationship between the current document and the linked document

type

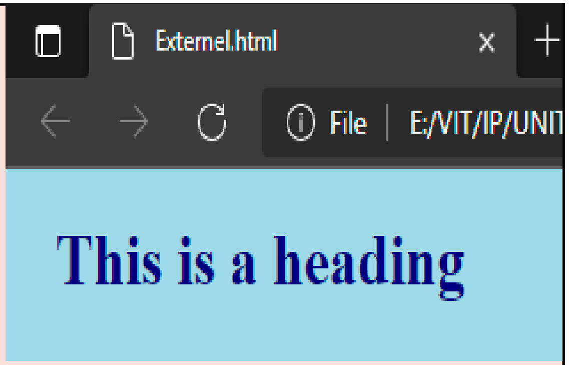
- Specifies the media type of the linked document

href

- Specifies the location of the linked document

Attaching Style sheet: External

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
```



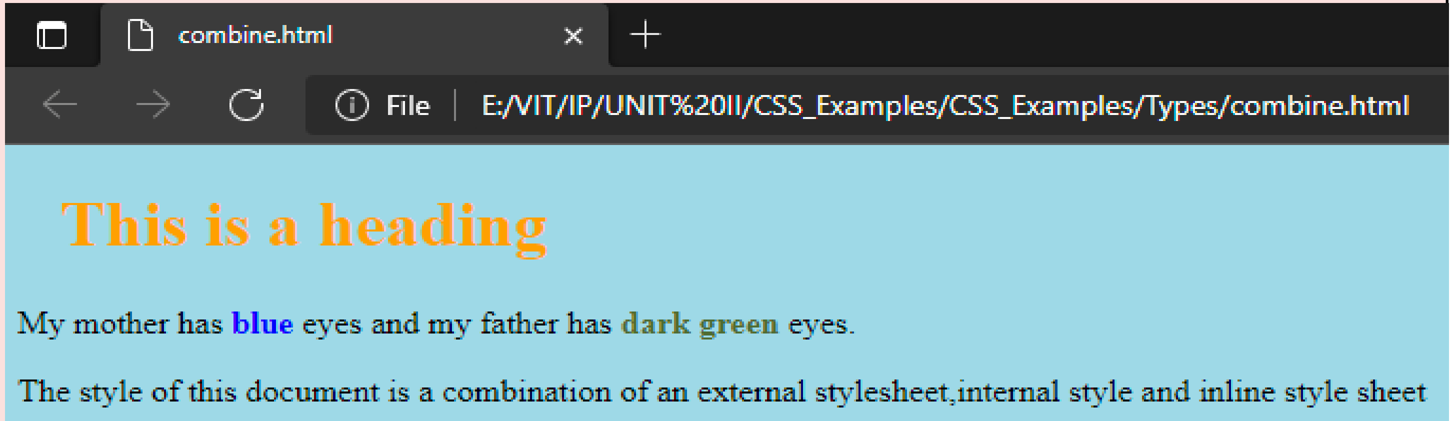
This is a heading

mystyle.css

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```

Attaching Style sheet: Combined

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
   color: orange;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and my father has <span
style="color:darkolivegreen;font-weight:bold">dark green</span> eyes.</p>
<p>The style of this document is a combination of an external style sheet, internal style and inline style sheet</p>
```


combine.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Types/combine.html

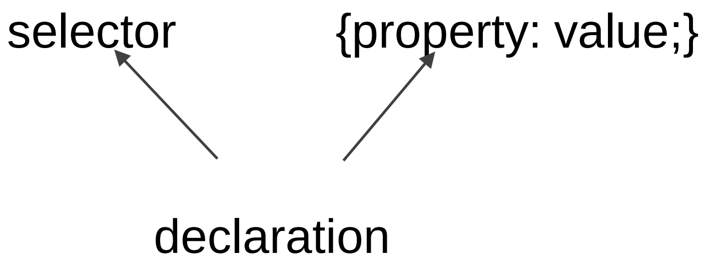
This is a heading

My mother has **blue** eyes and my father has **dark green** eyes.

The style of this document is a combination of an external stylesheet,internal style and inline style sheet

CSS Rule Structure

- A CSS RULE is made up of a **selector** and a **declaration**.
- A declaration consists of **property and value**.



selector {property: value;}

declaration

Selectors

- Selectors **determine which element the rule applies to**
- All elements of **specific type** (tag)
- Those that mach a **specific attribute** (id, class)
- Elements may be matched depending on **how they are nested** in the document tree (HTML).

Examples:

```
h1 {color: green; }
```

↓

Selectors

Properties and Values

- Properties and values tell an HTML element **how to display.**

```
body {background: purple;}

h1 {color: green; }

h2 {font-size: large;}

p {color: #ff0000;}

/*hexadecimal for red*/
```

- CSS code can be written in a **linear format** or in a **block format.**

```
body {

background: purple;

color: green;

}
```

Grouping Selectors

- **Group the same selector** with different declarations together on one line.
- **Example** of grouping selectors (both are correct):

```
h1 {color: black;}

h1 {font-weight: bold;}

h1 {background: white;}
```

OR

```
h1 {

color: black;

font-weight: bold;

background: white;

}
```

Grouping Selectors

- **Group different selectors** with the same declaration on one line.
- **Example** of grouping selectors (both are correct):

```
h1 {color: yellow;}

h2 {color: yellow;}

h3 {color: yellow;}
```

OR

```
h1, h2, h3 {color: yellow;}
```

Grouping Selectors

Three primary kinds of selectors:

By tag (type selector): **h1 { font-family: verdana,sans-serif; }**

By element id (#): **#element_id { color: #ff0000; }**

By element class name (.): **.myClass {border: 1px solid red}**

Selectors can be combined with commas:

h1, .link, #top-link {font-weight: bold}

This will match <h1> tags, elements with class link, and element with id top-link

ID’s and Classes

- **ID's are unique**
  - Each element can have only one ID
  - Each page can have only one element with that ID
- **Classes are NOT unique**
  - You can use the same class on multiple elements.
  - You can use multiple classes on the same element.
  - Any styling information that needs to be applied to multiple objects on a page should be done with a class.

Selectors

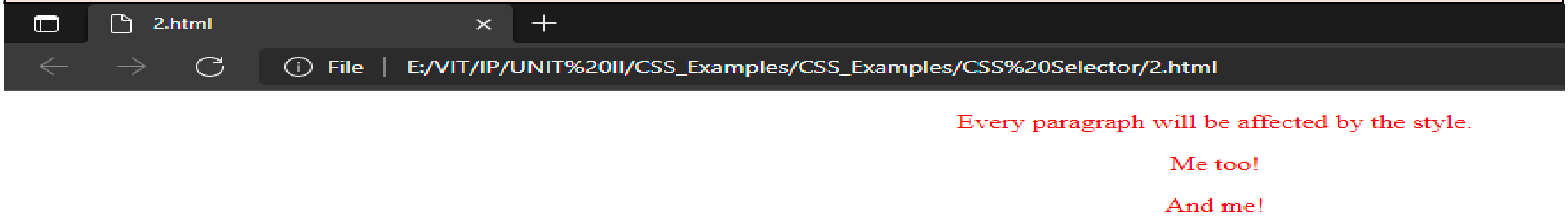
Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
.class1.class2	.name1.name2	Selects all elements with both *name1* and *name2* set within its class attribute
.class1 .class2	.name1 .name2	Selects all elements with *name2* that is a descendant of an element with *name1*
#id	#firstname	Selects the element with id="firstname"
*****	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"
element,element	div, p	Selects all <div> elements and all <p> elements
element element	div p	Selects all <p> elements inside <div> elements

Selectors

Selector	Example	Example description
element>element	div > p	Selects all <p> elements where the parent is a <div> element
element+element	div + p	Selects the first <p> element that is placed immediately after <div> elements
element1~element2	p ~ ul	Selects every <ul> element that is preceded by a <p> element
[*attribute*]	[target]	Selects all elements with a target attribute
:active	a:active	Selects the active link
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
:checked	input:checked	Selects every checked <input> element
More Selectors: https://www.w3schools.com/cssref/css_selectors.asp		

Selectors: Example #1

```
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

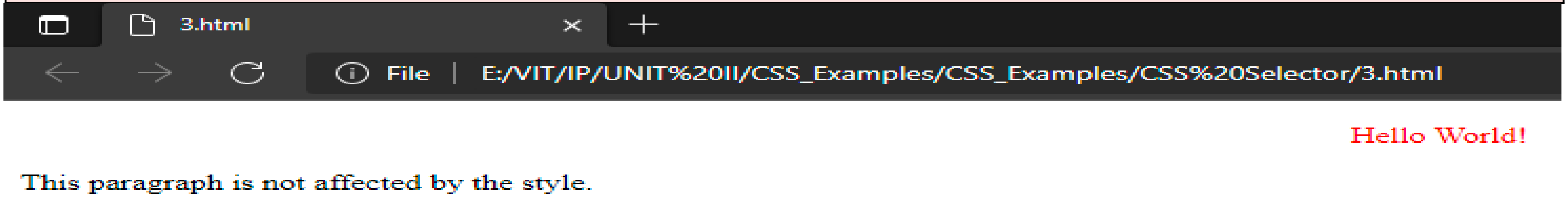


E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS%20Selector/2.html

```
Every paragraph will be affected by the style.

Me too!

And me!
```

Selectors: Example #2

```
<head>
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
```

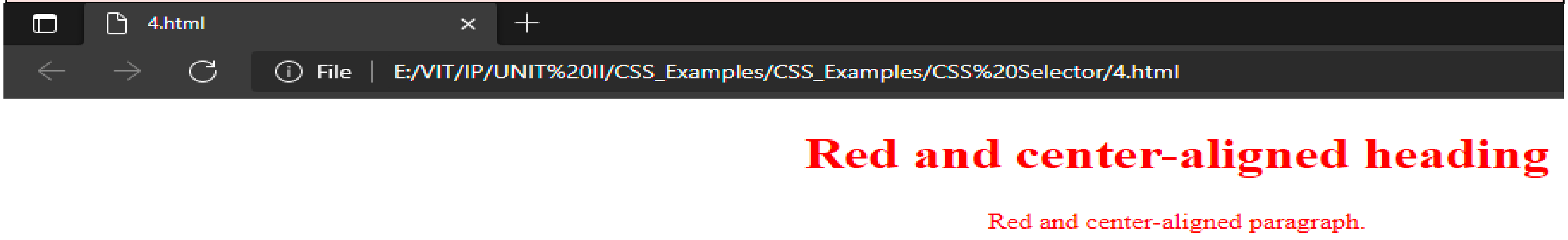


3.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS%20Selector/3.html

```
Hello World!

This paragraph is not affected by the style.
```

Selectors: Example #3

```
<head>
<style>
.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
```

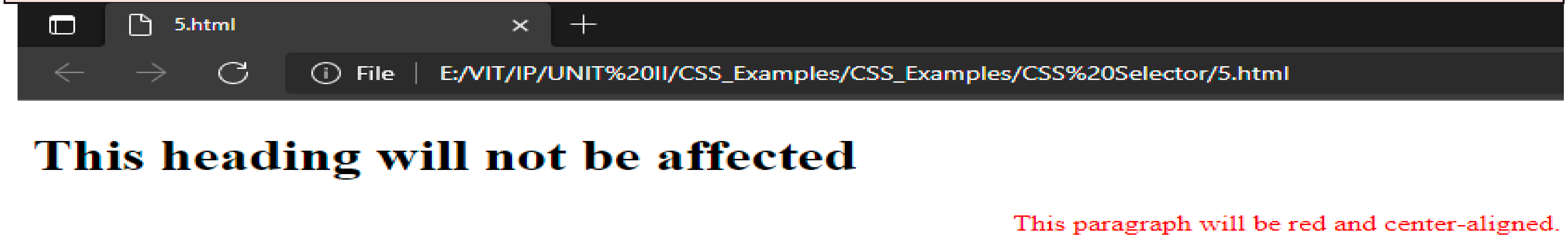

4.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS%20Selector/4.html

Red and center-aligned heading

Red and center-aligned paragraph.

Selectors: Example #4

```
<head>
<style>
p.center {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
```

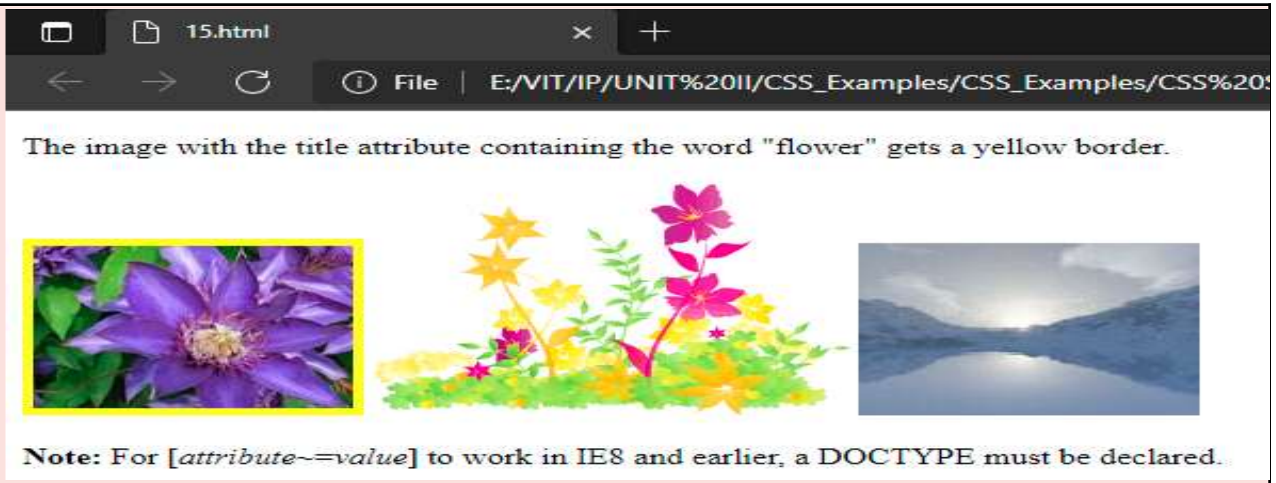

5.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS%20Selector/5.html

This heading will not be affected

This paragraph will be red and center-aligned.

Selectors: Example #5

```
<head>
<style>
[title~=flower]
{
border:5px solid yellow;
}
</style>
</head>
<body>

<p>The image with the title attribute containing the word "flower" gets a yellow border.</p>

<img src="klematis.jpg" title="klematis flower" width="150" height="113" />
<img src="img_flwr.gif" title="flowers" width="224" height="162" />
<img src="landscape.jpg" title="landscape" width="160" height="120" />

<p><b>Note:</b> For [<i>attribute</i>~=<i>value</i>] to work in IE8 and earlier, a DOCTYPE must be declared.</p>
```

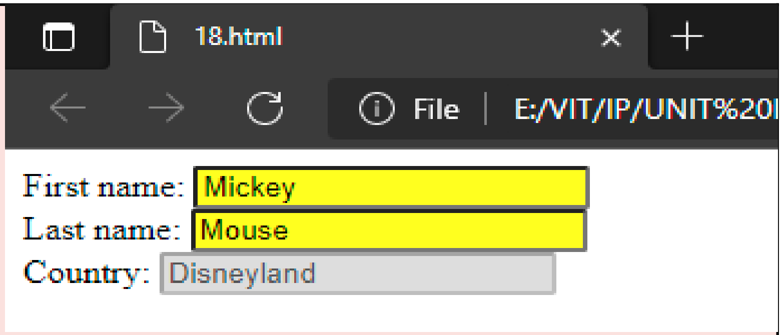


The image with the title attribute containing the word "flower" gets a yellow border.

Note: For [*attribute*~=*value*] to work in IE8 and earlier, a DOCTYPE must be declared.

Selectors: Example #6

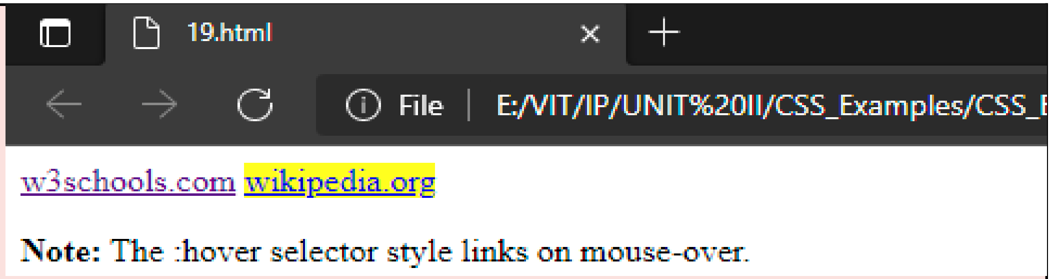
```
<head>
<style>
input[type=text]:enabled {
    background: #ffff00;
}

input[type=text]:disabled {
    background: #dddddd;
}
</style>
</head>
<body>

<form action="">
First name: <input type="text" value="Mickey"><br>
Last name: <input type="text" value="Mouse"><br>
Country: <input type="text" disabled="disabled" value="Disneyland">
</form>
```



First name: Mickey
Last name: Mouse
Country: Disneyland

Selectors: Example #7

```
<head>
<style>
a:hover
{
background-color:yellow;
}
</style>
</head>

<body>
<a href="http://www.w3schools.com">w3schools.com</a>
<a href="http://www.wikipedia.org">wikipedia.org</a>

<p><b>Note:</b> The :hover selector style links on mouse-over.</p></form>
```

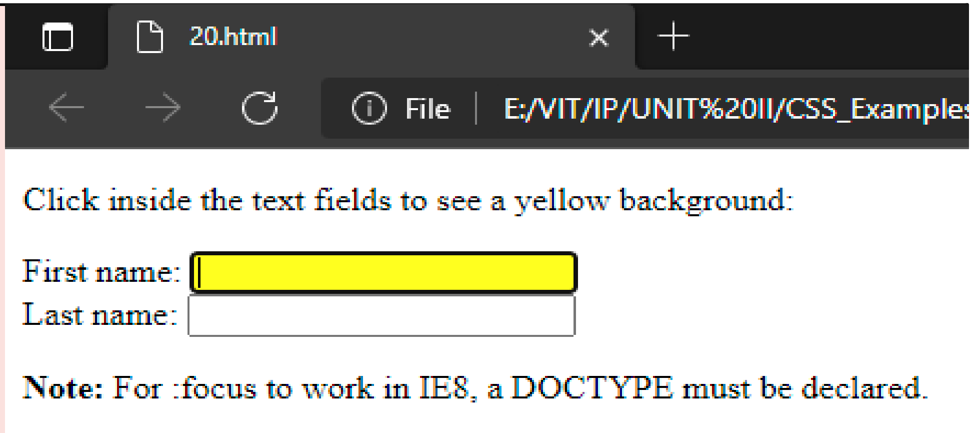


w3schools.com wikipedia.org

Note: The :hover selector style links on mouse-over.

Selectors: Example #8

```
<head>
<style>
input:focus
{
background-color:yellow;
}
</style>
</head>
<body>

<p>Click inside the text fields to see a yellow background:</p>

<form>
First name: <input type="text" name="firstname" /><br>
Last name: <input type="text" name="lastname" />
</form>
```



Click inside the text fields to see a yellow background:

First name:
Last name:

Note: For :focus to work in IE8, a DOCTYPE must be declared.

Comments in CSS

- Explain the purpose of the coding
- Help others read and understand the code
- Serve as a reminder to you for what it all means
- Starts with **/*and ends with*/**

Example:

p {color: #ff0000;} /*Company Branding*/

/* This is a single-line comment */

/* This is

a multi-line

comment */

Values in CSS Rules

- **Colors** are set in **RGB format** (decimal or hex):
  - Example: #a0a6aa = rgb(160, 166, 170)
  - Predefined color aliases exist: black, blue, etc.
- **Numeric values are specified in:**
  - Pixels, ems, e.g. 12px , 1.4em
  - Points, inches, centimeters, millimeters
  - E.g. 10pt , 1in, 1cm, 1mm
  - Percentages, E.g. 50%
  - Zero can be used with no unit: border: 0;

Default Browser Styles

- Browsers have **default CSS styles**
  - Used when there is no CSS information or any other style information in the document
- **Caution**: default styles differ in browsers
- E.g. margins, paddings and font sizes differ most often and usually developers reset them

```
* { margin: 0; padding: 0; }
```

```
body, h1, p, ul, li { margin: 0; padding: 0; }
```

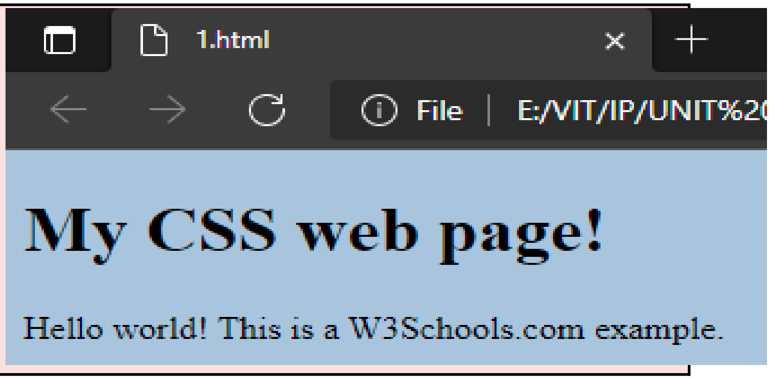
CSS Background

- CSS background properties are used to **define the background effects** of an element.
- **CSS properties** used for background effects:
- background-color
- background-image
  - background-repeat
- background-attachment
- background-position

CSS Background : Color

- The background-color property specifies the **background color** of an element.

```
<head>
<style>
body {
    background-color: #b0c4de;
}
</style>
</head>
```



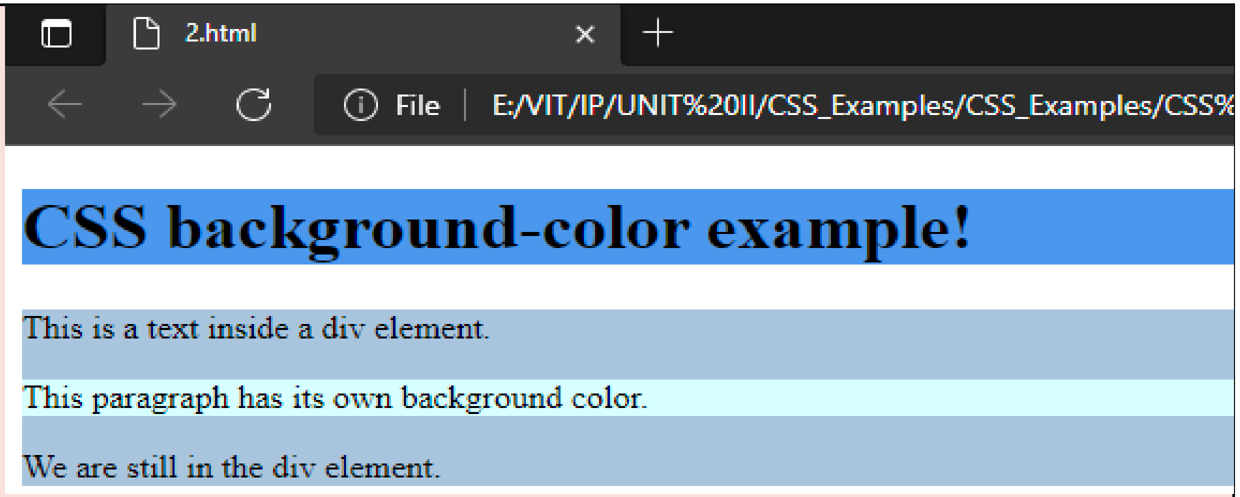
My CSS web page!

Hello world! This is a W3Schools.com example.

CSS Background : Color

- The background-color property specifies the **background color** of an element.

```
<head>
<style>
h1 { background-color: #6495ed; }

p { background-color: #e0ffff; }

div { background-color: #b0c4de; }
</style>
</head>
<body>
<h1>CSS background-color Example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>
```



CSS background-color example!

This is a text inside a div element.

This paragraph has its own background color.

We are still in the div element.

CSS Background: Repeat

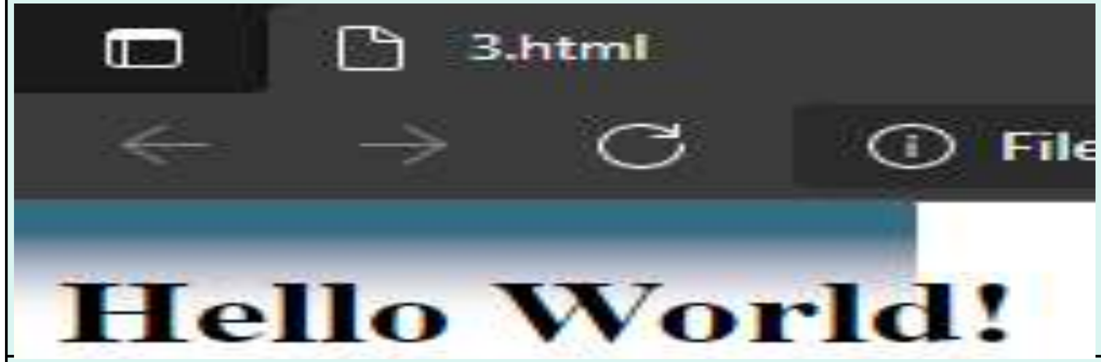
- The background-image property specifies an image to use as the **background of an element.**
- **By default,** a background-image is placed at the top-left corner of an element, and **repeated both vertically and horizontally.**

<style>
body {
    background-image: url("gradient_bg.jpg");
    background-repeat: **no-repeat;**
}
</style>
	<style>
	body {
	    background-image: url("gradient_bg.jpg");
	    background-repeat: **repeat-x;**
	}
	</style>

<style>
body {
    background-image: url("gradient_bg.jpg");
    background-repeat: **repeat-y;**
}
</style>
	<style>
	body {
	    background-image: url("gradient_bg.jpg");
	    /* Default both x and y*/
	}
	</style>

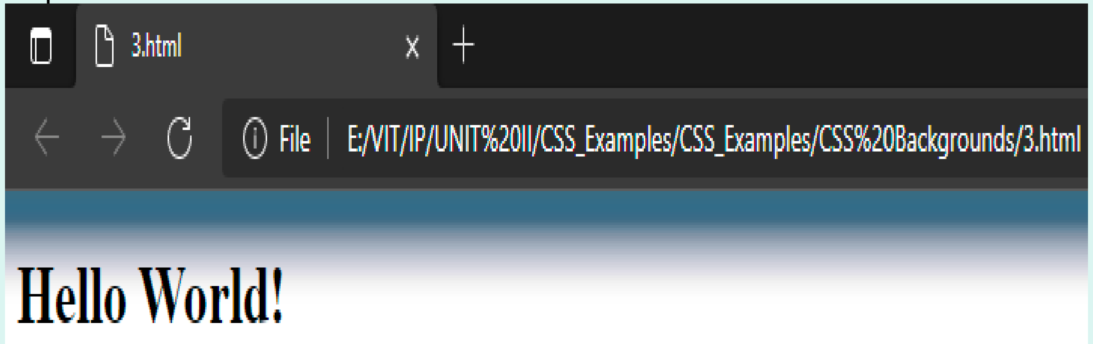
CSS Background : Repeat

background-repeat: no-repeat;



Hello World!

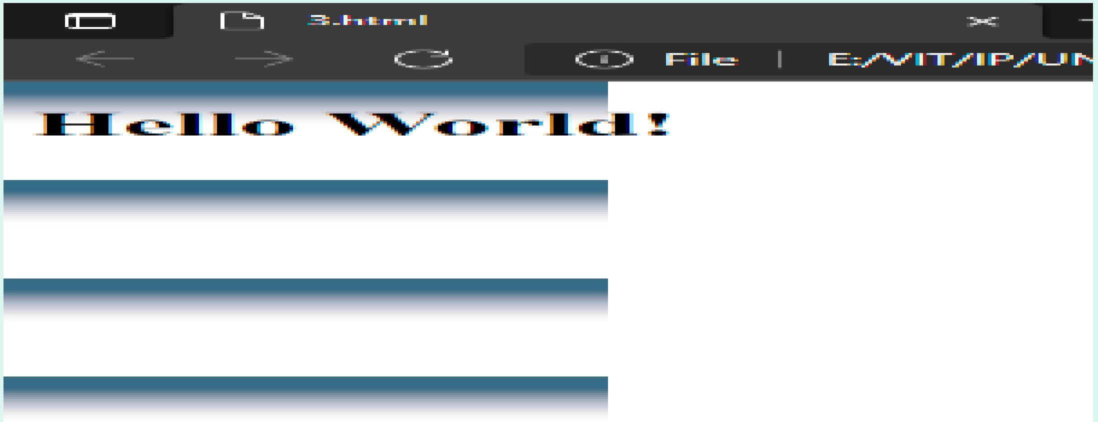
background-repeat: repeat-x;



E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS%20Backgrounds/3.html

Hello World!

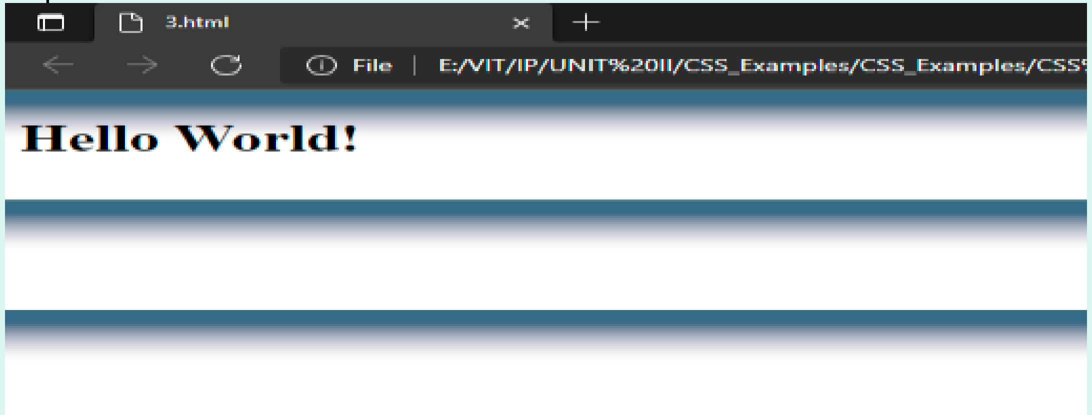
background-repeat: repeat-y;



E:/VIT/IP/UN

Hello World!

Default both x and y

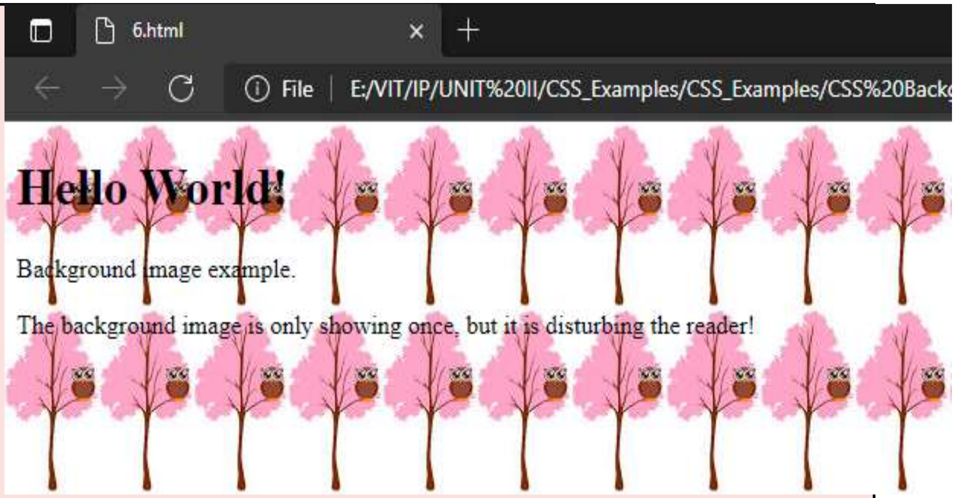


E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS

Hello World!

CSS Background: Repeat

```
<style>
body {
  background-image: url("img_tree.jpg");
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Background image example.</p>
<p>The background image is only showing once, but it is disturbing the reader!</p>
```



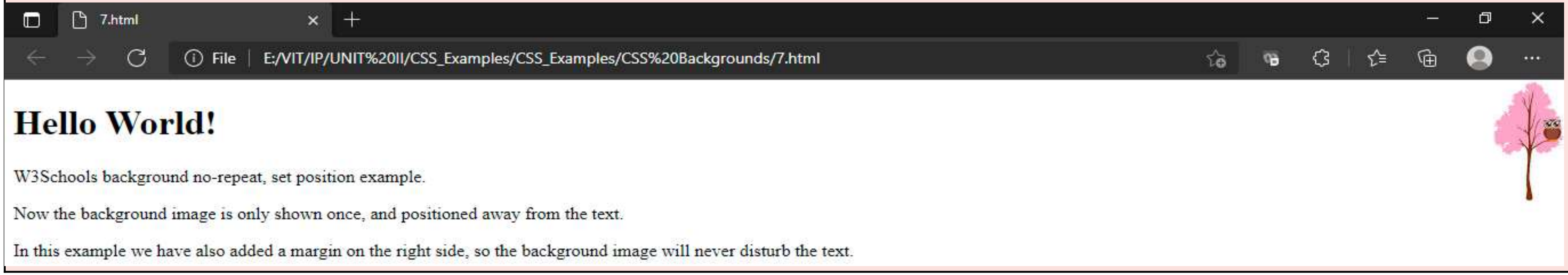
Hello World!

Background image example.

The background image is only showing once, but it is disturbing the reader!

CSS Background: Repeat

```
<style>
body {
   background: #ffffffurl ("img_tree.jpg") no-repeat right top;
   margin-right: 200px;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>Now the background image is only shown once, and it is also positioned away from the text.</p>
<p>In this example we have also added a margin on the right side, so that the background image will not disturb the text.</p>
```



7.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS%20Backgrounds/7.html

Hello World!

W3Schools background no-repeat, set position example.

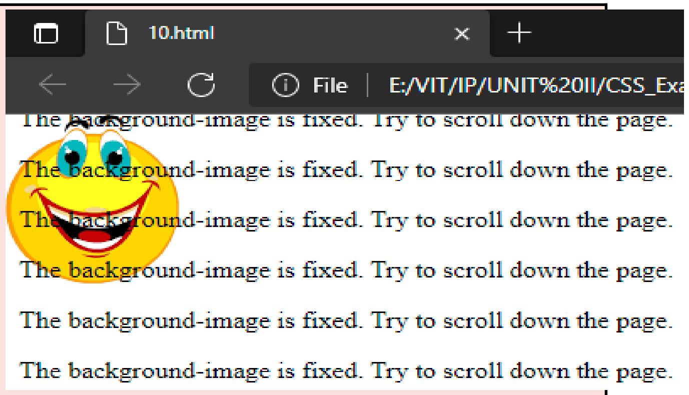
Now the background image is only shown once, and positioned away from the text.

In this example we have also added a margin on the right side, so the background image will never disturb the text.

CSS Background: Attachment

- The **background-attachment** property sets whether a **background image is fixed** or **scrolls** with the rest of the page.

```
<style>
body {
    background-image: url('smiley.png');
    background-repeat: no-repeat;
    background-attachment: fixed;
   //background-attachment: scroll;
}
</style>
</head>

<body>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
```



The background-image is fixed. Try to scroll down the page.

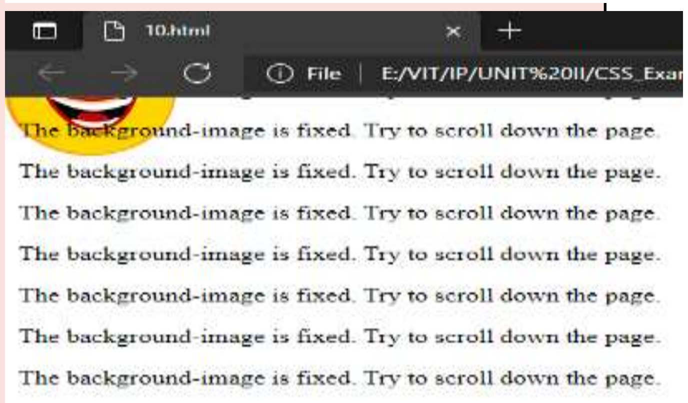
The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.



The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

The background-image is fixed. Try to scroll down the page.

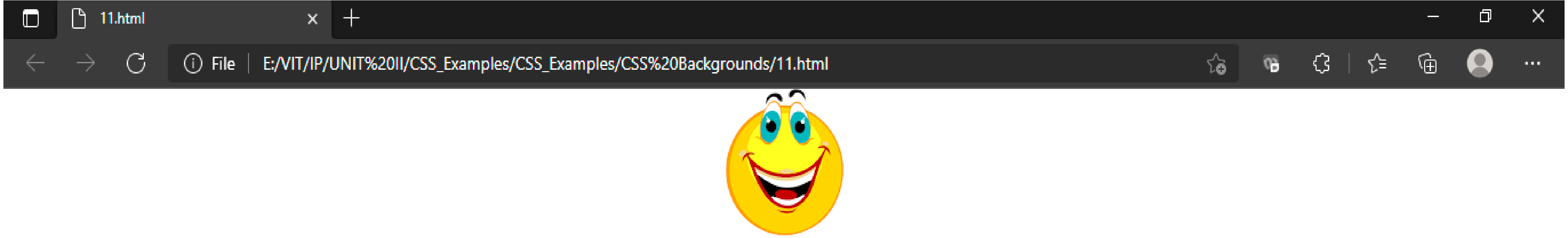
CSS Background: Position

- The **background-position** property sets the starting position of a background image.
- **By default**, a background-image is placed at the **top-left corner of an element**, and **repeated both vertically and horizontally**.

Value	Description
left top, left center, left bottom right top, right center, right bottom center top, center center, center bottom	If you only specify one keyword, the other value will be "center"
x% y%	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. . Default value is: 0% 0%
xpos ypos	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0.

CSS Background: Position

```
<head>
<style>
body {
    background-image: url('smiley.png');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center top;
}
</style>
```

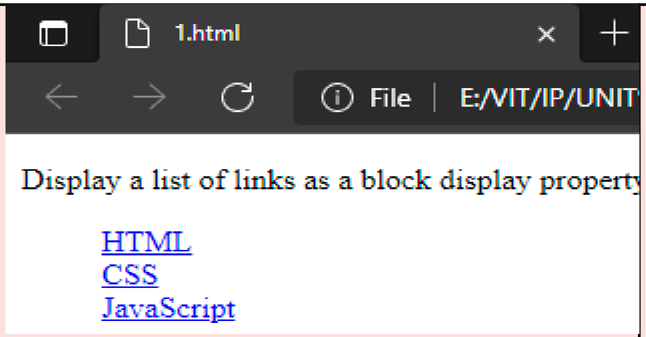

11.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/CSS%20Backgrounds/11.html

CSS Layout : Display Property

- The display property specifies **how an element** is displayed.
- Every HTML element has a **default display value** depending on what type of element it is. The default display value for most elements is **block or inline**.
- A **block-level element** always starts on a new line and takes up the full width available
- An **inline element** does not start on a new line and only takes up as much width as necessary.
- Changing an **inline element to a block element**, or **vice versa**, can be useful for making the page look a specific way.

CSS Layout : Display Property

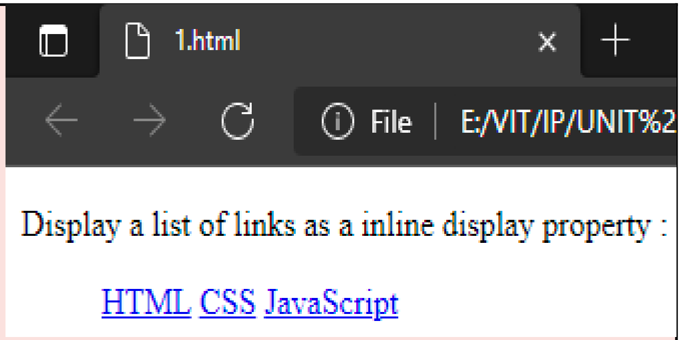
```
<style>
li {
    display: block;
}
</style>
</head>
<body>
<p>Display a list of links as a block display property :</p>
```



Display a list of links as a block display property

HTML
CSS
JavaScript

```
<style>
li {
    display: inline;
}
</style>
</head>
<body>
<p>Display a list of links as a inline display property :</p>
```



Display a list of links as a inline display property :

HTML CSS JavaScript

```
<ul>
<li><a href="/html/default.asp" target="_blank">HTML</a></li>
<li><a href="/css/default.asp" target="_blank">CSS</a></li>
<li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>
```

CSS Layout : float property

- The **float property** is used for **positioning** and **formatting content.**
- The float property can have one of the following **values**:
- **left** - The element floats to the left of its container
- **right**- The element floats to the right of its container
- **none** - The element does not float (will be displayed just where it occurs in the text). This is default
- **inherit** - The element inherits the float value of its parent

CSS Layout : float property

```
<style>
.thumbnail {
    float: none;
    width: 110px;
    height: 90px;
    margin: 5px;
}
</style>
</head>
```

```
<style>
.thumbnail {
    float: left;
    width: 110px;
    height: 90px;
    margin: 5px;
}
</style>
</head>
```

```
<body>
<h3>Image Gallery</h3>
<p>In this example, the image and the text in the paragraph without float property.</p>
<p><img class="thumbnail" src="rose-flower.jpg" width="107" height="90">
The rose is a type of flowering shrub. Its name comes from the Latin word Rosa. The flowers of the rose grow in many
different colors, from the well-known red rose or yellow rose and sometimes white or purple rose. Roses belong to the family
of plants called Rosaceae.</p>
</body>
```

CSS Layout : float property

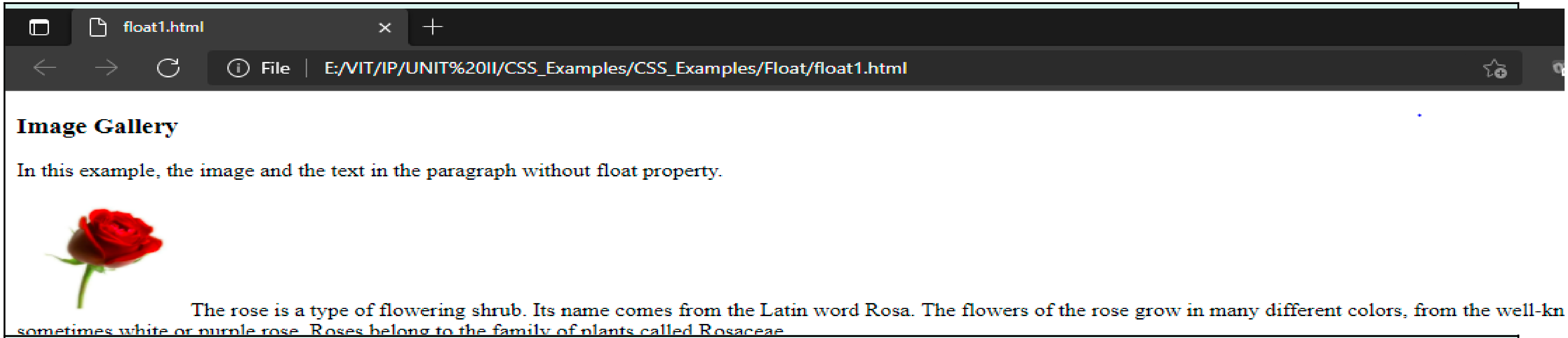

float1.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Float/float1.html

Image Gallery

In this example, the image and the text in the paragraph without float property.

The rose is a type of flowering shrub. Its name comes from the Latin word Rosa. The flowers of the rose grow in many different colors, from the well-kn
sometimes white or purple rose. Roses belong to the family of plants called Rosaceae.

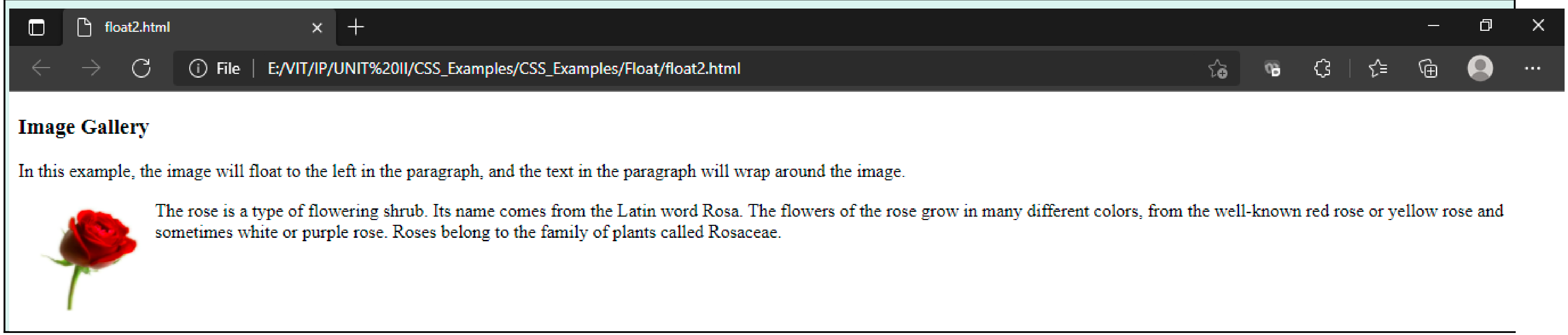

float2.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Float/float2.html

Image Gallery

In this example, the image will float to the left in the paragraph, and the text in the paragraph will wrap around the image.

The rose is a type of flowering shrub. Its name comes from the Latin word Rosa. The flowers of the rose grow in many different colors, from the well-known red rose or yellow rose and sometimes white or purple rose. Roses belong to the family of plants called Rosaceae.

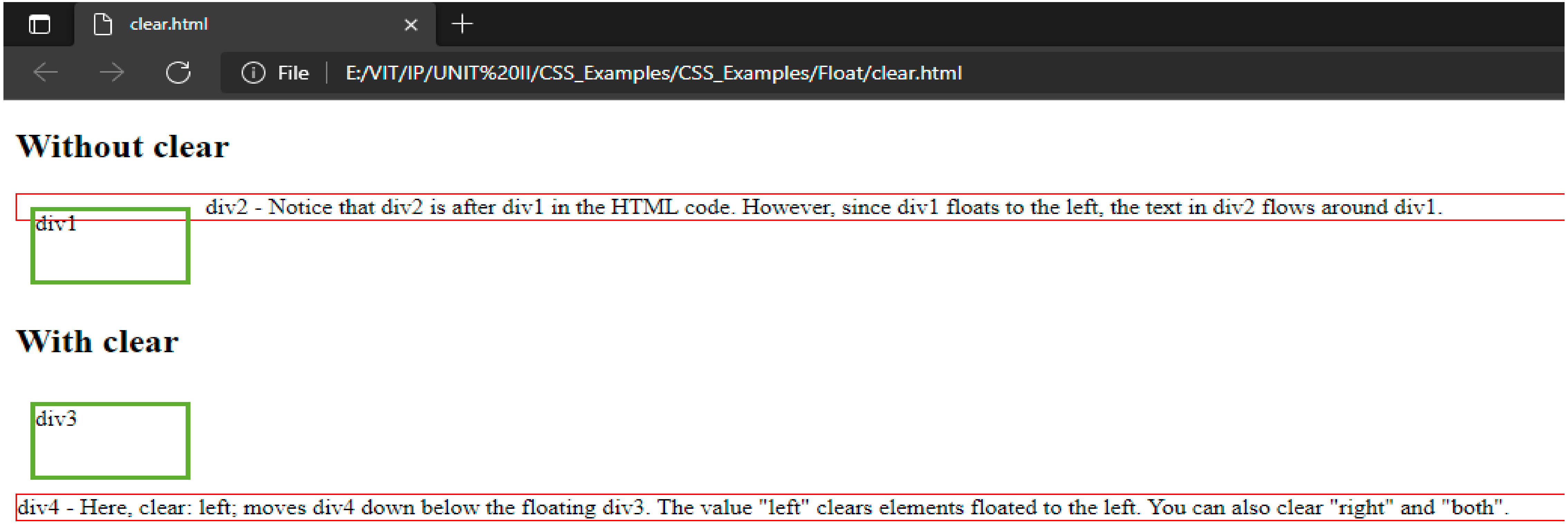
CSS Layout : clear property

- The clear property specifies **what elements can float beside the cleared element and on which side.**
- The most common way to use the clear property is **after** you have used a **float property** on an element.
- When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left.
- Your floated element will continue to float, but the cleared element will appear below it on the web page.

CSS Layout : clear property

```
<head>
<style>
.div1 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #73AD21;
}
.div2 {
  border: 1px solid red;
}
.div3 {
  float: left;
  width: 100px;
  height: 50px;
  margin: 10px;
  border: 3px solid #73AD21;
}
.div4 {
  border: 1px solid red;
  clear: left;
}
</style>
</head>
<body>
<h2>Without clear</h2>
<div class="div1">div1</div>
<div class="div2">div2 - Notice that div2 is after div1 in the HTML code.
However, since div1 floats to the left, the text in div2 flows around
div1.</div>
<br><br><br>
<h2>With clear</h2>
<div class="div3">div3</div>
<div class="div4">div4 - Here, clear: left; moves div4 down below the
floating div3. The value "left" clears elements floated to the left. You can also
clear "right" and "both".</div>
```

CSS Layout : clear property



clear.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Float/clear.html

Without clear

div1

div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

With clear

div3

div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

CSS Layout : overflow property

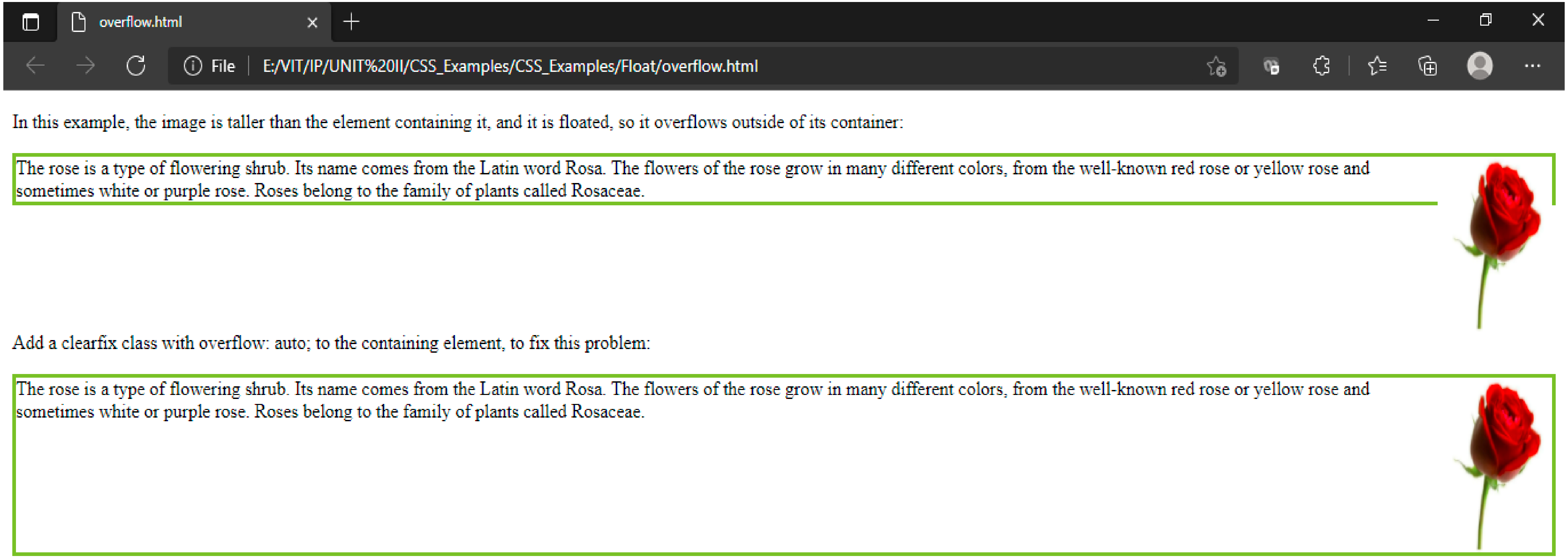
- The overflow property specifies whether to **clip the content** or to **add scrollbars** when the content of an element is too big **to fit in the specified area**.
- The overflow property has the **following values**:
- **visible** - Default. The overflow is not clipped. The content renders outside the element's box
- **hidden** - The overflow is clipped, and the rest of the content will be invisible
- **scroll** - The overflow is clipped, and a scrollbar is added to see the rest of the content
- **auto** - Similar to scroll, but it adds scrollbars only when necessary

CSS Layout : overflow property

```
<head>
<style>
div {
    border: 3px solid #8AC007;
}

.img1 {
    float: right;
}

.clearfix {
    overflow: auto;
}

.img2 {
    float: right;
}
</style>
</head>
```

```
<body>
<p>In this example, the image is taller than the element containing it, and it is floated,
so it overflows outside of its container:</p>

<div><img class="img1" src="rose-flower.jpg" alt="Rose" width="100" height="140">
The rose is a type of flowering shrub. Its name comes from the Latin word Rosa. The
flowers of the rose grow in many different colors, from the well-known red rose or
yellow rose and sometimes white or purple rose. Roses belong to the family of plants
called Rosaceae.</div>
<p style="clear:right">Add a clearfix class with overflow: auto; to the containing
element, to fix this problem:</p>

<div class="clearfix"><img class="img2" src="rose-flower.jpg" alt=“Rose" width="100"
height="140">
The rose is a type of flowering shrub. Its name comes from the Latin word Rosa. The
flowers of the rose grow in many different colors, from the well-known red rose or
yellow rose and sometimes white or purple rose. Roses belong to the family of plants
called Rosaceae.</div>

</body>
```

CSS Layout : overflow property


overflow.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Float/overflow.html

In this example, the image is taller than the element containing it, and it is floated, so it overflows outside of its container:

The rose is a type of flowering shrub. Its name comes from the Latin word Rosa. The flowers of the rose grow in many different colors, from the well-known red rose or yellow rose and sometimes white or purple rose. Roses belong to the family of plants called Rosaceae.

Add a clearfix class with overflow: auto; to the containing element, to fix this problem:

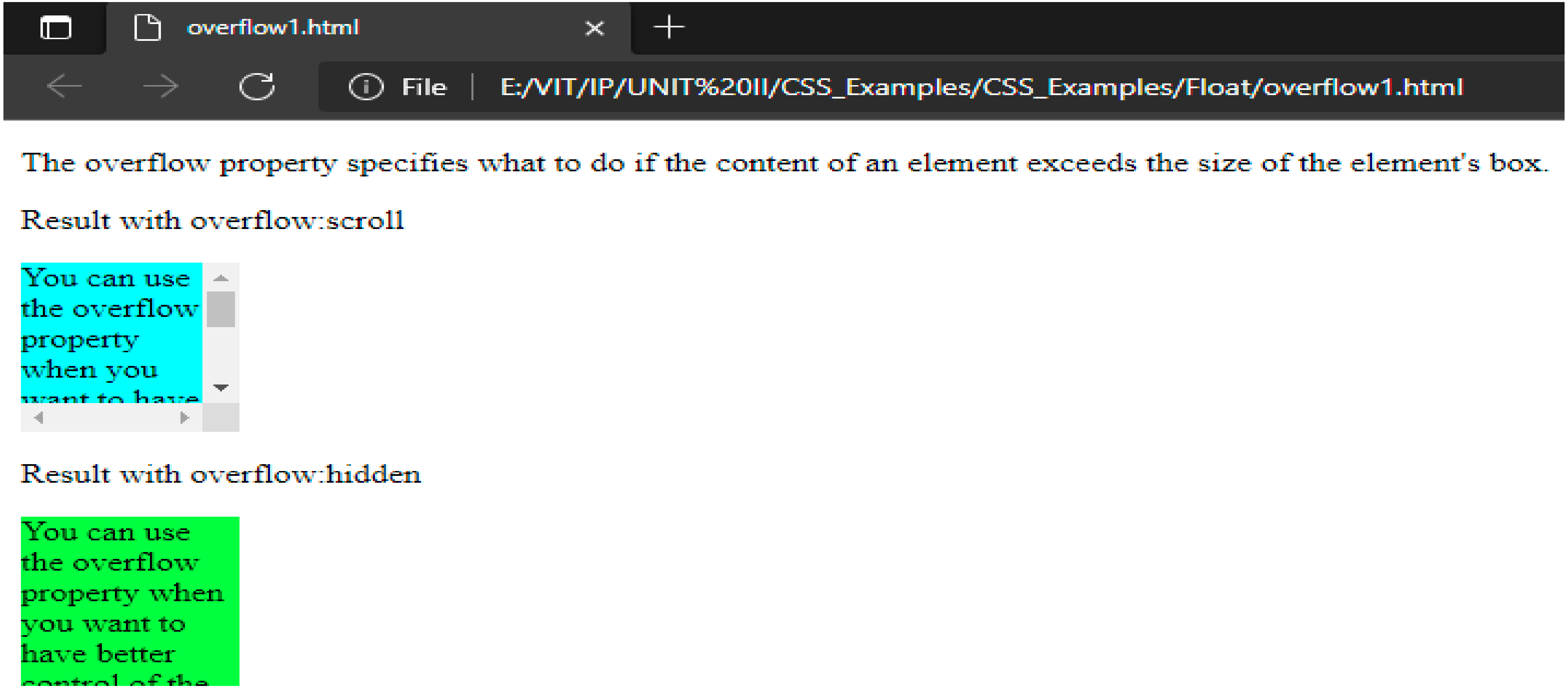
The rose is a type of flowering shrub. Its name comes from the Latin word Rosa. The flowers of the rose grow in many different colors, from the well-known red rose or yellow rose and sometimes white or purple rose. Roses belong to the family of plants called Rosaceae.

CSS Layout : overflow property

```
<head>
<style>
div.scroll {
    background-color: #00FFFF;
    width: 100px;
    height: 100px;
    overflow: scroll;
}

div.hidden {
    background-color: #00FF00;
    width: 100px;
    height: 100px;
    overflow: hidden;
}
</style>
</head>
```

```
<body>

<p>The overflow property specifies what to do if the content of an element exceeds
the size of the element's box.</p>

<p>Result with overflow:scroll</p>
<div class="scroll">You can use the overflow property when you want to have better
control of the layout. The default value is visible.</div>

<p>Result with overflow:hidden</p>
<div class="hidden">You can use the overflow property when you want to have
better control of the layout. The default value is visible.</div>

</body>
```

CSS Layout : overflow property



overflow1.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Float/overflow1.html

The overflow property specifies what to do if the content of an element exceeds the size of the element's box.

Result with overflow:scroll

You can use the overflow property when you want to have

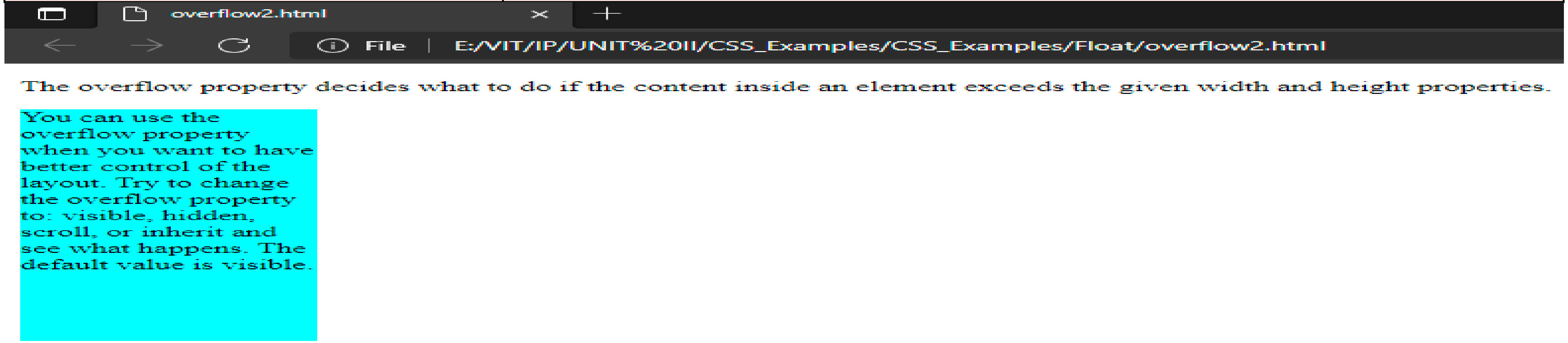
Result with overflow:hidden

You can use the overflow property when you want to have better control of the

CSS Layout : overflow property

```
<head>
<style>
div {
    background-color: #00FFFF;
    width: 150px;
    height: 350px;
    overflow: auto;
}
</style>
</head>
```

```
<body>

<p>The overflow property decides what to do if the content inside an element
exceeds the given width and height properties.</p>
<div>You can use the overflow property when you want to have better control of the
layout. Try to change the overflow property to: visible, hidden, scroll, or inherit and
see what happens. The default value is visible.</div>

</body>
```

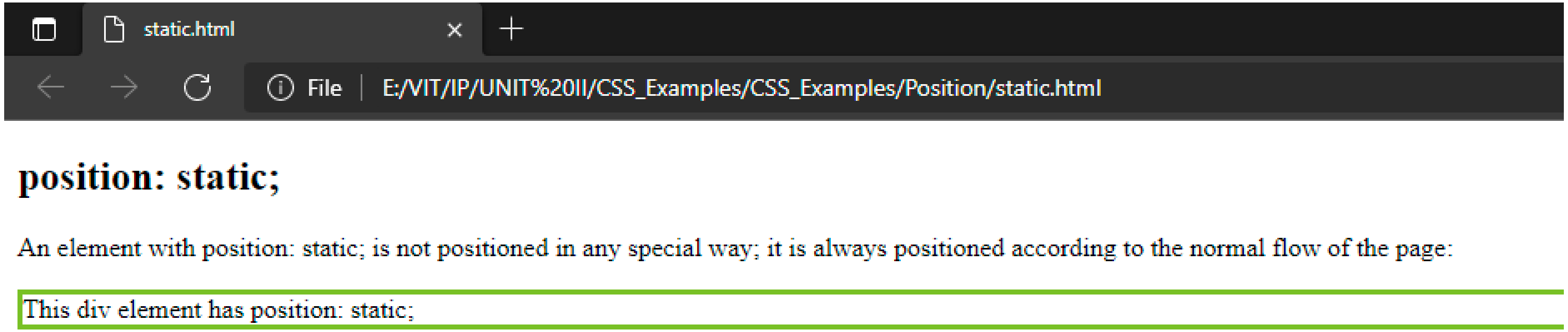

overflow2.html
File | E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Float/overflow2.html

The overflow property decides what to do if the content inside an element exceeds the given width and height properties.

```
You can use the
overflow property
when you want to have
better control of the
layout. Try to change
the overflow property
to: visible, hidden,
scroll, or inherit and
see what happens. The
default value is visible.
```

CSS Layout : Position

- The position property specifies the **type of positioning method** used for an element (static, relative, fixed or absolute).
- HTML elements are positioned **static by default**.
  - Static positioned elements are not affected by the top, bottom, left, and right properties.


static.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Position/static.html

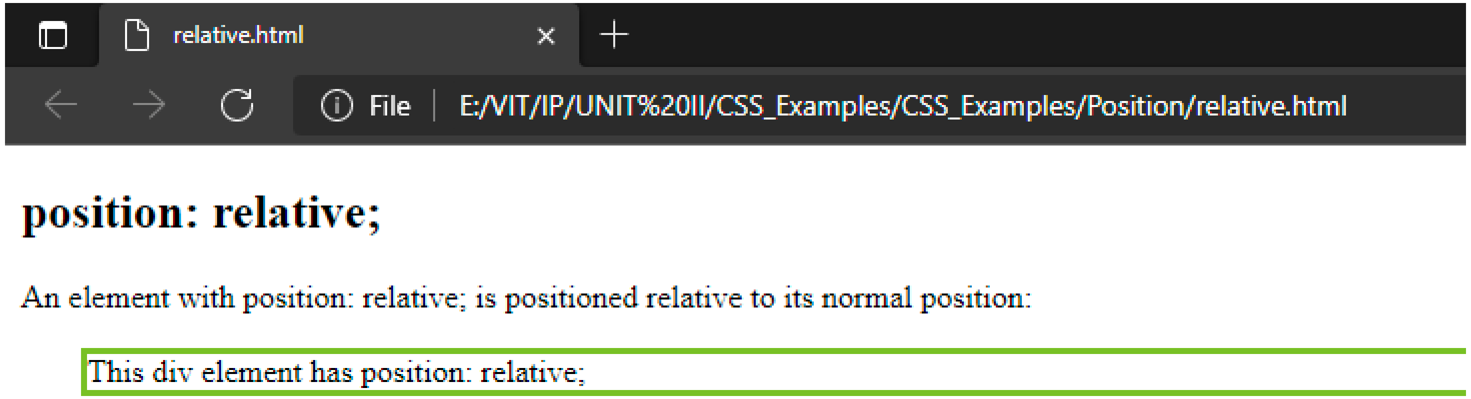
position: static;

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;

CSS Layout : Position

- 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.


relative.html
File
E:/VIT/IP/UNIT%20II/CSS_Examples/CSS_Examples/Position/relative.html

position: relative;

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

This div element has position: relative;

CSS Layout : Position

- 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**.
- 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.

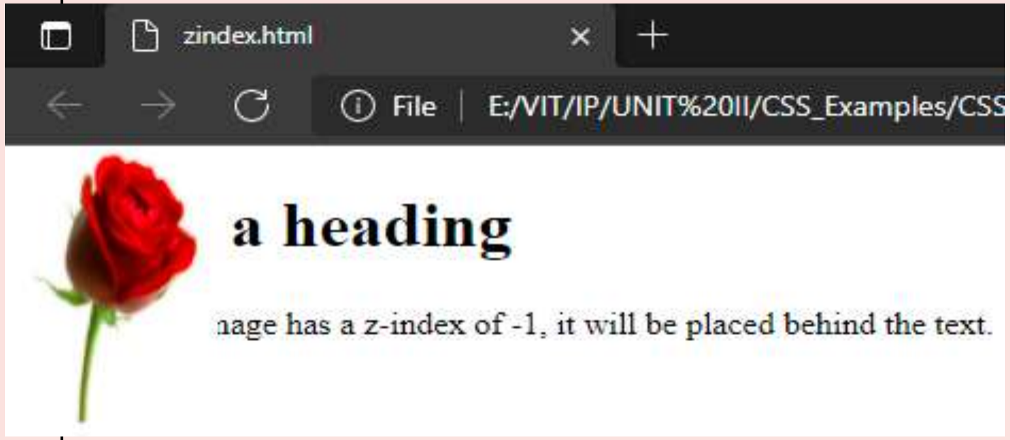
CSS Layout : Position

- When elements are positioned, they can **overlap other elements.**
- The **z-index property** specifies the stack order of an element (which element should be placed in front of, or behind, the others).
- An element can have a positive or negative stack order.
- An element with **greater stack order is always in front** of an element with a lower stack order.

CSS Layout : overflow property

```
<head>
<style>
img {
    position: absolute;
    left: 0px;
    top: 0px;
    z-index:-1;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<img src="rose-flower.jpg" width="100" height="140">
<p>Because the image has a z-index of -1, it will be placed
behind the text.</p>

</body>
```

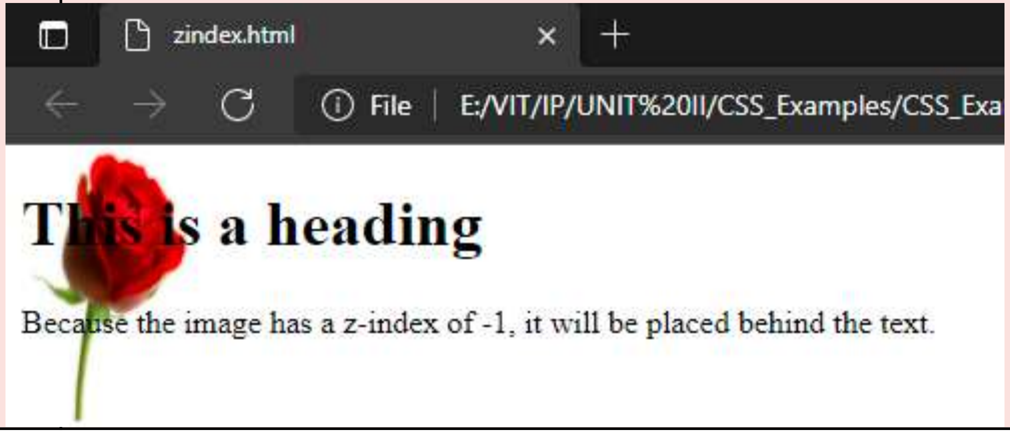
Without z-index property



a heading

nage has a z-index of -1, it will be placed behind the text.

With z-index property

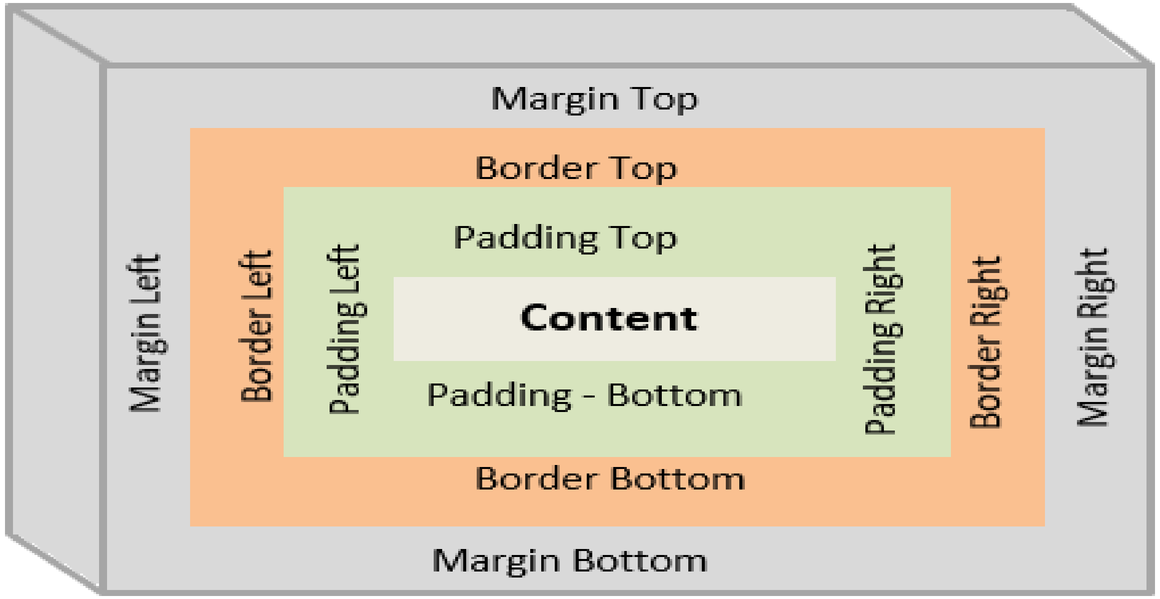


This is a heading

Because the image has a z-index of -1, it will be placed behind the text.

CSS Layout : Box Model

- All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.



CSS Layout : Box Model

- The CSS box-sizing property allows us to include the padding and border in an element's total width and height.
- By default, the **width and height of an element** is calculated like this:

  width + padding + border = actual width of an element

  height + padding + border = actual height of an element
- **content-box**: The width and height properties (and min/max properties) includes only the content. Border and padding are not included
- **border-box**: The width and height properties (and min/max properties) includes content, padding and border

CSS3 Features

CSS3 Evolution

- **Cascading Style Sheets, level 1** (CSS1) was came out of **W3C** as a recommendation in **December 1996**.
- This version describes the CSS language as well as a **simple visual formatting model** for all the HTML tags.
- **CSS2** became a W3C recommendation **in May 1998** and **builds on CSS1**.
- This version adds support for **media-specific style sheets.**

e.g. printers and aural devices, downloadable fonts, element positioning, and tables.

CSS3 Evolution

- **CSS3** became a W3C recommendation **in June 1999** and builds on older versions CSS.
- It has divided into documentation is called as **Modules** and here each module having **new extension features defined in CSS2.**
- **CSS3** is the latest standard of CSS earlier versions (CSS2).
- The main difference between CSS2 and CSS3 is as follows:
  - Media Queries
  - Namespaces
  - Selectors Level 3
  - Color

What’s new with CSS3?

- There’s a **huge amount** of improvements over the last version, like the **ability to easily add video and 3D objects to your webpage.**
- In fact there’s much more that can be done on this platform.

Backwards compatible

- CSS3 is completely **backwards compatible with earlier CSS versions.**
- That old site running on a previous CSS version can be reworked with CSS3.

What’s new with CSS3?

Independent modules

- CSS3 is broken up into many **individual modules**, improving both functionality and ease of working that it makes CSS3 a lot easier to handle.
- The most important **modules** include Selectors, Color, Box Model, Backgrounds and Borders, Text Effects, 2D/3D Transformations and user interface.
- Though many of these are in old CSS specifications, they have been split into small functional pieces called modules in CSS3.

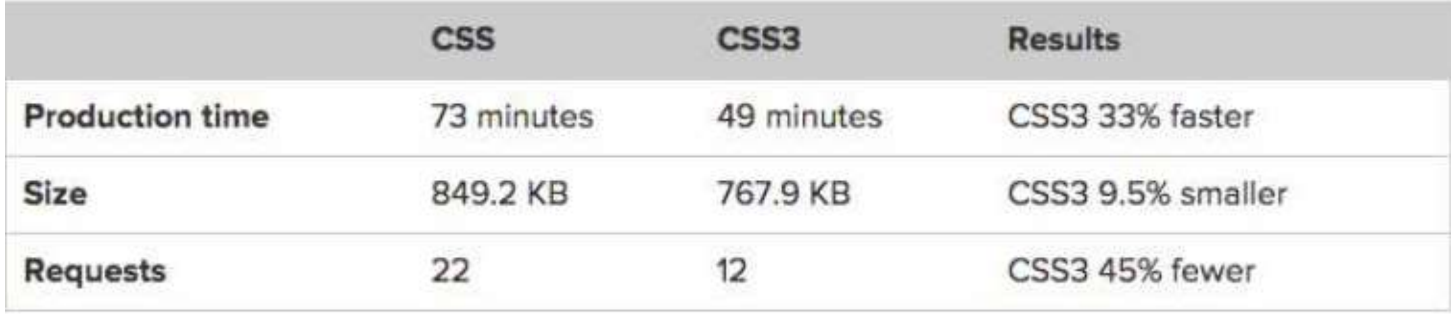
Easy to change. Easy to view

- With the modular structure and new functionalities, CSS3 makes it much easier to make changes.
- This allows individual module testing!

What’s new with CSS3?

Faster development speed

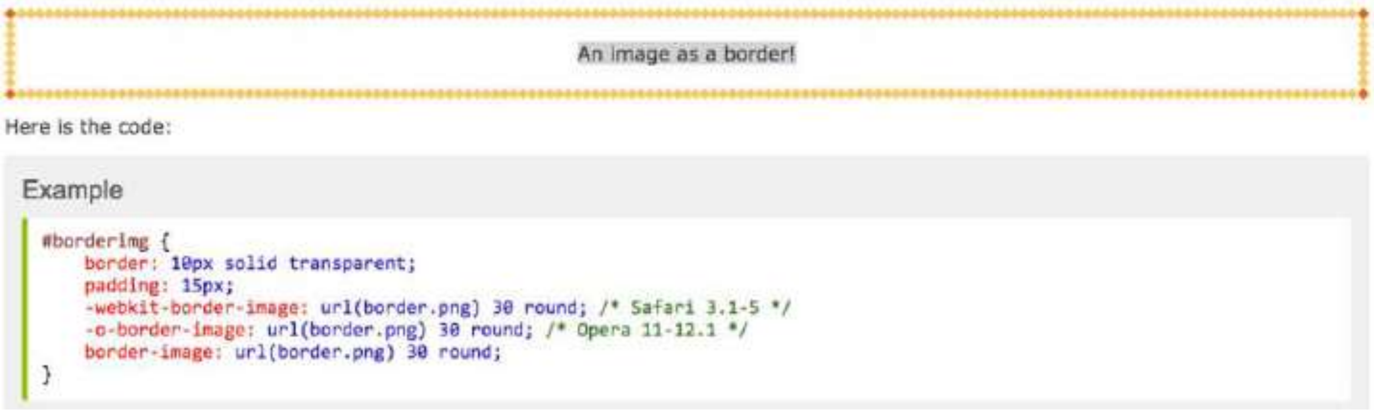
- CSS3 has also managed to set new benchmarks in development speed.
- The speed of website built with CSS to the same one with CSS3 was compared by ‘The Smashing Magazine’.
- According to the results, CSS3 really pays off for production time and size.



	CSS	CSS3	Results
Production time	73 minutes	49 minutes	CSS3 33% faster
Size	849.2 KB	767.9 KB	CSS3 9.5% smaller
Requests	22	12	CSS3 45% fewer

What’s new with CSS3?

- **Multiple background:** Multiple background images can be included on a page and layered.
- **CSS3 background size:** The CSS3 background-size property allows to specify the size of these background images
- **Borders and text effects:** CSS3 has also managed to set new benchmarks in development speed



An image as a border!

Here is the code:

Example

```
#borderImg {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 30 round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 30 round; /* Opera 11-12.1 */
    border-image: url(border.png) 30 round;
}
```

CSS3

- CSS3 is still under development
- Modules: A new approach to standardization
- Each browser still implementing slightly different syntax for certain properties:

  moz – <property> ; //Firefox

  webkit – <property>; //Chrome, Safari (webkit)

  Opera (in most cases) uses the default with no prefix but in some cases you will see

  – o – <property>;

CSS3 Features

CSS3

- CSS3 Rounded corners are used to add special colored corner to body or text by using the border-radius property.
- Makes creating rounded divs a breeze

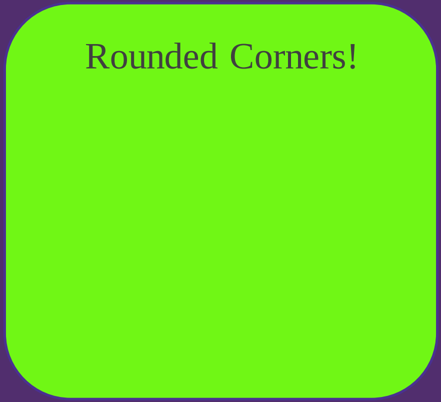
```
.div2
{
   width:300px;
   height:300px;
   border: 3px solid #000;
   background:#cc0000;
   border-radius:30px;
   //Prefix to make this work in Firefox
   -moz-border-radius:30px;
   //Prefix to make this work in webkit
   -webkit-border-radius:30px;
    browsers
}
```

CSS3 Rounded Corners

- border-radius: Use this element for setting four boarder radius property
- border-top-left-radius: Use this element for setting the boarder of top left corner
- border-top-right-radius: Use this element for setting the boarder of top right corner
- border-bottom-right-radius: Use this element for setting the boarder of bottom right corner
- border-bottom-left-radius: Use this element for setting the boarder of bottom left corner

CSS3 Rounded Corners

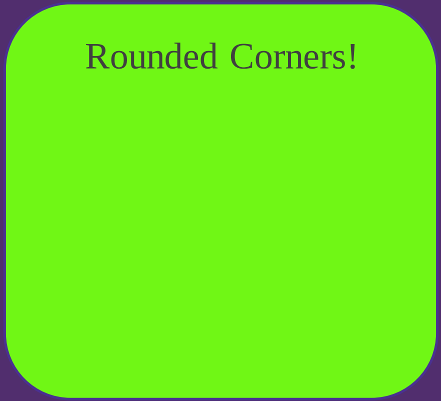
```
<html>
  <head>
    <style>
      #rcorners1 {
        border-radius: 25px;
        background: #8AC007;
        padding: 20px;
        width: 200px;
        height: 150px;
      }
```



Rounded Corners!

CSS3 Rounded Corners

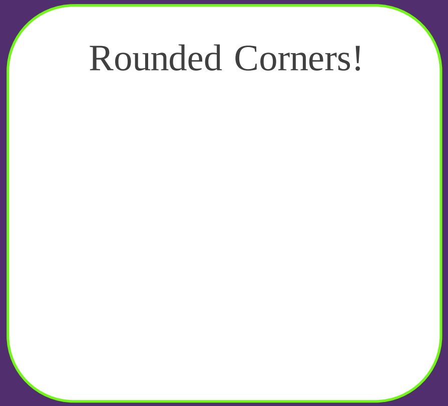
```
</style>
   </head>
<body>
 <p id = "rcorners1">Rounded corners!</p>
</body>
</html>
```



Rounded Corners!

CSS3 Rounded Corners

```
<html>
  <head>
    <style>
       #rcorners2 {
         border-radius: 25px;
         border: 2px solid #8AC007;
         padding: 20px;
         width: 200px;
         height: 150px;
      }
```

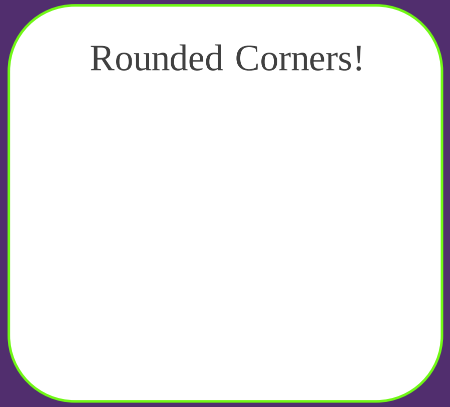


Rounded Corners!

CSS3 Features

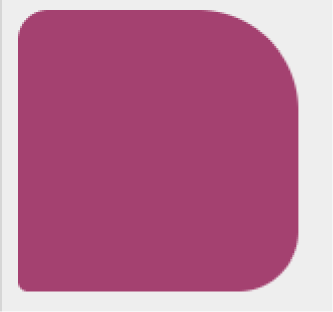
CSS3 Rounded Corners

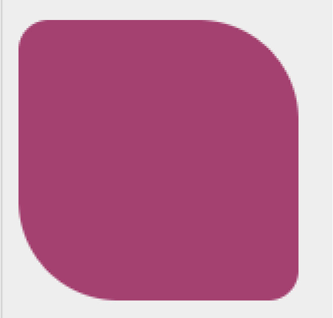
```
</style>
   </head>
<body>
 <p id = "rcorners2">Rounded corners!</p>
</body>
</html>
```



Rounded Corners!

CSS3 Rounded Corners





```
<html>
   <head>
      <style>
        #rcorners1 {
            border-radius: 15px 50px 30px 5px;
            background: #a44170;
            padding: 20px;
            width: 100px;
            height: 100px;
         }
        #rcorners3 {
            border-radius: 15px 50px;
            background: #a44170;
            padding: 20px;
            width: 100px;
            height: 100px;
         }
       </style>
   </head>
<body>
 <p id = "rcorners1">Rounded corners!</p>
 <p id = "rcorners3">Rounded corners!</p>
</body>
</html>
```

CSS3 Border Images

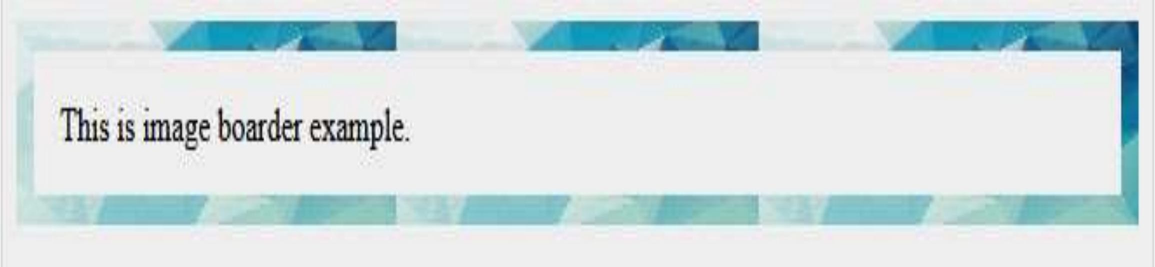
- CSS Border image property is used to add image boarder to some elements.
- you don't need to use any HTML code to call boarder image.
  - border-image-source: Used to set the image path
  - border-image-slice: Used to slice the boarder image
  - border-image-width: Used to set the boarder image width
  - border-image-repeat: Used to set the boarder image as rounded, repeated and stretched

CSS3 Border Images

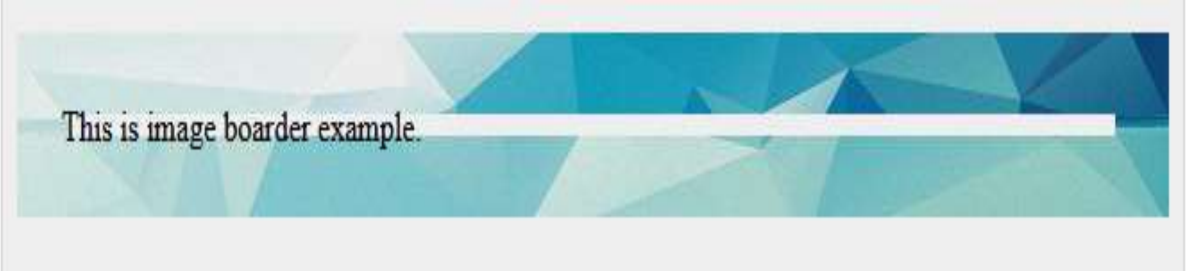
- Format: border-image: source slice width outset repeat;
- Markup:

```
border-image: url(borderbg.jpg) 30 30 repeat;

-webkit-border-image: url(borderbg.jpg) 30 30 repeat;

-moz-border-image: url(borderbg.jpg) 30 30  repeat;
```

CSS3 Border Images



This is image boarder example.



This is image boarder example.

```
<html> <head> <style>
#borderimg1 {
border: 10px solid transparent; padding: 15px;
border-image-source: url(/css/images/border.png);
border-image-repeat: round;
border-image-slice: 30;
border-image-width: 10px;
}
#borderimg3 {
border: 10px solid transparent; padding: 15px;
]border-image-source:url(/css/images/border.png);
border-image-repeat: round;
border-image-slice: 30;
border-image-width: 30px;
}
</style> </head>
<body>
 <p id = "borderimg1">This is image boarder example.</p>
 <p id = "borderimg3">This is image boarder example.</p>
</body> </html>
```

CSS3 Shadows

- CSS3 supported to add shadow to text or elements.
- Shadow property has divided as follows:
  - Text shadow
  - Box Shadow

CSS3 Shadows

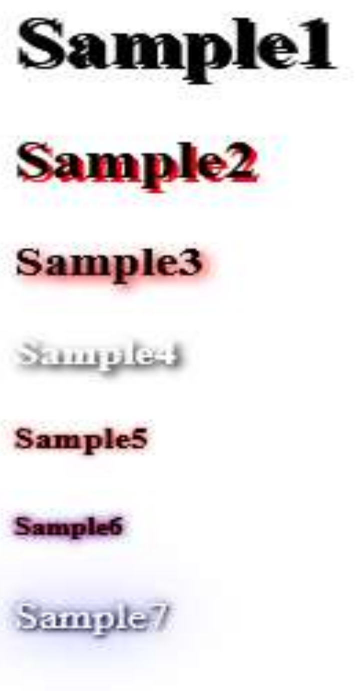
- Once again no IE support
- Format: box-shadow: h-shadow v-shadow blur spread color inset;

  h-shadow == Position of horizontal shadow

  V-shadow == Position of vertical shadow
- Blur == Blur distance
- Spread == Size of shadow
- Color == Shadow Color
- Inset == Make the shadow an inner (shadow)

CSS3 Features

CSS3 Shadows



Sample1

Sample2

Sample3

Sample4

Sample5

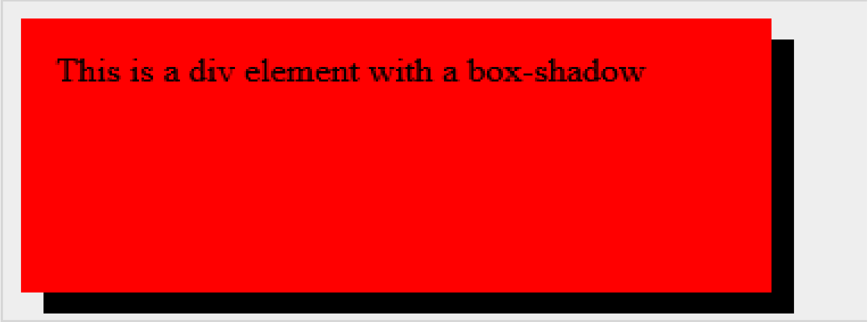
Sample6

Sample7

```
<html>  <head>   <style>
h1 {text-shadow: 2px 2px;          }
h2 {text-shadow: 2px 2px red;         }
h3 {text-shadow: 2px 2px 5px red;         }
h4 {color: white;  text-shadow: 2px 2px 4px #000000; }
h5 {text-shadow: 0 0 3px #FF0000;        }
h6 {text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF; }
p {
color: white;
text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px
darkblue;
}
</style> </head>
<body>
   <h1>Sample1</h1>
   <h2>Sample2</h2>
   <h3>Sample3</h3>
   <h4>Sample4</h4>
   <h5>Sample5</h5>
   <h6>Sample6</h6>
   <p>Sample7</p>
</body> </html>
```

CSS3 Box shadow

- Used to add shadow effects to elements, Following is the example to add shadow effects to element.



This is a div element with a box-shadow

```
<html>
  <head>
    <style>
      div {
        width: 300px;
        height: 100px;
        padding: 15px;
        background-color: red;
        box-shadow: 10px 10px;
      }
    </style>
  </head>
  <body>
   <div>
    This is a div element with a box-shadow
   </div>
  </body>
</html>
```

CSS3 Backgrounds

- CSS Multi background property is used to add one or more images at a time without HTML code, We can add images as per our requirement.
- Format: background-size: length | percentage | cover | contain;
- Sample Markup:

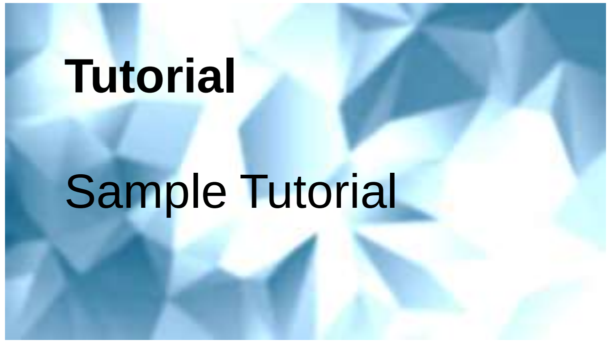
```
background-size: 400px 500px; //width, height

background-size:30% 40%; //width, height

background-size:100%; //Omitting first value sets

                                              //the second to auto

background-size:cover;

background-size:contain;
```

CSS3 Multi Background

- CSS Multi background property is used to add one or more images at a time without HTML code, We can add images as per our requirement.
  - Background: Used to setting all the background image properties in one section
  - background-clip: Used to declare the painting area of the background
  - background-image: Used to specify the background image
  - background-origin: Used to specify position of the background images
  - background-size: Used to specify size of the background images

CSS3 Features

CSS3 Multi Background



Tutorial

Sample Tutorial

```
<html>
<head>
<style>
#multibackground
{
background-image: url(/css/images/logo.png),
url(/css/images/border.png);
background-position: left top, left top;
background-repeat: no-repeat, repeat;            padding:
75px;
}
</style>
</head>
<body>
<div id = "multibackground">
<h1>Tutorial</h1>
<p>    Sample Tutorial   </p>
</div>
</body>
</html>
```

CSS3 Multi Background

- Size of Multi background
- Multi background property is accepted to add different sizes for different images.

```
#multibackground {
   background: url(/css/imalges/logo.png) left top no-repeat,
   url(/css/images/boarder.png) right bottom no-repeat,
   url(/css/images/css.gif) left top repeat;
   background-size: 50px, 130px, auto;
}
```

- Each image is having specific sizes as 50px, 130px and auto size.

CSS3-Colors

- CSS3 has Supported additional color properties as follows −
  - RGBA colors
  - HSL colors
  - HSLA colors
  - Opacity

CSS3-Colors

- RGBA stands for Red Green Blue Alpha.
- It is an extension of CSS2,Alpha specifies the opacity of a color and parameter number is a numerical between 0.0 to 1.0.

```
#d1 {background-color: rgba(255, 0, 0, 0.5);}

#d2 {background-color: rgba(0, 255, 0, 0.5);}

#d3 {background-color: rgba(0, 0, 255, 0.5);}
```

CSS3-Colors

- HSL stands for hue, saturation, lightness.
- Here Hue is a degree on the color wheel, saturation and lightness are percentage values between 0 to 100%.

```
#g1 {background-color: hsl(120, 100%, 50%);}

#g2 {background-color: hsl(120, 100%, 75%);}

#g3 {background-color: hsl(120, 100%, 25%);}
```

CSS3-Colors

- HSLA stands for hue, saturation, lightness and alpha.
- Alpha value specifies the opacity of the color as a number between 0.0 to 1.0.

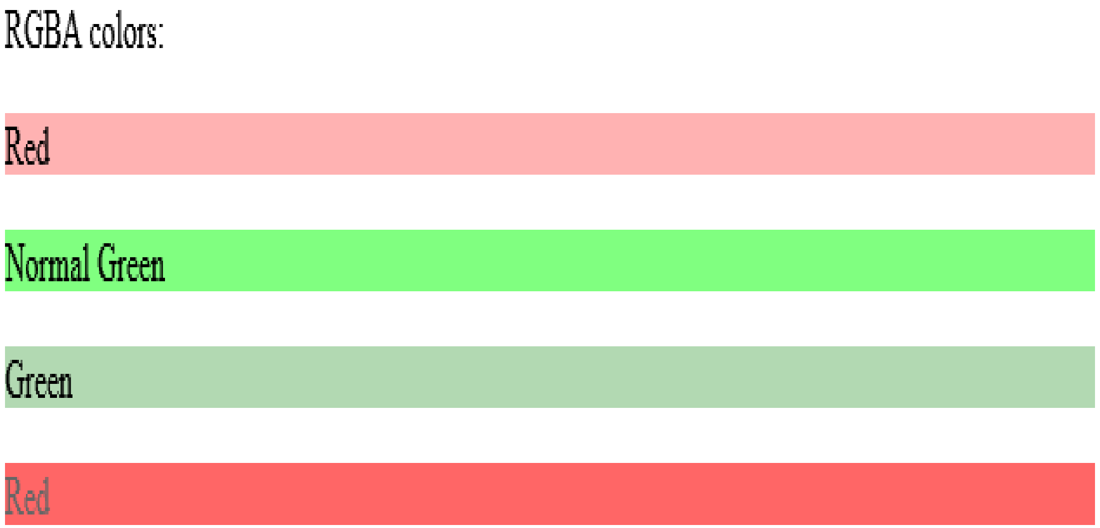
```
#g1 {background-color: hsla(120, 100%, 50%, 0.3);}

#g2 {background-color: hsla(120, 100%, 75%, 0.3);}

#g3 {background-color: hsla(120, 100%, 25%, 0.3);}
```

CSS3-Colors

- Opacity adds transparency to the background of an element.

```
#g1 {background-color:rgb(255,0,0);opacity:0.6;}

#g2 {background-color:rgb(0,255,0);opacity:0.6;}

#g3 {background-color:rgb(0,0,255);opacity:0.6;}
```

CSS3-Colors



RGBA colors:

Red

Normal Green

Green

Red

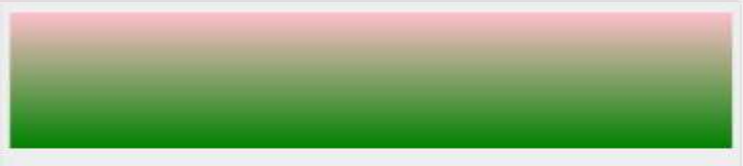
```
<html>
<head>
<style>
#p1 {background-color:rgba(255,0,0,0.3);}
#g2 {background-color:hsl(120,100%,75%);}
#d3 {background-color:hsla(120,100%,25%,0.3);}
#m1 {background-color:rgb(255,0,0);opacity:0.6;}
</style>
</head>
<body>
<p>RGBA colors:</p>
<p id = "p1">Red</p>
<p id = "g2">Normal Green</p>
<p id = "d3">Green</p>
<p id = "m1">Red</p>
</body></html>
```

CSS3 - Gradients

- Gradients displays the combination of two or more colors
- Types of gradients
  - Linear Gradients(down/up/left/right/diagonally)
  - Radial Gradients

CSS3 – Linear Gradients

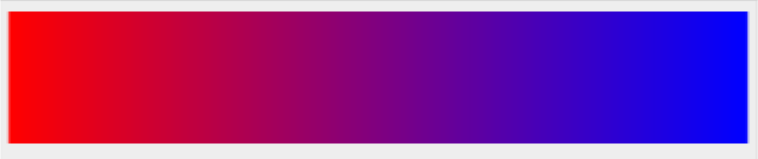
- Linear gradients are used to arrange two or more colors in linear formats like top to bottom.



```
<html>
<head>
<style>
#grad1 {
height: 100px;
background: -webkit-linear-gradient(pink,green);
background: -o-linear-gradient(pink,green);
background: -moz-linear-gradient(pink,green);
background: linear-gradient(pink,green);
}
</style>
</head>
   <body>
   <div id = "grad1">
   </div>
   </body>
</html>
```

CSS3 – Linear Gradients

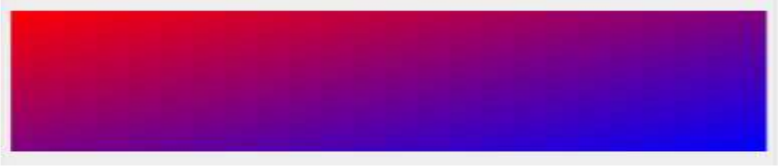
- Linear gradients are used to arrange two or more colors in linear formats like left to right.



```
<html>
<head>
<style>
#grad1 {
height: 100px;
background: -webkit-linear-gradient(left, red , blue);
background: -o-linear-gradient(right, red, blue);
background: -moz-linear-gradient(right, red, blue);
background: linear-gradient(to right, red , blue);}
</style>
</head>
  <body>
  <div id = "grad1">
  </div>
  </body>
</html>
```

CSS3 – Linear Gradients

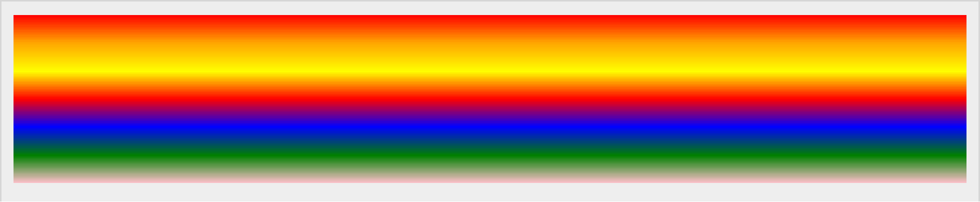
- Linear gradients are used to arrange two or more colors in linear formats like diagonal.



```
<html>
<head>
<style>
#grad1 {
 height: 100px;
background: -webkit-linear-gradient(left top, red , blue);
background: -o-linear-gradient(bottom right, red, blue);
background: -moz-linear-gradient(bottom right, red, blue);
background: linear-gradient(to bottom right, red , blue);
}
</style>
</head>
   <body>
   <div id = "grad1">
   </div>
   </body>
</html>
```

CSS3 – Linear Gradients

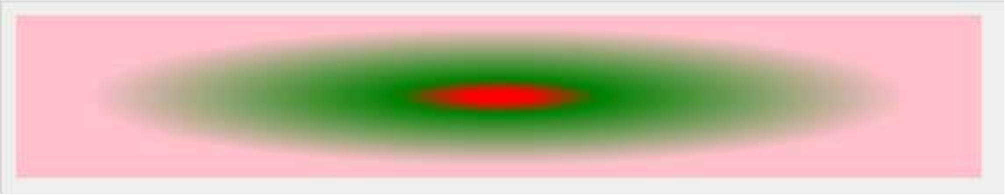
- Linear gradients are used to arrange two or more colors in linear formats like Multi color.



```
<html>
<head>
<style>
#grad2 {
 height: 100px;
background: -webkit-linear-gradient(red, orange, yellow, red, blue, green,pink);
background: -o-linear-gradient(red, orange, yellow, red, blue, green,pink);
background: -moz-linear-gradient(red, orange, yellow, red, blue, green,pink);
background: linear-gradient(red, orange, yellow, red, blue, green,pink);
}
</style>
</head>
   <body>
   <div id = "grad2">
   </div>
   </body>
</html>
```

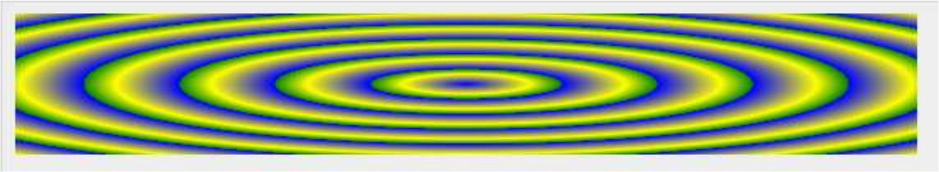
CSS3 - Radial Gradients

- Radial gradients appears at center.



```
<html>
<head>
<style>
#grad2 {
height: 100px; width: 550px;
background: -webkit-radial-gradient(red 5%, green 15%, pink 60%);
background: -o-radial-gradient(red 5%, green 15%, pink 60%);
background: -moz-radial-gradient(red 5%, green 15%, pink 60%);
background: radial-gradient(red 5%, green 15%, pink 60%);
}
</style>
</head>
   <body>
   <div id = "grad2">
   </div>
   </body>
</html>
```

CSS3 – Repeat Radial Gradients



```
<html>
<head>
<style>
#grad2 {
height: 100px; width: 550px;
background: -webkit-repeating-radial-gradient(blue, yellow 10%, green 15%);
background: -o-repeating-radial-gradient(blue, yellow 10%, green 15%);
background: -moz-repeating-radial-gradient(blue, yellow 10%, green 15%);
background: repeating-radial-gradient(blue, yellow 10%, green 15%);
}
</style>
</head>
  <body>
  <div id = "grad2">
  </div>
  </body>
</html>
```

CSS Text Properties

- Text Overflow
- Word Break
- Text Shadow
- Various values
  - text-align-last: Used to align the last line of the text
  - text-emphasis: Used to emphasis text and color
  - text-overflow: used to determines how overflowed content that is not displayed is signaled to users
  - word-break: Used to break the line based on word
  - word-wrap: Used to break the line and wrap onto next line

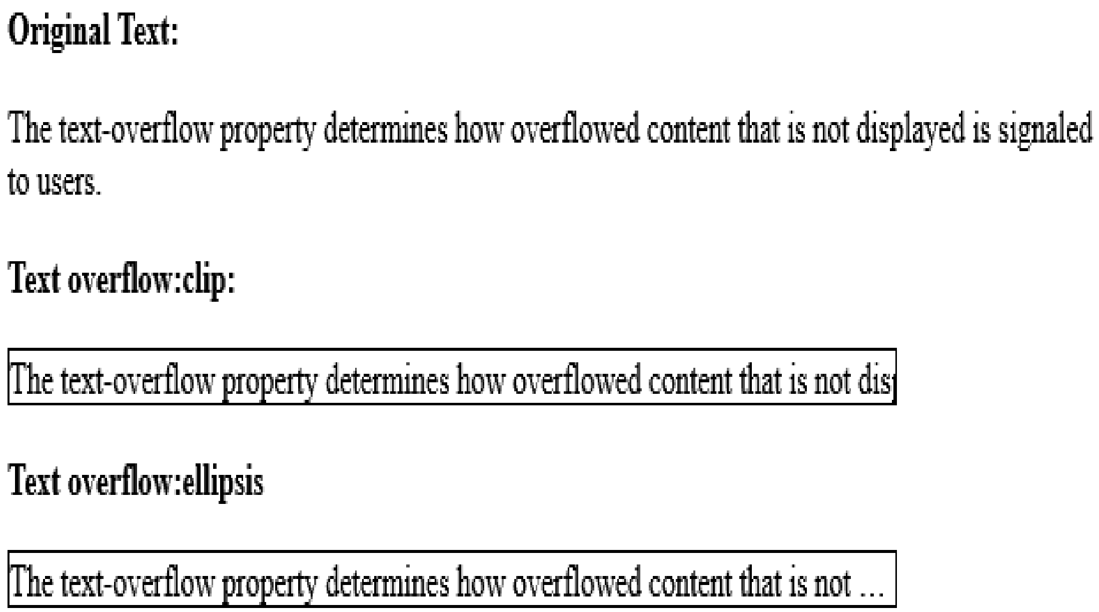
CSS Text Properties – Text Overflow

- Supported in Internet Explorer (+ other browsers)
- The text-overflow property determines how overflowed content that is not displayed is signalled to users.
- Format: text-overflow: clip | ellipsis | string;
- Also requires the whitespace:nowrap; & overflow:hidden properties to be set.

```
 #div1
{
    width:200px;
    overflow:hidden;
    white-space:nowrap;
    text-overflow:ellipsis;
}
```

CSS3 Features

CSS Text Properties – Text Overflow



Original Text:

The text-overflow property determines how overflowed content that is not displayed is signaled to users.

Text overflow:clip:

The text-overflow property determines how overflowed content that is not dis

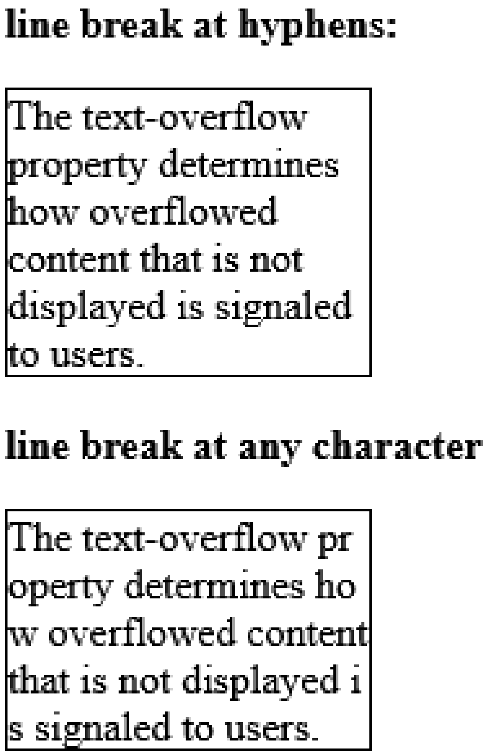
Text overflow:ellipsis

The text-overflow property determines how overflowed content that is not …

```
<html>   <head>     <style>
p.text1 {
white-space: nowrap; width: 500px;
border: 1px solid #000000;
overflow: hidden; text-overflow: clip; }
p.text2 {
white-space: nowrap;  width: 500px;
border: 1px solid #000000;
overflow: hidden; text-overflow: ellipsis; }
</style>   </head>   <body>

<b>Original Text:</b>
<p>  The text-overflow property determines how overflowed
content that is not displayed is signaled to users. </p>
<b>Text overflow:clip:</b>
<p class = "text1"> The text-overflow property determines
how overflowed content that is not displayed is signaled to
users. </p>
<b>Text overflow:ellipsis</b>
<p class = "text2"> The text-overflow property determines
how overflowed content that is not displayed is signaled to
users. </p>
</body> </html>
```

CSS3 Features

CSS Text Properties – Word Breaking



line break at hyphens:

The text-overflow
property determines
how overflowed
content that is not
displayed is signaled
to users.

line break at any character

The text-overflow pr
operty determines ho
w overflowed content
that is not displayed i
s signaled to users.

```
<html>
<head>
<style>
p.text1 {
width: 140px;  border: 1px solid #000000;
word-break: keep-all;  }
p.text2 {
width: 140px;  border: 1px solid #000000;
word-break: break-all;  }
</style>
</head>
<body>
<b>line break at hyphens:</b>
<p class = "text1">
The text-overflow property determines how overflowed
content that is not displayed is signaled to users.      </p>
<b>line break at any character</b>
<p class = "text2">
The text-overflow property determines how overflowed
content that is not displayed is signaled to users.      </p>
</body>
</html>
```

CSS3 Text Properties – Word Wrap

- Word wrapping is used to break the line and wrap onto next line
- Force a line break even with a long word
- Format: word-wrap:break-word|normal;

```
p {
  word-wrap: break-word;
}
```

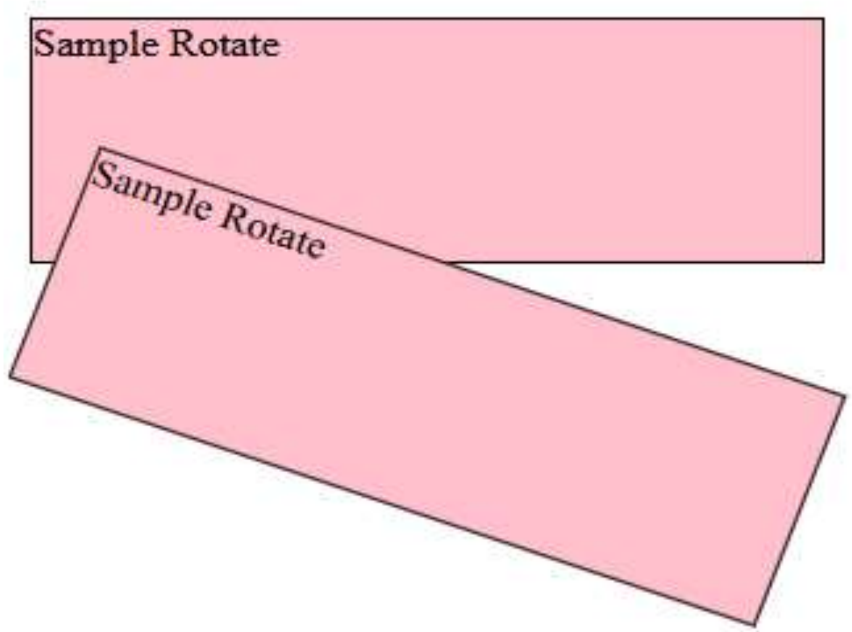
CSS3 Text Properties – Text Shadow

- Can be added outside the graphic
- No IE support
- Format: text-shadow: h-shadow v-shadow blur color;

CSS3 Transformations

- CSS3 supports 2D & 3D transformations
- 2D transforms are used to re-change the element structure as translate, rotate, scale, and skew.
- 2D transforms:
  - Translate
  - Rotate
  - Scale
  - Skew

CSS3 Transformations - 2D Rotate (20 Degrees)

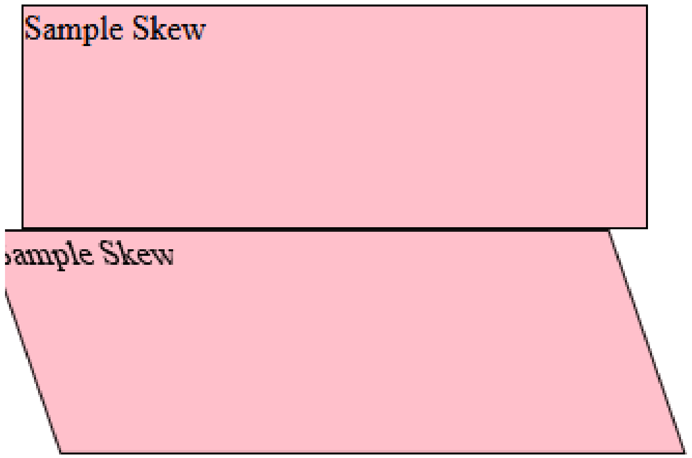


Sample Rotate

Sample Rotate

```
<html>
   <head>
      <style>
                  div {
                  width: 300px;
                  height: 100px;
                  background-color:
                  pink; border: 1px solid black;
                  }
                  div#myDiv {
           /* IE 9 */
            -ms-transform: rotate(20deg);
                   /* Standard syntax */
            transform: rotate(20deg);
         }
      </style>
   </head>
   <body>
      <div>    Sample Rotate       </div>
      <div id = "myDiv"> Sample Rotate   </div>
   </body>
</html>
```

CSS3 Transformations - 2D Skew(X axis)



Sample Skew

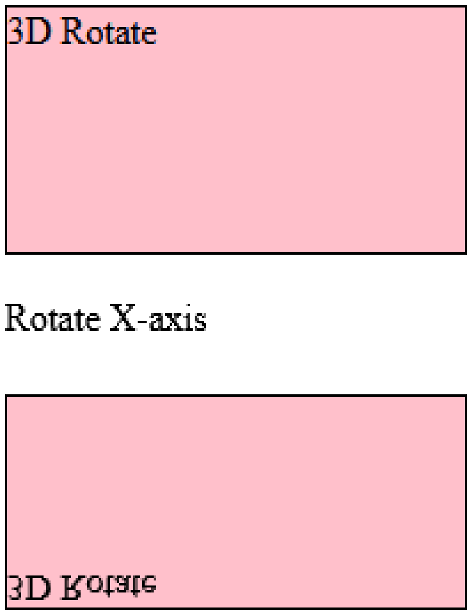
ample Skew

```
<html>
  <head>
    <style>
            div {
            width: 300px;
            height: 100px;
            background-color:
            pink; border: 1px solid black;
            }
            div#skewDiv {
        /* IE 9 */
        -ms-transform: skewX(20deg);
              /* Standard syntax */
        transform: skewX(20deg);
      }
    </style>
  </head>
  <body>
    <div>    Sample Skew      </div>
    <div id = “skewDiv"> Sample Skew   </div>
  </body>
</html>
```

CSS3 3D Transformations

- Using with 3D transforms, we can move element to x-axis, y-axis and z-axis, Below example clearly specifies how the element will rotate.
- 3D transforms:
  - Translate
  - Rotate
  - Scale
  - Skew

CSS3 Transformation- Rotate (X Axis)



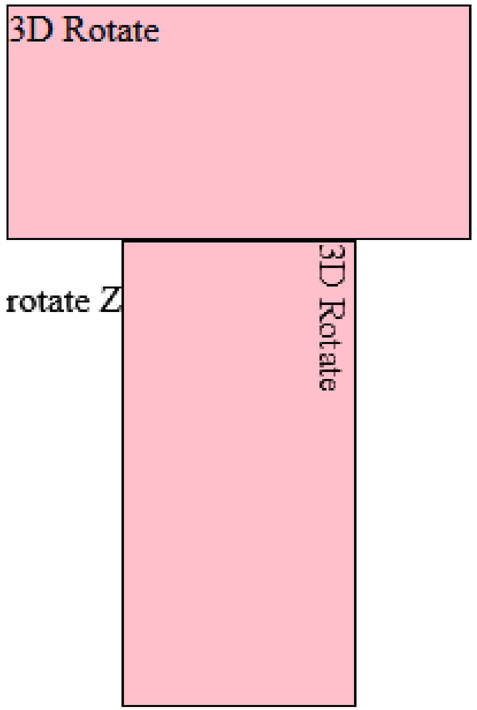
3D Rotate

Rotate X-axis

3D Rotate

```
<html>
  <head>
    <style>
      div {
        width: 200px;
        height: 100px;
        background-color: pink;
        border: 1px solid black; }
      div#myDiv {
        /* IE */
        -ms-transform: rotateX(150deg);
        /* Standard Syntax */
         transform: rotateX(150deg);
         }
    </style>
  </head>
  <body>
    <div> 3D Rotate </div>
            <p>Rotate X-axis</p>
    <div id = "myDiv"> 3D Rotate      </div>
  </body>
</html>
```

CSS3 Transformation- Rotate (Z Axis)



```
<html>
  <head>
    <style>
      div {
        width: 200px;
        height: 100px;
        background-color: pink;
        border: 1px solid black; }
      div#myDiv {
        /* IE */
        -ms-transform: rotateZ(90deg);
        /* Standard Syntax */
         transform: rotateZ(90deg);
         }
    </style>
  </head>
  <body>
    <div> 3D Rotate </div>
            <p>Rotate Z-axis</p>
    <div id = "myDiv"> 3D Rotate      </div>
  </body>
</html>
```

CSS3 Transitions

- Effects that let an element change from one style to another.
- Limited Support – Currently only supported by Webkit based browsers (Chrome, Safari)
- Cautions:
  - Must specify the property needed to add an effect.
  - Must specify a duration for the effect
  - Effect is typically applied on property change.

Format:

transition: <property> <duration>

-webkit-transition: <property> <duration>

*You can specify multiple properties separated by commas

CSS3 Animations

- Animation is process of making shape changes and creating motions with elements.
- Effects that let an element change from one style to another.
- Limited Support – Currently only supported by Webkit based browsers (Chrome, Safari)

CSS3 Animations - @keyframes

- Keyframes will control the intermediate animation steps in CSS3.

```
@keyframes animation {
  from {background-color: pink;}
   to {background-color: green;}
}
div { width: 100px; height: 100px;
   background-color: red; animation-name: animation;
   animation-duration: 5s;
}
```

CSS3 Nicely Formatted Columns

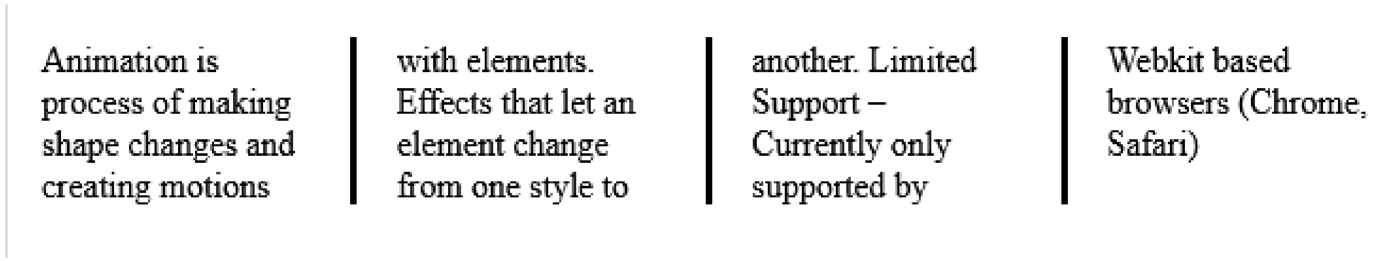
- CSS3 supported multi columns to arrange the text as news paper structure.
- Putting content into columns is super easy
- Again, No IE support

CSS3 Nicely Formatted Columns

- Values
  - column-count: Used to count the number of columns that element should be divided.
  - column-fill: Used to decide, how to fill the columns.
  - column-gap: Used to decide the gap between the columns.
  - column-rule: Used to specifies the number of rules.
  - rule-color: Used to specifies the column rule color.
  - rule-style: Used to specifies the style rule for column.
  - rule-width: Used to specifies the width.
  - column-span: Used to specifies the span between columns.

CSS3 Features

CSS3 Multi Columns



Animation is process of making shape changes and creating motions with elements. Effects that let an element change from one style to another. Limited Support – Currently only supported by Webkit based browsers (Chrome, Safari)

```
<html>   <head>     <style>
        .multi {
                /* Column count property */
           -webkit-column-count: 4;
           -moz-column-count: 4;
           column-count: 4;
                /* Column gap property */
           -webkit-column-gap: 40px;
           -moz-column-gap: 40px;
           column-gap: 40px;
                /* Column style property */
           -webkit-column-rule-style: solid;
           -moz-column-rule-style: solid;
           column-rule-style: solid;      }
     </style> </head>
<body>  <div class = "multi">
Animation is process of making shape changes and creating
motions with elements.Effects that let an element change
from one style to another.Limited Support – Currently only
supported by Webkit based browsers (Chrome, Safari)
]</div> </body> </html>
```

CSS3 Font Faces

- Allows you to use non-standard fonts
- Mixed support for otf, ttf, svg and eot
- Things to remember:

– Internet Explorer only supports EOT

– Mozilla browsers support OTF and TTF

– Safari and Opera support OTF, TTF and SVG

– Chrome supports TTF and SVG.

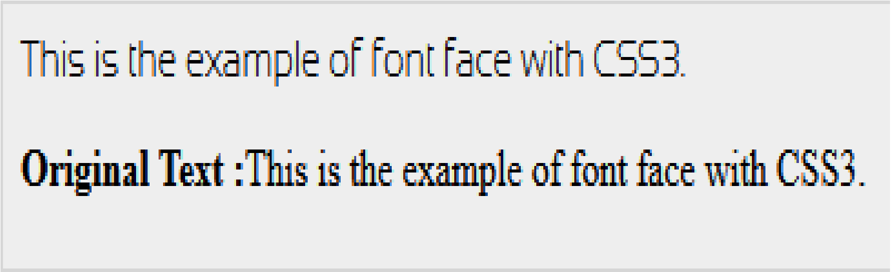
- Format:

```
@font-face
{
   font-family:<makeupaname>;
   src:url(‘localfontname.ttf’) ||
   url(‘http://fontlocation’);
}
```

CSS3 Different web fonts formats

- TrueType Fonts (TTF): TrueType is an outline font standard developed by Apple and Microsoft in the late 1980s, It became most common fonts for both windows and MAC operating systems.
- OpenType Fonts (OTF): OpenType is a format for scalable computer fonts and developed by Microsoft
- The Web Open Font Format (WOFF): WOFF is used for develop web page and developed in the year of 2009. Now it is using by W3C recommendation.
- SVG Fonts/Shapes: SVG allow SVG fonts within SVG documentation. We can also apply CSS to SVG with font face property.
- Embedded OpenType Fonts (EOT): EOT is used to develop the web pages and it has embedded in webpages so no need to allow 3rd party fonts

CSS3 Different web fonts formats



This is the example of font face with CSS3.

Original Text : This is the example of font face with CSS3.

```
<html>
   <head>
     <style>
       @font-face {
        font-family: myFirstFont;
        src: url(/css/font/SansationLight.woff);
        }
        div {
         font-family: myFirstFont;
         }
      </Style>
   </head>
   <body>
   <div>
    This is the example of font face with CSS3.
   </div>
   <p>
    <b>Original Text :   </b>
    This is the example of font face with
     CSS3.
   </p>
   </body>
</html>
```

CSS3 Fonts Descriptions

- font-family: Used to defines the name of font
- Src: Used to defines the URL
- font-stretch: Used to find, how font should be stretched
- font-style: Used to defines the fonts style
- font-weight: Used to defines the font weight(boldness)

Responsive Web Design

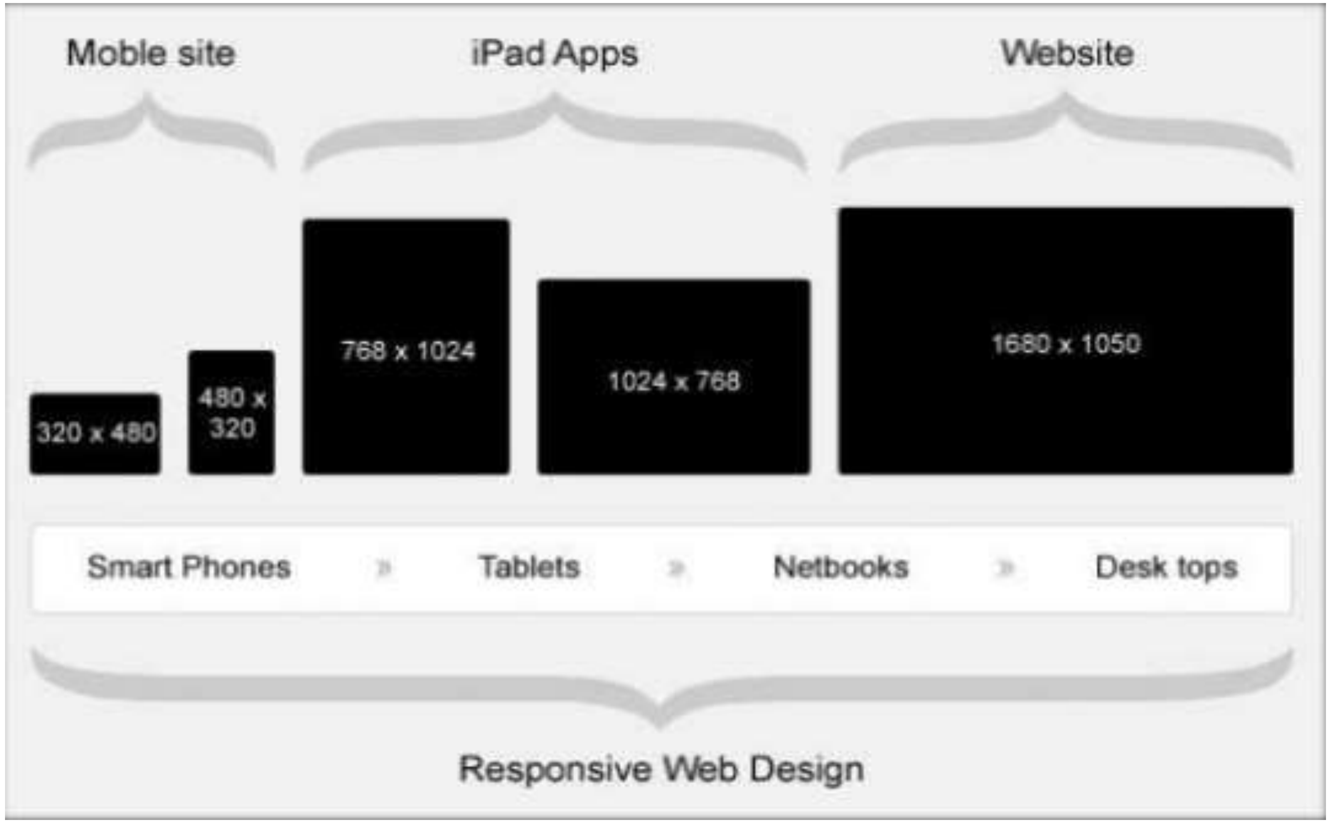
- Responsive web design makes the web page look good on all devices.
- The term Responsive Web Design was given by Ethan Marcotte.
- Web pages can be viewed using many different devices: desktops, tablets, and phones.
- It is called responsive web design when we use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

CSS3 Features

Screen resolutions

There’s no common or standard resolution its based on the websites requirements:

- Smartphone: 480px and below
- Phones to tablets: 767px and below
- Portrait Tablet: 768px and above
- Netbook: 990px to 1024px
- Desktop: 10240px and above



The Viewport

- The viewport is the user's visible area of a web page.
- The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.
- Before tablets and mobile phones, web pages were designed only for computer screens, and it was common for web pages to have a static design and a fixed size.
- Then, when we started surfing the internet using tablets and mobile phones, fixed size web pages were too large to fit the viewport.
- Users are used to scroll websites vertically on both desktop and mobile devices - but not horizontally!
- So, if the user is forced to scroll horizontally, or zoom out, to see the whole web page it results in a poor user experience.

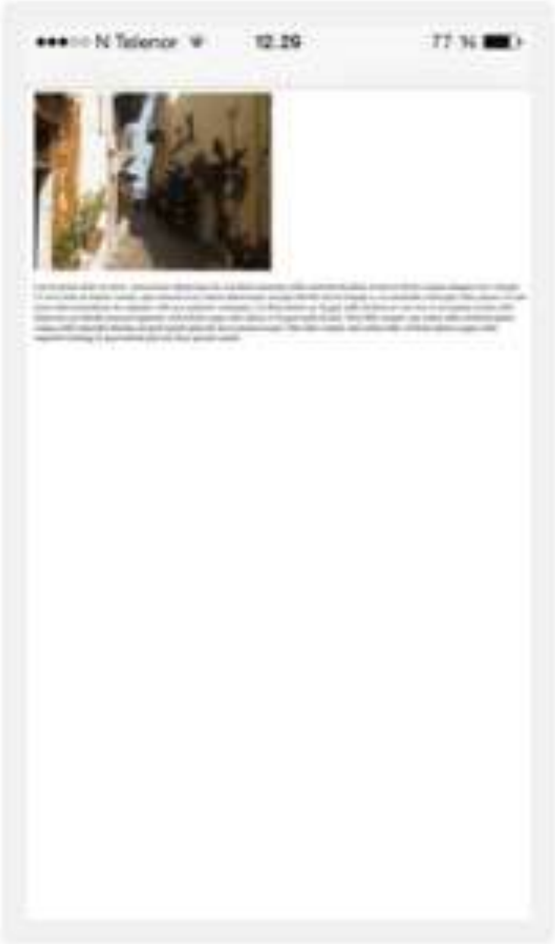
The Viewport - Setting The Viewport

- HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.
- The <meta> tag does not display on the webpage, but it is used by browsers which scan the site or webpage to know about the webpage.

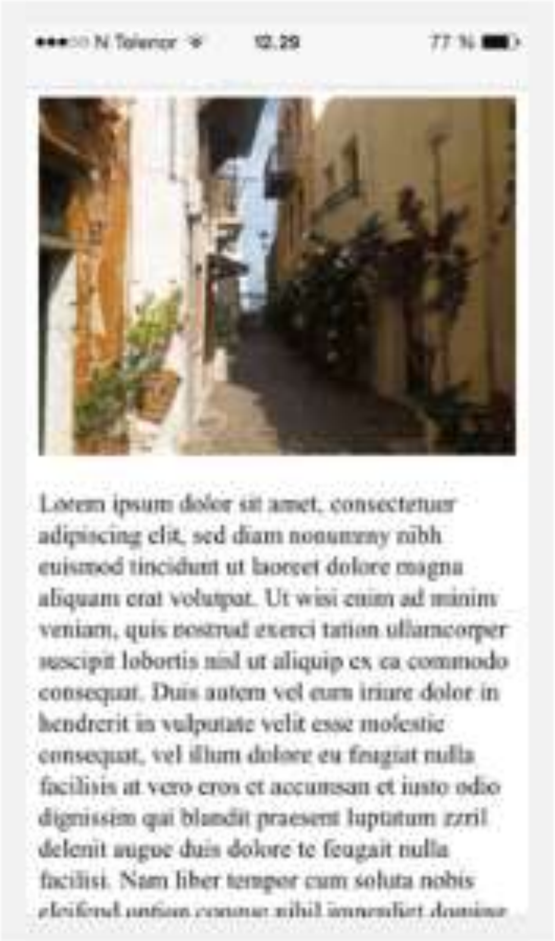
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- This gives the browser instructions on how to control the page's dimensions and scaling.
- The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
- The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

The Viewport



Without the viewport meta tag



Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

With the viewport meta tag

CSS3 Media Queries

- **CSS media queries** is used to assign different stylesheets depending on browser window size.
- Media queries can be used to check many things, such as:
  - width and height of the viewport
  - width and height of the device
  - orientation (is the tablet/phone in landscape or portrait mode)
  - resolution
- Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones.

CSS3 Media Queries

Media Query Syntax

- A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.

```
@media not|only mediatype and (expressions) {
  CSS-Code;
}
```

- A media type, which tells the browser what kind of media this code is for (e.g. print, or screen).
- A media expression, which is a rule, or test that must be passed for the contained CSS to be applied.
- A set of CSS rules that will be applied if the test passes and the media type is correct.
- Unless we use the not or only operators, the media type is optional and the all type will be implied.

CSS3 Media Queries

Media Query Syntax

The CSS media query syntax for calling an external stylesheet is like this:

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css“>
```

CSS3 Media Queries

Media Query Types

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

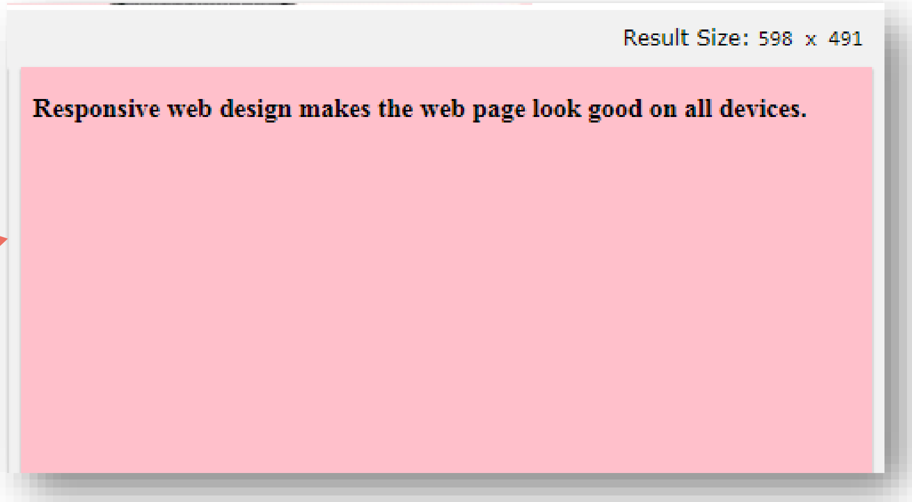
CSS3 Media Queries

Media Query Expressions

Value	Description
width	The viewport width
height	The viewport height
max-width	The maximum width of the display area, such as a browser window
min-width	The minimum width of the display area, such as a browser window
max-height	The maximum height of the display area, such as a browser window
min-height	The minimum height of the display area, such as a browser window
orientation	The orientation of the viewport (landscape or portrait mode)

CSS3 Media Queries Expressions -Width

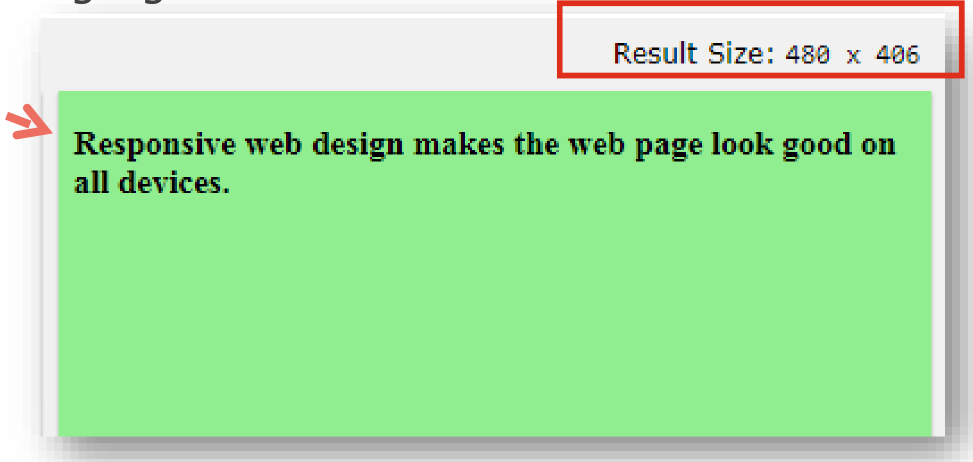
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body {
  background-color: pink;
}
@media screen and (width: 480px) {
  body {
    background-color: lightgreen;
  }
}
</style>
</head>
<body>
<p>Responsive web design makes the web page look good on all devices.</p>
</body>
</html>
```



Result Size: 598 x 491

Responsive web design makes the web page look good on all devices.

When browser is minimized to width 480px the background color changes to lightgreen

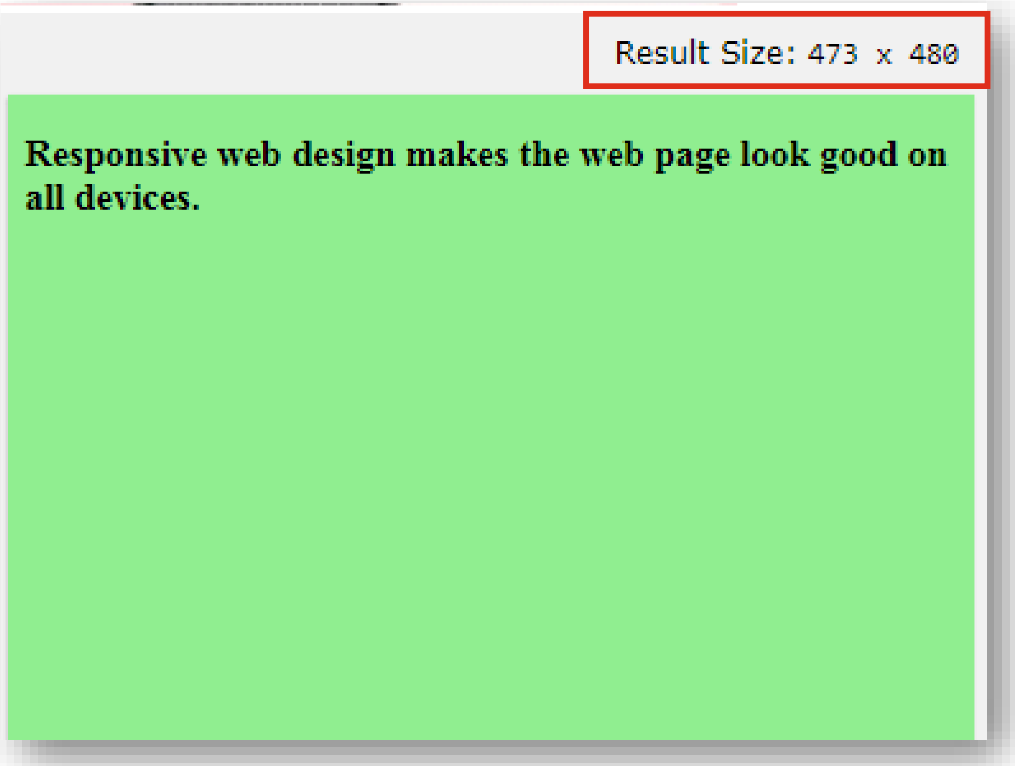


Result Size: 480 x 406

Responsive web design makes the web page look good on all devices.

CSS3 Media Queries Expressions - Height

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body {
  background-color: pink;
}
@media screen and (height: 480px) {
  body {
    background-color: lightgreen;
  }
}
</style>
</head>
<body>
<p>Responsive web design makes the web page look good on all devices.</p>
</body>
</html>
```



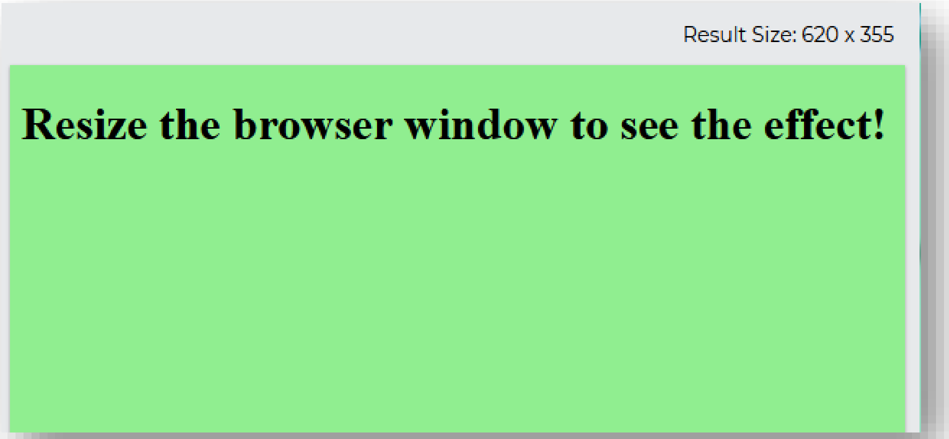
Result Size: 473 x 480

Responsive web design makes the web page look good on all devices.

CSS3 Media Queries Expressions – Min-width

To changes the background-color to lightgreen if the viewport is 480 pixels wide or wider

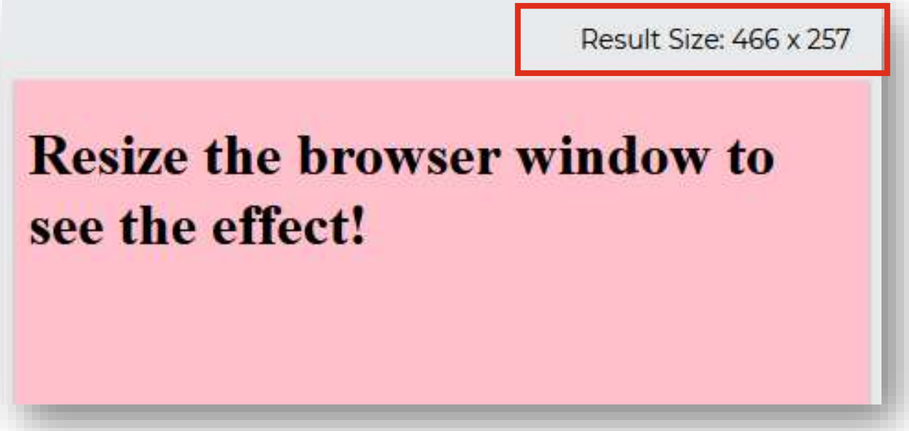
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<style>
body {
  background-color: pink;
}
@media screen and (min-width: 480px) {
  body {
    background-color: lightgreen;
  }
}
</style>
</head>
<body>
<p>Resize the browser window to see the effect!</p>
</body>
</html>
```



Result Size: 620 x 355

Resize the browser window to see the effect!

If the viewport is less than 480 pixels, the background-color will be pink



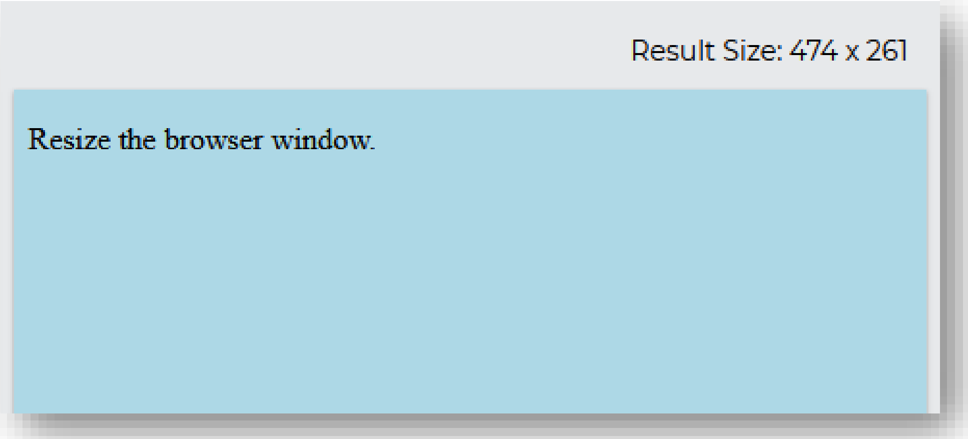
Result Size: 466 x 257

Resize the browser window to see the effect!

CSS3 Media Queries Expressions – Max-width

To changes the background-color to lightblue if the viewport is 600 pixels or less

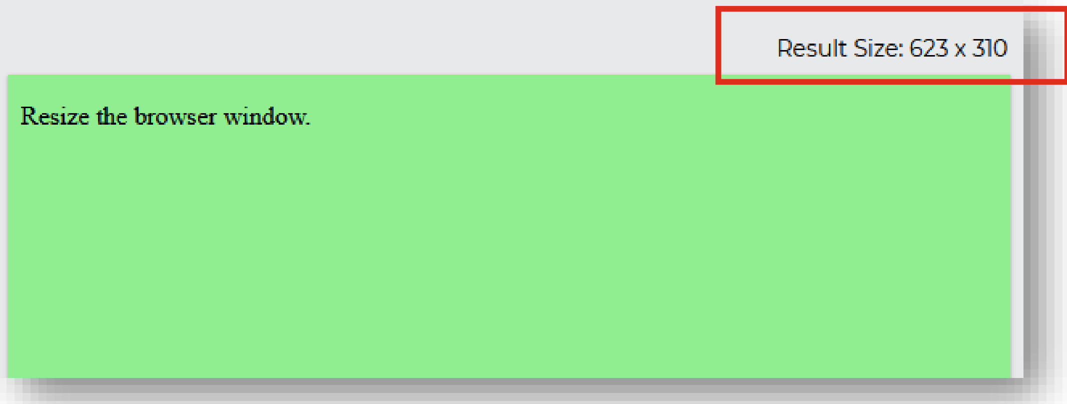
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body {
  background-color: lightgreen;
}
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }}
</style>
</head>
<body>
<p>Resize the browser window. </p>
</body>
</html>
```



Result Size: 474 x 261

Resize the browser window.

If the viewport is more than 600 pixels, the background-color will be lightgreen



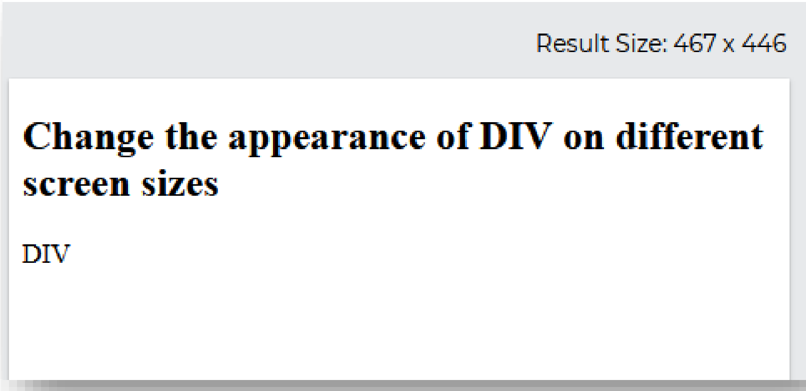
Result Size: 623 x 310

Resize the browser window.

CSS3 Media Queries Expressions – Min Width to Max Width

(max-width: ..) and (min-width: ..) values are used to set a minimum width and a maximum width. For example, when the browser's width is between 600 and 900px, change the appearance of a <div> element:

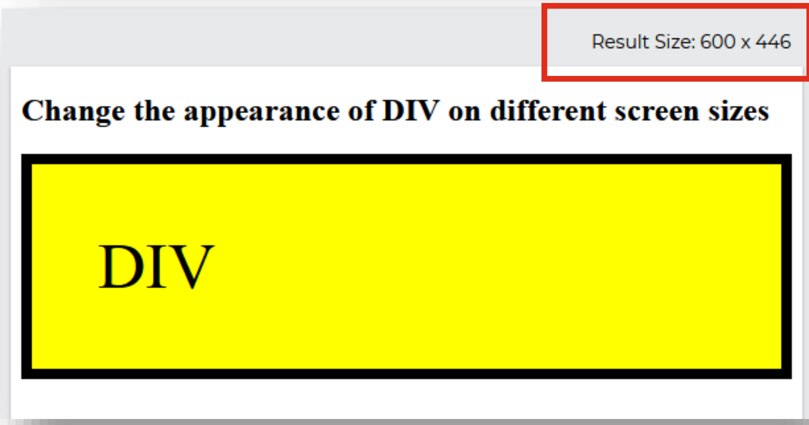
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
@media screen and (max-width: 900px) and (min-width: 600px) {
  div.example {
    font-size: 50px;
    padding: 50px;
    border: 8px solid black;
    background: yellow;
  }}
</style></head>
<body>
<h2>Change the appearance of DIV on different screen sizes</h2>
<div class="example"> DIV</div>
</body></html>
```



Result Size: 467 x 446

Change the appearance of DIV on different screen sizes

DIV



Result Size: 600 x 446

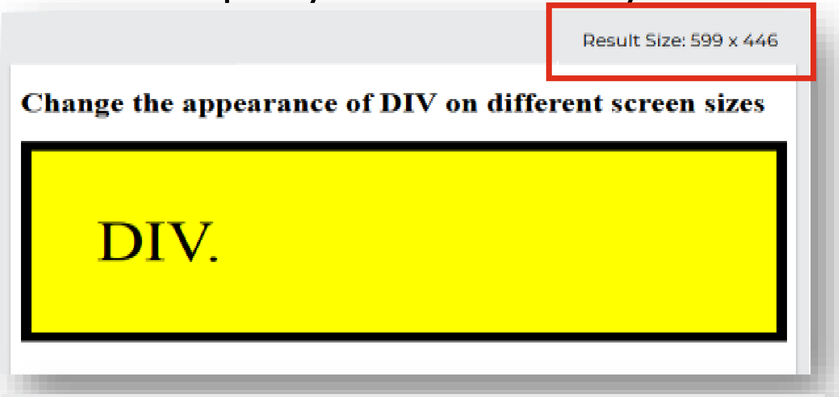
Change the appearance of DIV on different screen sizes

DIV

CSS3 Media Queries Expressions – Min Width to Max Width

Using an additional value: In the example below, we add an additional media query to our already existing one using a comma (this will behave like an OR operator):

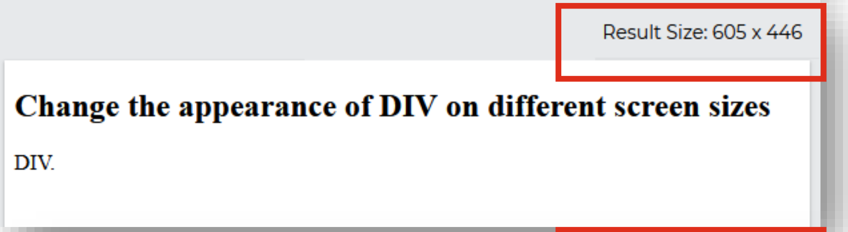
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
@media screen and (max-width: 600px) and (min-width: 400px), (min-
width: 650px) {
  div.example{
    font-size: 50px;
    padding: 50px;
    border: 8px solid black;
    background: yellow;
  }}
</style></head>
<body>
<h2>Change the appearance of DIV on different screen sizes</h2>
<div class="example">DIV.</div>
</body></html>
```



Result Size: 599 x 446

Change the appearance of DIV on different screen sizes

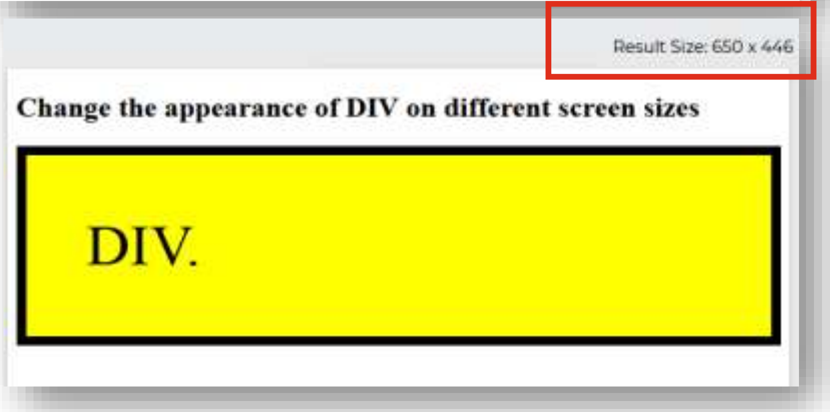
DIV.



Result Size: 605 x 446

Change the appearance of DIV on different screen sizes

DIV.



Result Size: 650 x 446

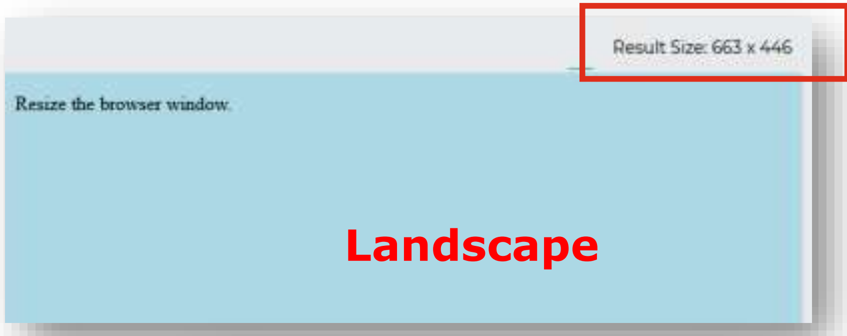
Change the appearance of DIV on different screen sizes

DIV.

CSS3 Media Queries Orientation: Portrait / Landscape

- Media queries can also be used to change layout of a page depending on the orientation of the browser.
- We can have a set of CSS properties that will only apply when the browser window is wider than its height, a so called "Landscape" orientation:

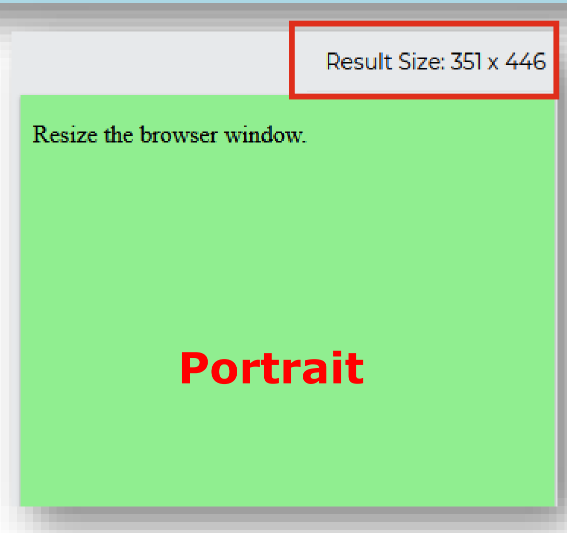
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body {
  background-color: lightgreen;
}
@media only screen and (orientation: landscape) {
  body {
    background-color: lightblue;
  }
}
</style></head>
<body>
<p>Resize the browser window. </p>
</body></html>
```



Result Size: 663 x 446

Resize the browser window.

Landscape



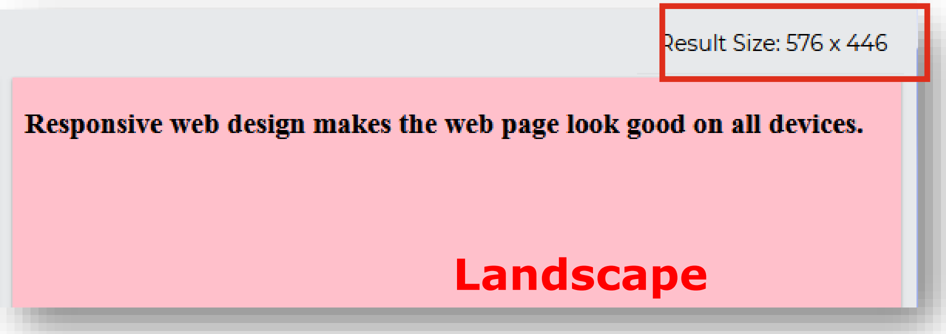
Result Size: 351 x 446

Resize the browser window.

Portrait

CSS3 Media Queries Orientation: Portrait / Landscape

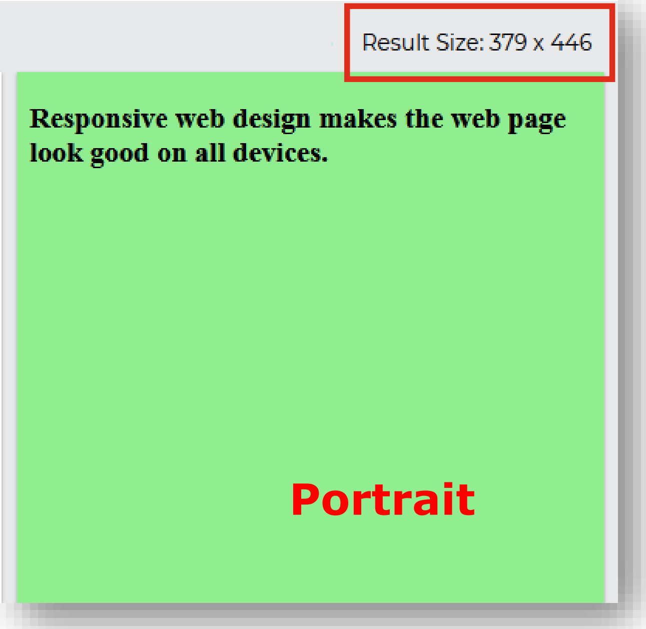
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: pink;
}
@media not screen and (orientation: landscape) {
  body {
    background-color: lightgreen;
  }
}
</style>
</head>
<body>
<h3>Responsive web design makes the web page look good on all
devices.</h3>
</body>
</html>
```



Result Size: 576 x 446

Responsive web design makes the web page look good on all devices.

Landscape



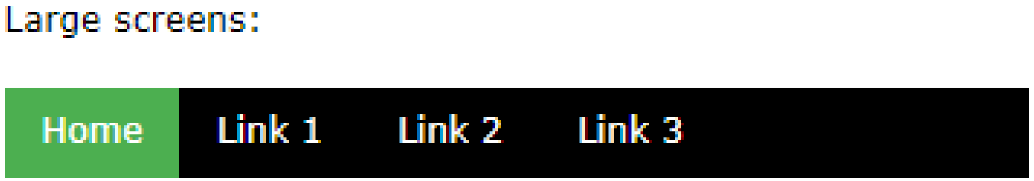
Result Size: 379 x 446

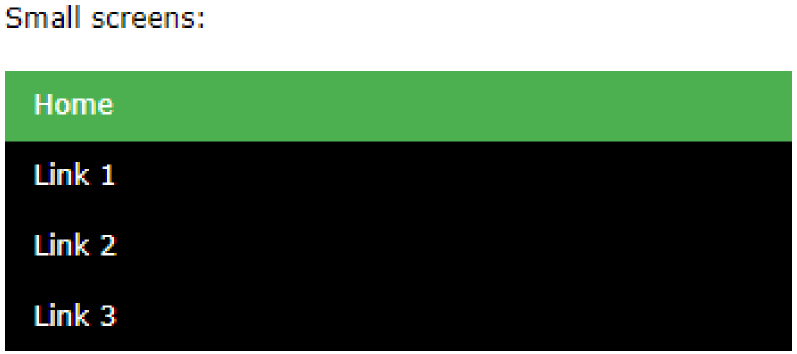
Responsive web design makes the web page look good on all devices.

Portrait

CSS3 Media Queries for Menu navbar

We use media queries to create a responsive navigation menu, that varies in design on different screen sizes.



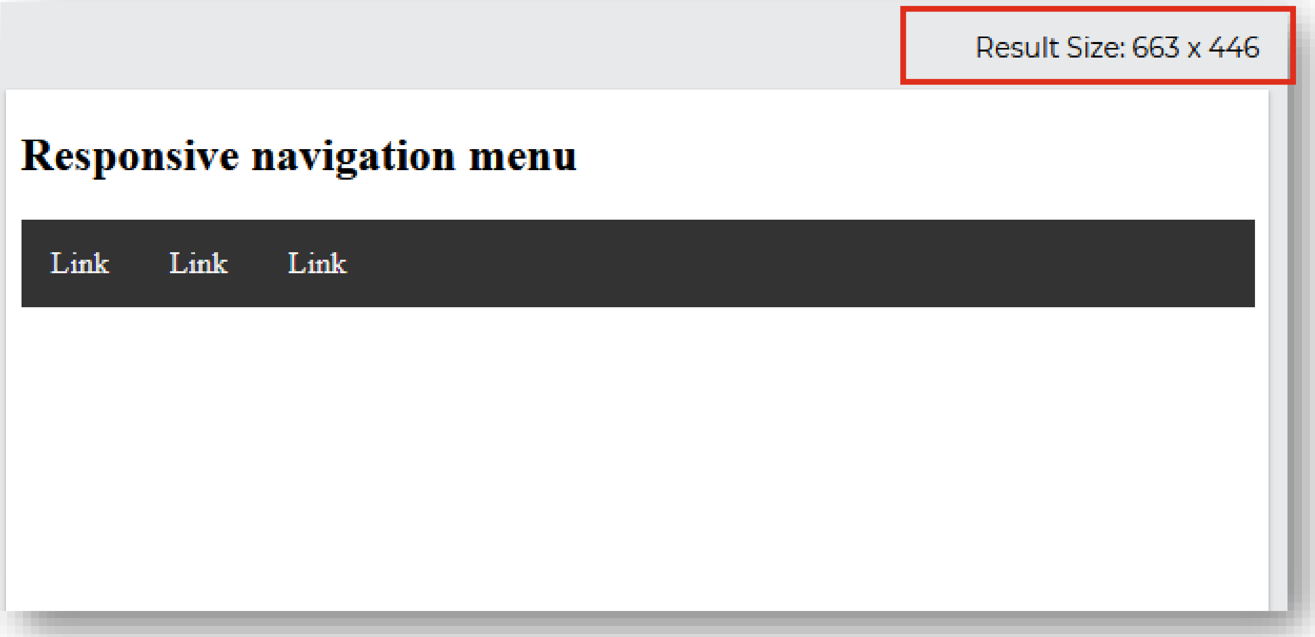


CSS3 Media Queries for Menu navbar

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;
}
/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}
/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
```

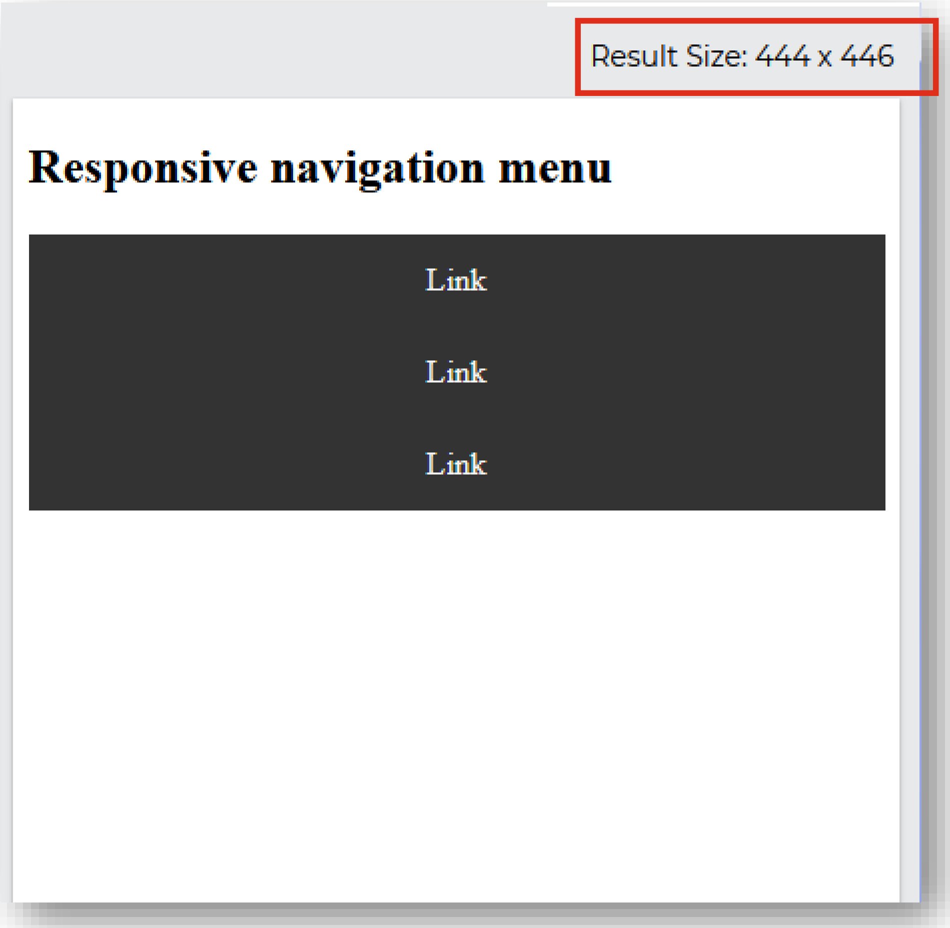
```
text-decoration: none;
}
/* On screens that are 600px wide or less, make the menu links stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .topnav a {
    float: none;
    width: 100%;
  }
}
</style>
</head>
<body>
<h2>Responsive navigation menu</h2>
<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
</div>
</body>
</html>
```

CSS3 Media Queries for Menu navbar



Result Size: 663 x 446

Responsive navigation menu

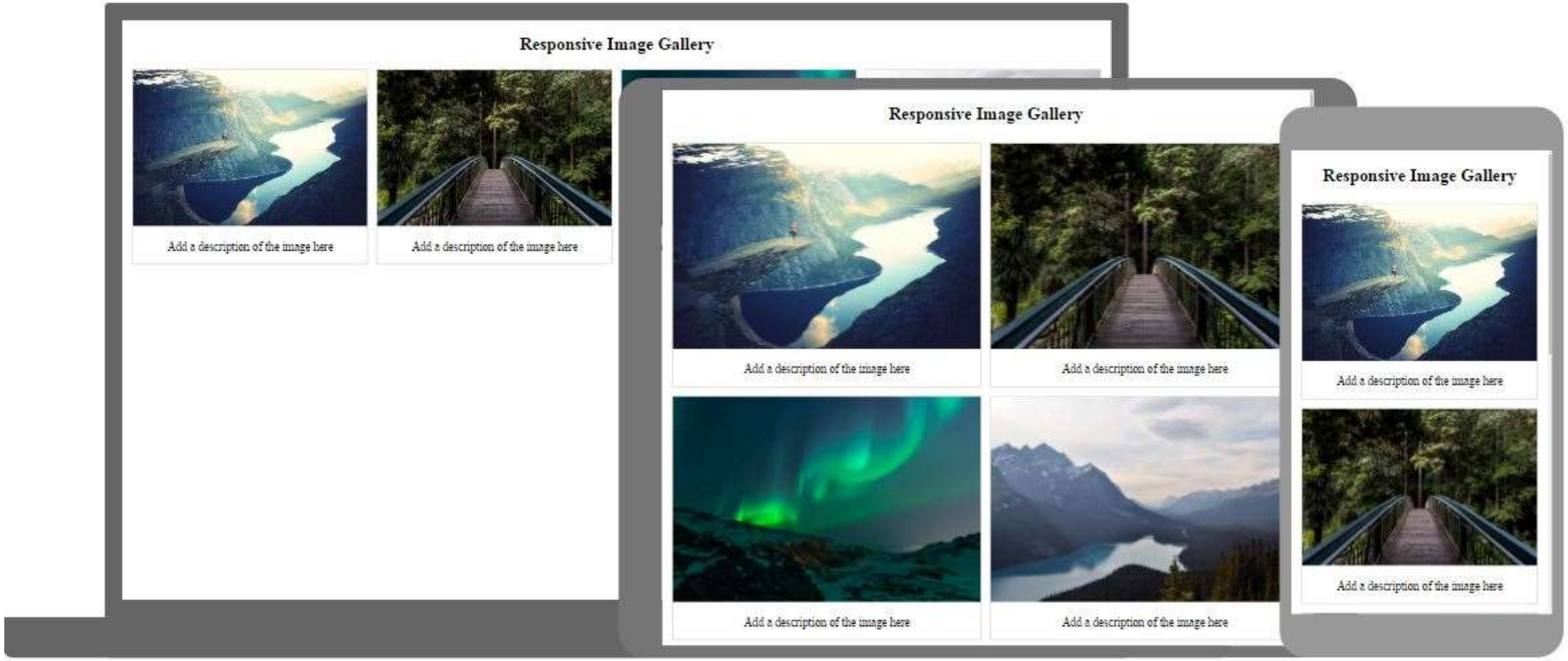


Result Size: 444 x 446

Responsive navigation menu

CSS3 Media Queries – Responsive Image Gallery

CSS media queries is used to create a responsive image gallery that will look good on desktops, tablets and smart phones.


Responsive Image Gallery
Add a description of the image here
Add a description of the image here
Responsive Image Gallery
Add a description of the image here
Add a description of the image here
Add a description of the image here
Add a description of the image here
Responsive Image Gallery
Add a description of the image here
Add a description of the image here

CSS3 Media Queries - Responsive Image Gallery

```
<style type="text/css">
* {
  box-sizing: border-box;
}
div.gallery {
  border: 1px solid #ccc;
}
div.gallery:hover {
  border: 1px solid #777;
}

div.gallery img {
  width: 100%;
  height: 150px;
}
div.desc {
  padding: 15px;
  text-align: center;
}
```

```
.responsive {
  float: left;
  padding: 6px;
  width: 25%;
}
@media only screen and (max-width: 700px)
{
  .responsive {
    width: 50%;
    margin: 6px 0;
  }
}
@media only screen and (max-width: 500px)
{
  .responsive {
    width: 100%;
  }
}
</style>
```

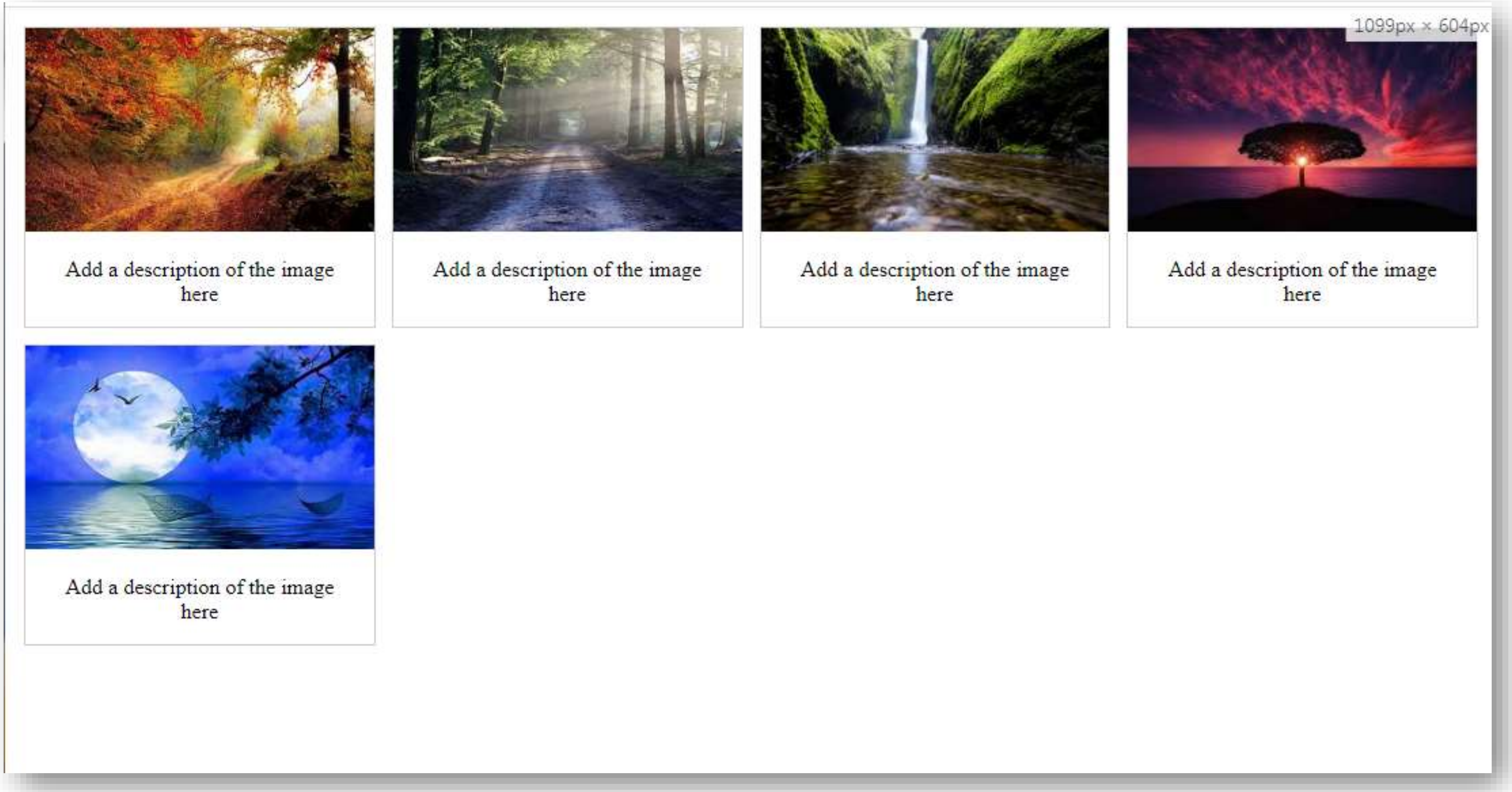
```
<body>
  <div class="responsive">
      <div class="gallery">
        <a target="_blank"
href="nature1.jpg">
          <img src="nature1.jpg"
alt="Nature 1" width="600"
height="400">
  </a>
          <div class="desc">Add a
description of the image here </div>
       </div>
</div>
```

CSS3 Media Queries - Responsive Image Gallery

```html
<div class="responsive">
<div class="gallery">
  <a target="_blank" href="nature2.jpg">
    <img src="nature2.jpg" alt="Nature 2" width="600"
height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>
</div>
 <div class="responsive">
<div class="gallery">
  <a target="_blank" href="nature3.jpg">
    <img src="nature3.jpg" alt="Nature 3" width="600"
height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>
</div>
```

```html
 <div class="responsive">
<div class="gallery">
  <a target="_blank" href="nature4.jpg">
    <img src="nature4.jpg" alt="Nature 4" width="600"
height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>
</div>
 <div class="responsive">
<div class="gallery">
  <a target="_blank" href="nature5.jpg">
    <img src="nature5.jpg" alt="Nature 5" width="600"
height="400">
  </a>
  <div class="desc">Add a description of the image here</div>
</div>
</div>
</body>
</html>
```

CSS3 Media Queries - Responsive Image Gallery



1099px × 604px

Add a description of the image here

Add a description of the image here

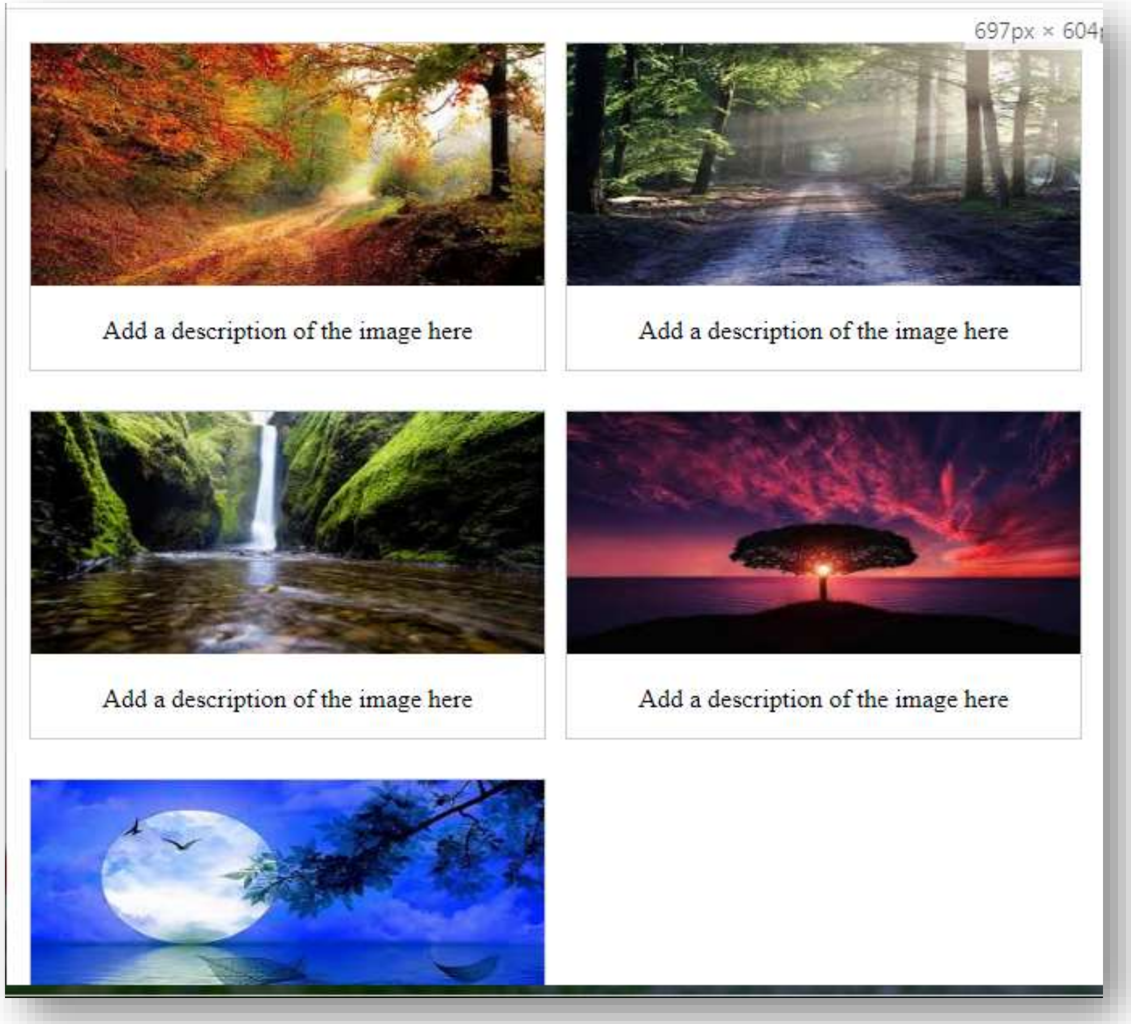
Add a description of the image here

Add a description of the image here

Add a description of the image here

CSS3 Media Queries - Responsive Image Gallery

@media only screen and (max-width: 700px)



697px × 604

Add a description of the image here

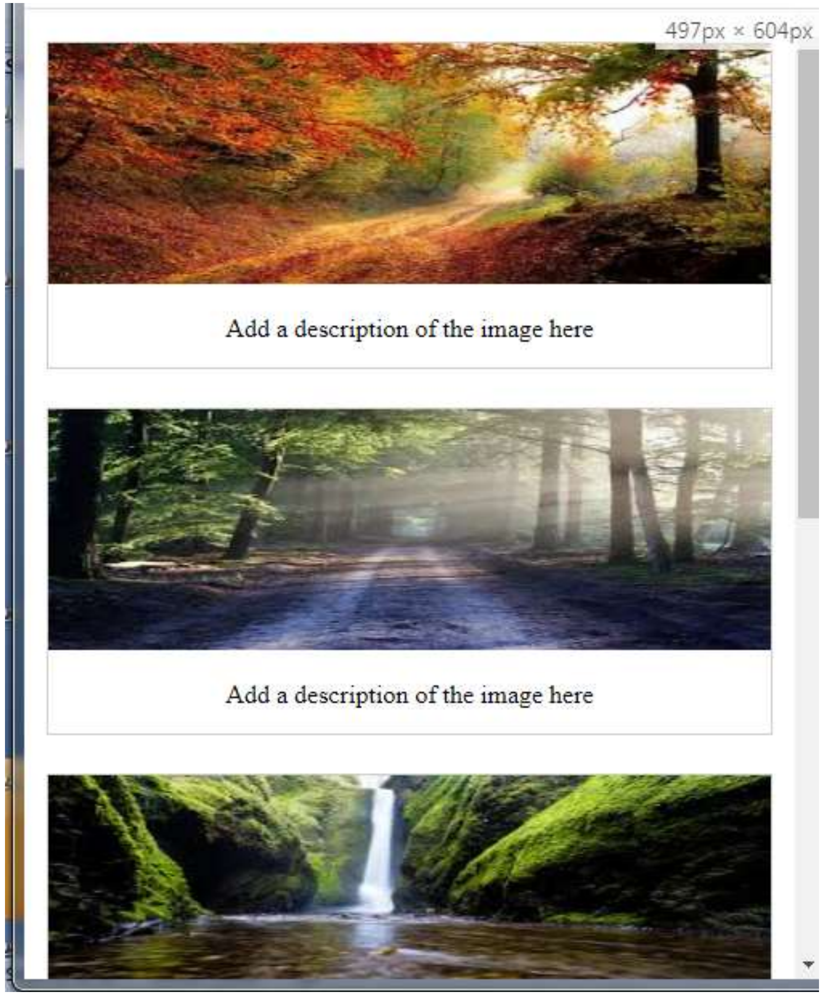
Add a description of the image here

Add a description of the image here

Add a description of the image here

CSS3 Media Queries - Responsive Image Gallery

@media only screen and (max-width: 500px)



Add a description of the image here

Add a description of the image here

Quiz

1. **What does a stand for in RGBa?**

a) alpha

b) aqua

c) Apple

d) about

Ans: a) alpha

Quiz

2. What are the first three values of text-shadow in order?

a) vertical, blur, horizontal

b) blur, vertical, horizontal

c) vertical, horizontal, blur

d) horizontal, vertical, blur

Ans: d) horizontal, vertical, blur

Quiz

3. How do four values work on border-radius?

a) top, bottom, left, right

b) up, down, front, behind

c) top-left, top-right, bottom-right, bottom-left

d) bottom-left, bottom-right, top-right, top-left

Ans: c) top-left, top-right, bottom-right, bottom-left

Quiz

4. How do you add shadow to elements in CSS3?

a) box-shadow: 10px 10px 5px grey;

b) shadow-right:10px shadow-bottom:10px; shadow-color: grey;

c) shadow-color: grey;

d) alpha-effect[shadow]: 10px 10px 5px grey

Ans: a) box-shadow: 10px 10px 5px grey;

Quiz

5. How to force a word wrap using CSS3?

a) word-wrap: break-word;

b) text-wrap: break-word

c) text-wrap: force;

d) text-width: set;

Ans: a) word-wrap: break-word

Quiz

6. How to rotate objects using CSS3?

a) object-rotation: 30deg;

b)transform:rotate(30deg);

c) rotate-object: 30deg;

d) transform: rotate-30deg-clockwise;

Ans: b) transform: rotate(30deg);

Quiz

7. If you put a value of 0 for a Border-Radius what will happen?

a) The Corner will curve horizontal.

b) The Corner will be square

c) The Corner will curve vertical

d) The world will end

Ans: b) The Corner will be square

Quiz

8. The ______________ selector is used to specify a style for a single, unique element.

a) id

b) class

c) text

d) bit

Ans: a) id

Quiz

9. The CSS property used to specify the transparency of an element is

a) opacity

b) filter

c) visibility

d) overlay

Ans: c) opacity

Quiz

10. The CSS property used to make the rounded borders, or rounded corners around an element is

a) border-collapse

b) border-radius

c) border-spacing

d) None of the above

Ans: b) border-radius

THANK YOU

SmartCliff
Career Mobility Solutions