Web Design Course

STRUCTURING STYLING WITH CSS

Lecture by Mr. Kaung Sett

Instructional Units in the Module

IU #	IU Description	Required / Optional
01	Web Design Concepts	Required
02	HTML Basics	Required
03	Advanced HTML & Web Browsers	Required
04	**Structuring & Styling with CSS**	**Required**
05	Working with CSS : An Example	Required
06	Javascript Basics	Required
07	Advanced Javascript	Required

Structuring & Styling with CSS

S. No.	Topic Description
01	CSS Introduction
02	CSS Editor
03	Creating & Using CSS
04	CSS Placement in HTML
05	CSS Syntax
06	CSS Box Model
07	Backgrounds
08	Borders
09	Margins
10	Padding
11	Height & Width
12	Outline

Structuring & Styling with CSS

S. No.	Topic Description
13	Text
14	Font
15	Icons
16	Links
17	CSS Overflow
18	Css Float & Clear
19	Image Sprites
20	Styling List in CSS
21	Styling Table in CSS
22	Forms

- **CSS** stands for Cascading **S**tyle **S**heets.
- language for describing the presentation of Web pages, including colors, layout, and fonts.
- Allows adapting the presentation to different types of devices, such as large screens, small screens, or printers.
- Independent of HTML and can be used with any XML-based markup language
- describes how HTML elements are to be displayed on paper, screen or in other media.
- saves a lot of work. can control the layout of various web pages all at a time.
- controls the look and feel part of a web page.
- is easy to learn and understand also it provides powerful control over the presentation of an HTML document

- Control the color of the text
- The style of fonts
- The spacing between paragraphs
- Columns are sized and laid out
- background images or colors are used
- Layout designs
- Variations in display for different devices and screen sizes as well as a variety of other effects.

Advantages of CSS

- **CSS saves time**
  - Write CSS once and reuse same sheet in multiple HTML pages.
  - Define a style for each HTML element and apply it to as many Web pages as required.
- **Pages load faster**
  - If you are using CSS, there is no need to write HTML tag attributes every time.
  - Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance**
  - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML**
  - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

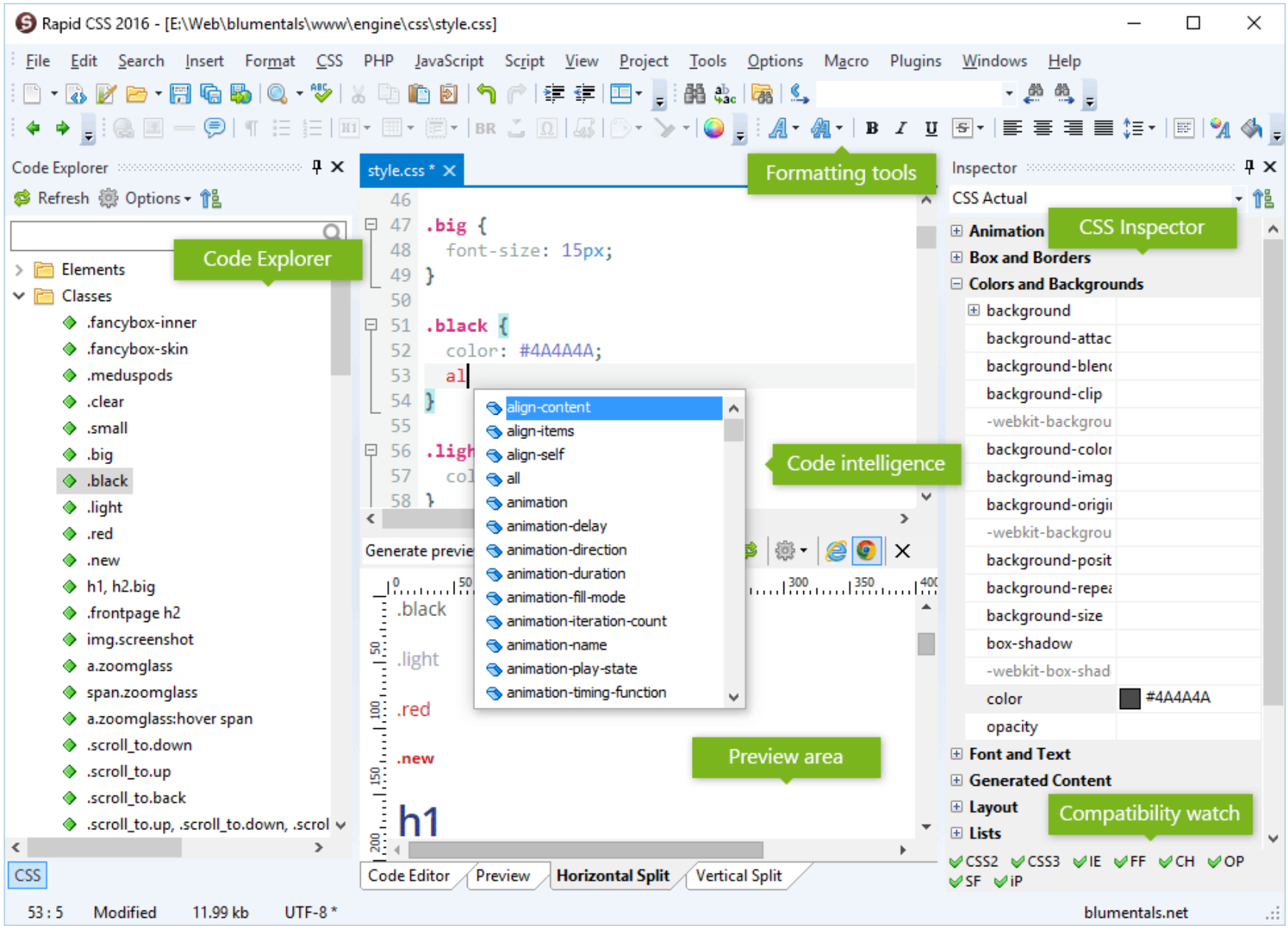
Advantages of CSS

- **Platform Independence**
  - CSS offers platform independence and can support latest browsers as well.

- **Multiple Device Compatibility**
  - Style sheets allow content to be optimized for more than one type of device.
  - By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

- Download and install Evaluation version of RapidCSS Editor from www.rapidcsseditor.com
- Rapid CSS editor makes it easy to create, design and edit modern CSS-based websites.
- Write the CSS code manually or let the style sheet editor do it for you
- It is easy because of the many features such as auto complete, code inspector, CSS checker and instant built-in multi-browser preview.
- Rapid CSS editor is designed to save you time and make your job easier.

RapidCSS Editor



Rapid CSS Editor Features

- **Quick and lightweight**
  Loads much faster than any other CSS editor or IDE with similar features
- **Powerful syntax highlighting Updated!**
  Supports HTML, CSS, LESS, SASS, JavaScript, PHP, XML, ASP, Perl and more
- **Code intelligence Updated!**
  Tons of intelligent HTML and CSS code completion, navigation and suggestion features
- **Smart code re-use**
  Code snippet library and code templates with assignable shortcuts
- **HTML5 and CSS3 ready Updated**
  Coding features are up-to-date with modern standards
- **Direct FTP/SFTP/FTPS Updated**
  Edit directly on your web server or publish local development copy updates with a single click

RapidCSS Editor Features contd.

- **Mobile web development**
  Media queries, viewport assistant, screen-size preview
- **Advanced search and replace**
  Quick search, detailed search, file search, regular expression support, detailed results and more
- **Powerful CSS tools Updated!**
  Compatibility watch, prefixer, shadow assistant, box assistant, web font assistant and much more
- **Browser preview**
  Built-in multi-browser preview, split-screen mode, screen-size testing, XRay
- **Powerful color picker Updated!**
  Advanced color picker with project color management
- **Integrated validation Updated!**
  Spell checker, CSS checker, W3 HTML and CSS validator

Creating & Using CSS

- **CSS can be created and used by any of the following options**
  - Inline CSS
  - Internal CSS
  - External CSS
  - The precedence of style sheet is Inline (High Priority), Internal & External (Low Priority)
- **Creating Inline CSS**
  - Using a style attribute in HTML elements
  - This is done by editing the HTML page in Notepad or Textpad
  - You can place the style in tags such as span, div, p etc.
- **Creating Internal CSS**
  - Using a <style> element in the HTML <head> section
  - This is done by editing the HTML page in Notepad or Textpad
- **Creating External CSS Sheets**
  - Using one or more external CSS files (.css)
  - Use a tool such as RapidCSS to create .css file

CSS Placement in HTML

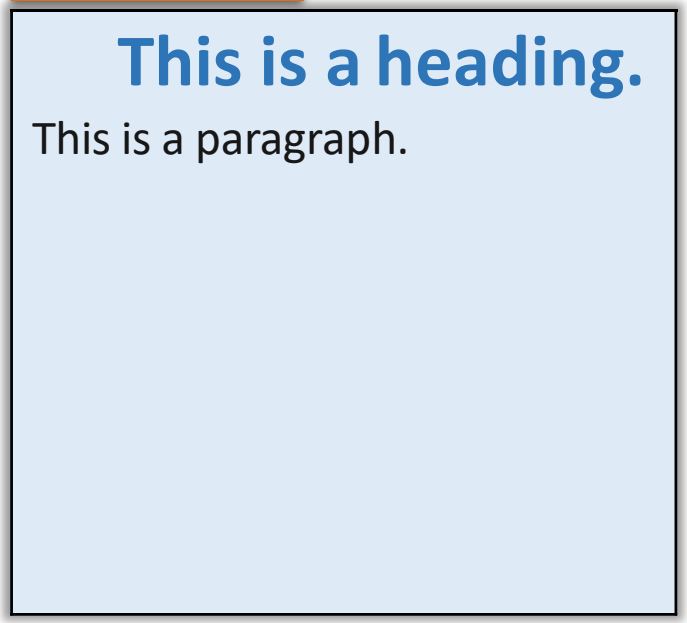
❑ Inline Styling (Inline CSS)

- Inline styling is used to apply a unique style to a single HTML element:
- To apply Inline styling the style attribute is used.
- To use inline styles, we should add the style attribute to the relevant element. The style attribute can contain any CSS property.

Example

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

Result



CSS Placement in HTML

- **Internal Styling (Internal CSS)**
  - Internal styling is used to define a style for only one HTML page.
  - Internal styling is defined in the <head> section of an HTML page, within a <style> element:

Example

```
!DOCTYPE html>
<html>
<head>
<style>
body {
   background-color: linen;
}
h1 {
   color: maroon;
   margin-left: 40px;
}
</style>
</head>
<body>
<h1>Thisis a heading</h1>
<p>This isa paragraph.</p>
</body>
</html>
```

Internal CSS between the head

Result

This is a heading

This is a paragraph.

CSS Placement in HTML

- **External Styling (External CSS)**
  - An external style sheet is used to define the style for many HTML pages.
  - With an external style sheet, you can change the look of an entire web site by making changes only in one file.
  - To use an external style sheet, add a link in the <head> section of the HTML page:

Example

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css"
href="mystyle.css">
</head><body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body> </html>
```

Result

This is a heading

This is a paragraph.

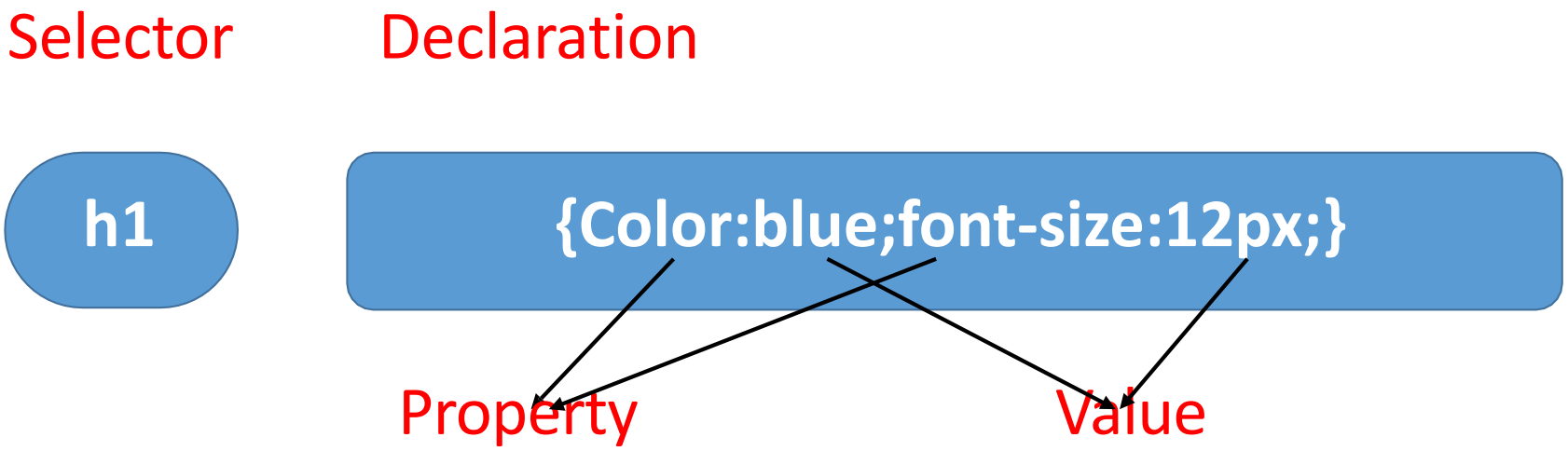
CSS Placement in HTML

- **External Styling (External CSS)**
  - 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.
  - Here is how the "myStyle.css" looks:

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

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

CSS Syntax

- **A CSS rule-set consists of a selector and a declaration block:**
  1. **Selector** – A selector is an HTML tag at which a style will be applied. It could be any tag like <h1> or <table> or class like .class or id like #id etc.
  2. **Property** - A property is a type of attribute of HTML tag where all the HTML attributes are converted into CSS properties. They could be color, border etc.
  3. **Value** - Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.



CSS Selector

Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
Element	p	Selects all <p> elements	1
element,element	div, p	Selects all elements and all <p> elements	1
element element	div p	Selects all <p> elements inside elements	1
element>element	div > p	Selects all <p> elements where the parent is a element	2
element+element	div + p	Selects all <p> elements that are placed immediately after elements	2
element1~element 2	p ~ ul	Selects every <ul> element that are preceded by a <p> element	3
[*attribute*]	[target]	Selects all elements with a target attribute	2

CSS Box Model

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

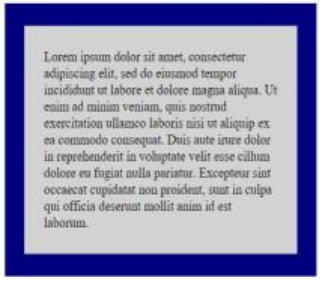
CSS Box Model

- **Explanation of the different parts:**
  1. **Content** - The content of the box, where text and images appear
  2. **Padding** - Clears an area around the content. The padding is transparent
  3. **Border** - A border that goes around the padding and content
  4. **Margin** - Clears an area outside the border. The margin is transparent
  5. The box model allows us to add a border around elements, and to define space between elements

```
div {
  background-color: lightgrey;
  width: 300px;
  padding: 25px;
  border: 25px solid navy;
  margin: 25px;
}
```

Demonstrating the Box Model

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 Box Model

❑ Width and Height of an Element

- To set the width and height of an element correctly in all browsers, you must know how the box model works.
- Assume we want to style a element to have a total width of 350px:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
</style>
</head><body>
<h2>Calculate the total width:</h2>
<img src="klematis4_big.jpg" width="350"height="263"
alt="Klematis">
<div>The picture above is 350px wide. The total width of this element
is also 350px.</div>
</body></html>
```

Result

The picture above is 350px wide. The total width of this element is also 350px.

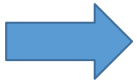
CSS

Backgrounds

- The CSS background properties are used to define the background effects for elements.
  - CSS background properties:
    - background-color
    - background-image
    - background-repeat
    - background-attachment
    - background-position

- **Background Color**
  - The background-color property specifies the background color of an element.
  - The background color of a page is set like this:

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



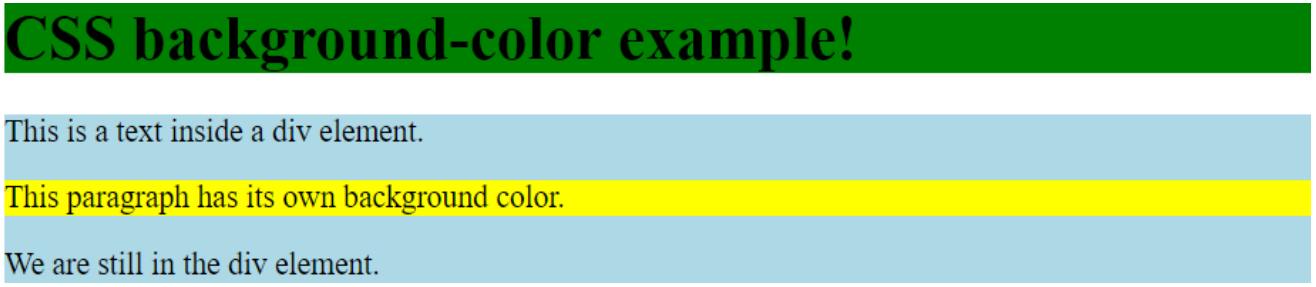
CSS Backgrounds - background-color

- In the example below, the <h1>, , and elements have different background colors:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {background-color: green; }
div {background-color: lightblue;}
p {background-color: yellow;}
</style>
</head>
<body>
```

- <h1>CSS background-color example!</h1>
-
- This is a text inside a div
- element.
- This paragraph has its own background color.
- We are still in the div element.
-

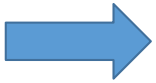
```
</body>
</html>
```

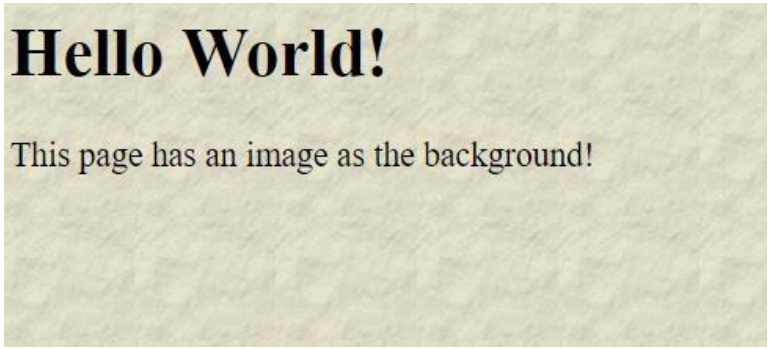


CSS Backgrounds - background-image

- The background-image property specifies an image to use as the background of an element.
- By default, the image is repeated so it covers the entire element.
- The background image for a page can be set like this:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image:
url("paper.gif");
}
</style>
</head><body>
<h1>Hello World!</h1>
<p>This page has an image as the
background!</p>
</body></html>
```

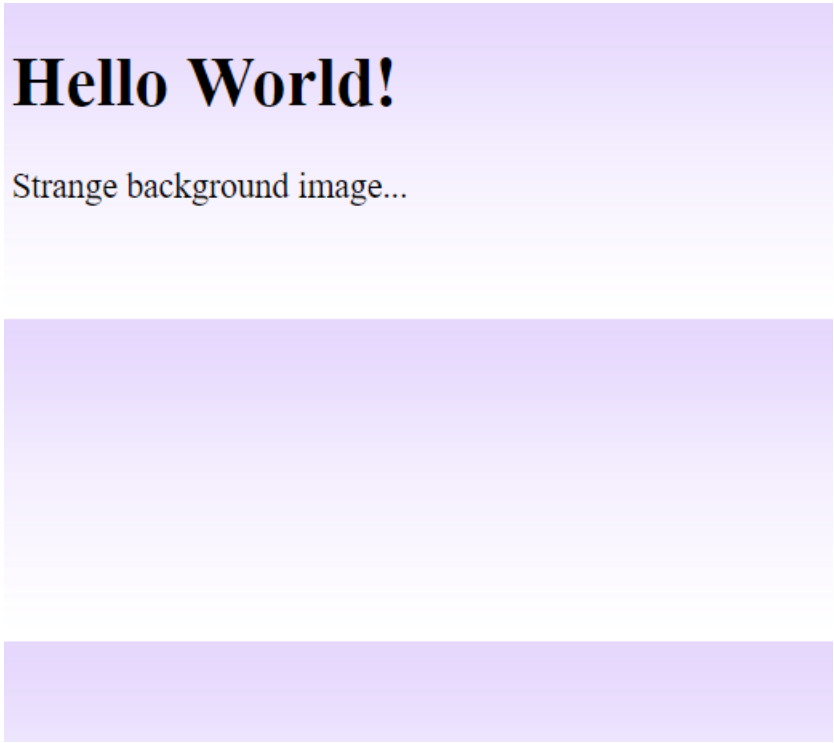




CSS Backgrounds - background-repeat

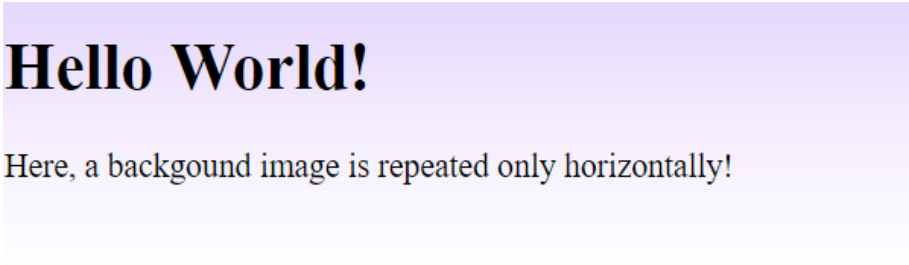
- By default, the background-image property repeats an image both horizontally and vertically.
- Some images should be repeated only horizontally or vertically, or they will look strange, like this:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image:
url("gradient_bg.png");
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>Strange background image...</p>
</body>
</html>
```



CSS Backgrounds - background-repeat

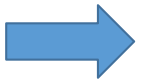
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image:
url("gradient_bg.png");
    background-repeat: repeat-x;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Here, a backgound image is
repeated only horizontally!</p>

</body>
</html>
```



CSS Backgrounds - background-repeat, background-repeat

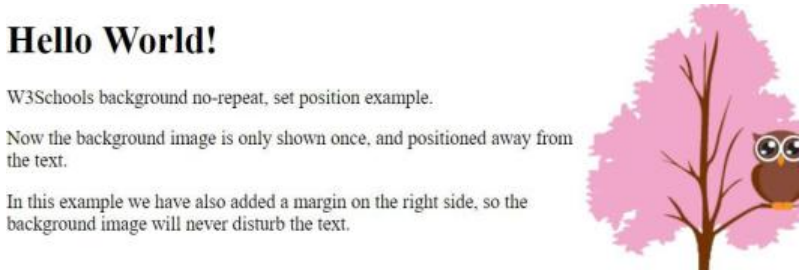
- Background Image - Set position and no-repeat
- Showing the background image only once is also specified by the background-repeat property:

```
body {
    background-
image: url("img_tree.png");
    background-repeat: no-repeat;
}
```



- In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

```
body {
    background-
image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right
top;
}
```



Borders

- The CSS border properties allow you to specify the style, width, and color of an element's border.

- **Border Style**
  - The border-style property specifies what kind of border to display.
    - dotted - Defines a dotted border
    - dashed - Defines a dashed border
    - solid - Defines a solid border
    - double - Defines a double border
    - groove - Defines a 3D grooved border. The effect depends on the border-color value

Borders

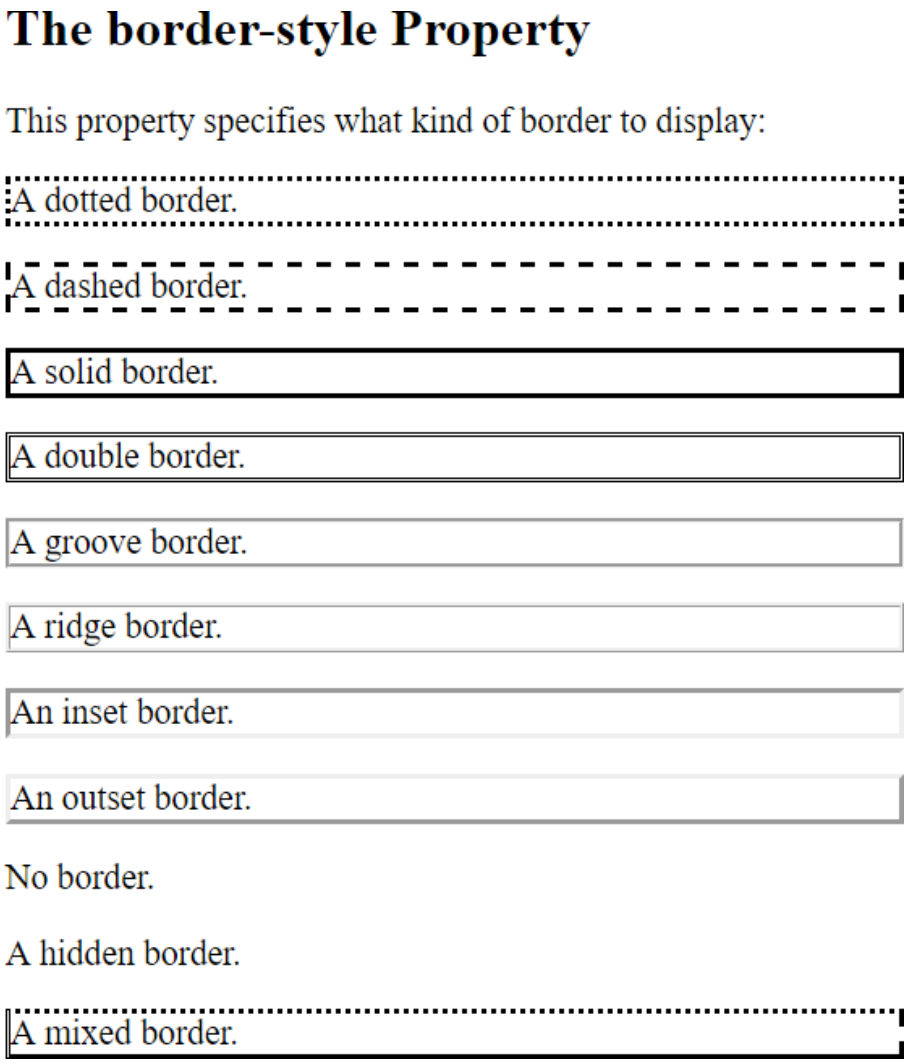
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border
- hidden - Defines a hidden border

Borders – border-style

Example

- Example

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed
solid double;}
```



Borders – border-width

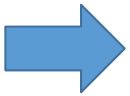
- The border-width property specifies the width of the four borders.
- The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.
- The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

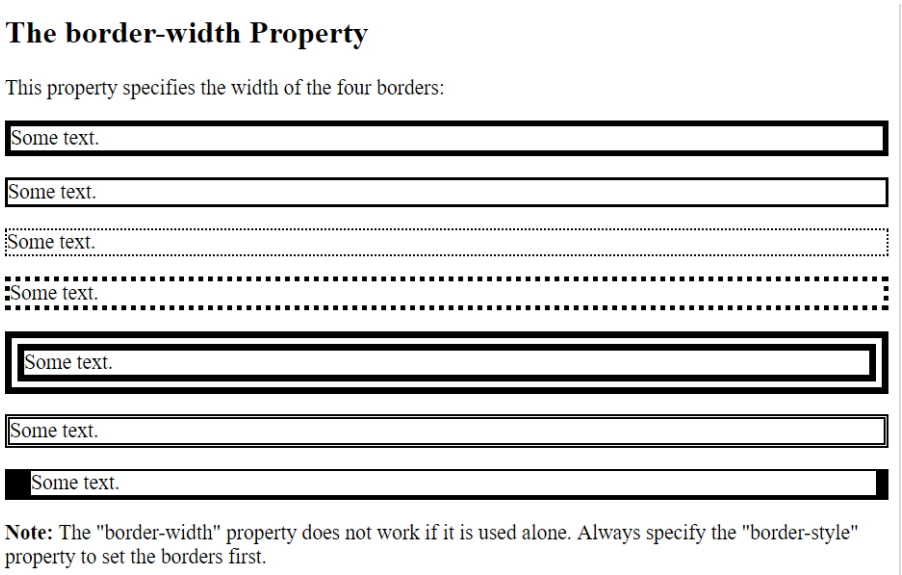
```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid; border-width: 5px;
}
p.two {
    border-style: solid; border-width: medium;
}
p.three {
    border-style: dotted; border-width: 2px;
}
p.four {
   border-style: dotted; border-width: thick;
}
p.five {
    border-style: double; border-width: 15px;
}
p.six {
    border-style: double; border-width: thick;
}
```

Borders – border-width Example

- Example

```
p.seven {
    border-style: solid; border-
width: 2px 10px 4px 20px;
}
</style>
</head>
<body>
<h2>The border-width Property</h2>
<p>This property specifies the width
of the four borders:</p>
<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>
<p class="four">Some text.</p>
<p class="five">Some text.</p>
<p class="six">Some text.</p>
<p class="seven">Some text.</p>
<p><b>Note:</b> The "border-width"
property does not work if it is used
alone.
Always specify the "border-style"
property to set the borders
first.</p>
</body>
</html>
```

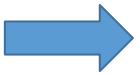




Borders - Border Color

- The border-color property is used to set the color of the four borders.
- The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).
- If border-color is not set, it inherits the color of the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {border-style: solid;border-
color: red;
}
</style>
</head>
<body>
<p class="one">A solid red
border</p>
</body>
</html>
```



A solid red border

Borders – Individual Sides

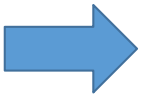
- In CSS, there is also properties for specifying each of the borders (top, right, bottom, and left)

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
}
</style>
</head>
<body>
<p>2 different border styles.</p>
</body>
</html>
```

2 different border styles.

Borders – Individual Sides

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    border-left: 6px solid red;
    background-color: lightgrey;
}
</style>
</head>
<body>

<h2>The border-left Property</h2>
<p>This property is a shorthand
property for border-left-width,
border-left-style, and border-
left-color.</p>

</body>
</html>
```



Margins

- The CSS margin properties are used to generate space around elements.
- The margin properties set the size of the white space outside the border.
- With CSS, you have full control over the margins. There are CSS properties for setting the margin for each side of an element (top, right, bottom, and left).
- **Margin - Individual Sides**
  - CSS has properties for specifying the margin for each side of an element
    - margin-top
    - margin-right
    - margin-bottom
    - margin-left

Margins - Individual Sides

- All the margin properties can have the following values:
  - auto - the browser calculates the margin
  - length - specifies a margin in px, pt, cm, etc.
  - % - specifies a margin in % of the width of the containing element
  - inherit - specifies that the margin should be inherited from the parent element
  - Negative values are allowed
- The following example sets different margins for all four sides of a <p> element:

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
}
```

This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

Margin Collapse

- Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins.
- This does not happen on left and right margins! Only top and bottom margins.

```
h1 {
    margin: 0 0 50px 0;
}

h2 {
    margin: 20px 0 0 0;
}
```

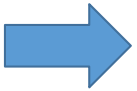
- In the example above, the <h1> element has a bottom margin of 50px. The <h2> element has a top margin set to 20px.
- Common sense would seem to suggest that the vertical margin between the <h1> and the <h2> would be a total of 70px (50px + 20px). But due to margin collapse, the actual margin ends up being 50px.

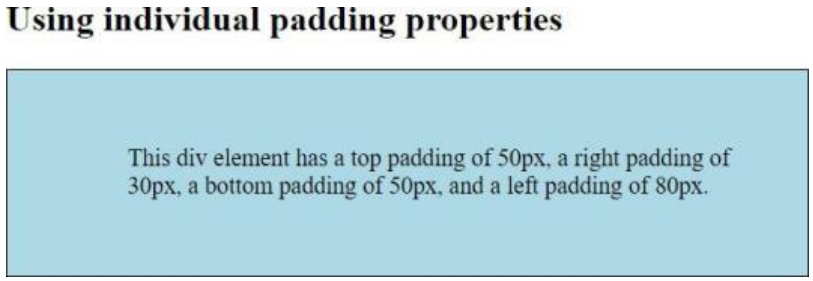
Paddi ng

- The CSS padding properties are used to generate space around content.
- The padding clears an area around the content (inside the border) of an element.
- With CSS, you have full control over the padding. There are CSS properties for setting the padding for each side of an element (top, right, bottom, and left).
- **Padding - Individual Sides**
  - CSS has properties for specifying the padding for each side padding-top, padding-right, padding-bottom, padding-left
  - All the padding properties can have the following values:
    - *length* - specifies a padding in px, pt, cm, etc.
    - *%* - specifies a padding in % of the width of the containing element
    - inherit - specifies that the padding should be inherited from the parent element

Padding Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    background-color: lightblue;
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
</style>
</head>
<body><h2>Using individual padding
properties</h2>
<div>This div element has a top
padding of 50px, a right padding
of 30px, a bottom padding of 50px,
and a left padding of 80px.</div>
</body></html>
```



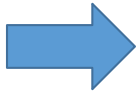


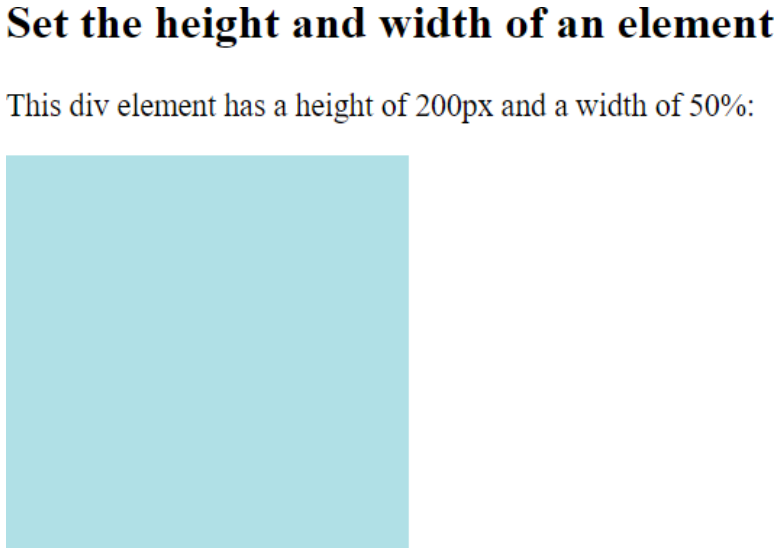
Height & Width

- The height and width properties are used to set the height and width of an element.
- The height and width can be set to auto (this is default. Means that the browser calculates the height and width), or be specified in length values, like px, cm, etc., or in percent (%) of the containing block.
- The height and width properties do not include padding, borders, or margins; they set the height/width of the area inside the padding, border, and margin of the element!
- **Setting max-width**
  - The max-width property is used to set the maximum width of an element.
  - The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

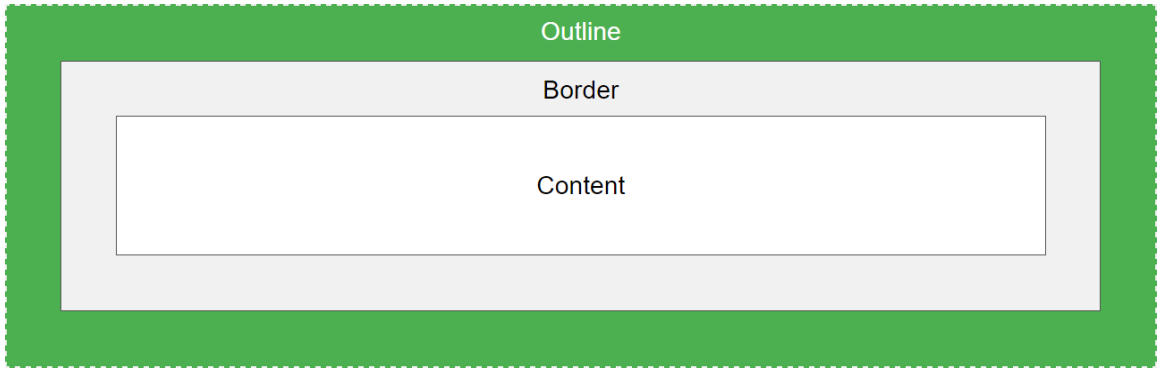
Height & Width - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    height: 200px;
    width: 50%;
    background-color: powderblue;
}
</style>
</head>
<body>
<h2>Set the height and width of an element</h2>
<p>This div element has a height of 200px and a width of 50%:</p>
<div></div>
</body>
</html>
```





- The CSS outline properties specify the style, color, and width of an outline.
- An outline is a line that is drawn around elements (outside the borders) to make the element "stand out".
- the outline property is different from the border property - The outline is NOT a part of an element's dimensions; the element's total width and height is not affected by the width of the outline.

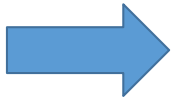


Outline

- The outline-style property specifies the style of the outline.
- The outline-style property can have one of the following values:
  - dotted - Defines a dotted outline
  - dashed - Defines a dashed outline
  - solid - Defines a solid outline
  - double - Defines a double outline
  - groove - Defines a 3D grooved outline. The effect depends on the outline-color value
  - ridge - Defines a 3D ridged outline. The effect depends on the outline-color value
  - inset - Defines a 3D inset outline. The effect depends on the outline-color value
  - outset - Defines a 3D outset outline. The effect depends on the outline-color value
  - none - Defines no outline
  - hidden - Defines a hidden outline

Outline - Example

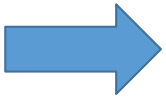
```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    border: 1px solid black;
    outline-color:red;
}
p.dotted {outline-style: dotted;}
</style>
</head>
<body>
<h2>The outline-style
Property</h2>
<p class="dotted">A dotted
outline</p>
</body>
</html>
```



lex
t

- The color property is used to set the color of the text.
- The default text color for a page is defined in the body selector.
- The text-align property is used to set the horizontal alignment of a text.
- A text can be left or right aligned, centred, or justified.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {text-align: center;color: blue;}
h2 {text-align: left;color: red;}
h3 {text-align: right;color: green;}
</style>
</head>
<body>
<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>
<p>aligned center, left and right.</p>
</body>
</html>
```



Heading 1 (center)

Heading 2 (left)

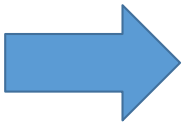
Heading 3 (right)

aligned center, left and right.

Text Decoration

- The text-decoration property is used to set or remove decorations from text.
- The value text-decoration: none; is often used to remove underlines from links

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {text-decoration: overline;}
h2 {text-decoration: line-through;}
h3 {text-decoration: underline;}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
</body>
</html>
```



This is heading 1

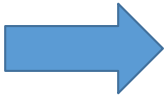
~~**This is heading 2**~~

This is heading 3

Text Transformation

- The text-transform property is used to specify uppercase and lowercase letters in a text.
- It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word

```
<!DOCTYPE html>
<html>
<head>
<style>
p.uppercase {text-transform: uppercase;}
p.lowercase  {text-transform:  lowercase;}
p.capitalize {text-transform:capitalize;}
</style>
</head>
<body>
<p class="uppercase">This is some text.</p>
<p class="lowercase">This is some text.</p>
<p class="capitalize">This is some text.</p>
</body>
</html>
```



THIS IS SOME TEXT.

this is some text.

This Is Some Text.

Text Indentation

- The text-indent property is used to specify the indentation of the first line of a text:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
   text-indent: 50px;
}
</style>
</head>
<body>
<p>In my younger and more vulnerable years
my fathergave me some advice</p>
</body>
</html>
```

In my younger and more vulnerable years my father gave me some advice

Text Letter Spacing

- letter-spacing property is used to specify the space between the characters in a text.
- below example demonstrates how to increase or decrease the space between characters:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {letter-spacing: 3px;}
h2 {letter-spacing:-3px;}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
</body>
</html>
```

This is heading 1

This is heading 2

Text - Line Height

- The line-height property is used to specify the space between lines

```
<!DOCTYPE html>
<html>
<head>
<style>
p.small {line-height:0.7;}
p.big {line-height: 1.8;}
</style>
</head>
<body>
<p>
This is a paragraph with a standard line-height.<br>
The default line height in most browsers is about 110% to 120%.<br>
</p>
<p class="small">
This is a paragraph with a smaller line-height.<br>
This is a paragraph with a smaller line-height.<br>
</p>
<p class="big">
This is a paragraph with a bigger line-height.<br>
This is a paragraph with a bigger line-height.<br>
</p>
</body>
</html>
```

This is a paragraph with a standard line-height.
The default line height in most browsers is about 110% to 120%.

This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.

This is a paragraph with a bigger line-height.

This is a paragraph with a bigger line-height.

Font

- The CSS font properties define the font family, boldness, size, and the style of a text.
- In CSS, there are two types of font family names:
  - generic family - a group of font families with a similar look (like "Serif" or "Monospace")
  - font family - a specific font family (like "Times New Roman" or "Arial")

- The font family of a text is set with the font-family property.
- The font-family property should hold several font names as a "fall back" system. If the browser does not support the first font, it tries the next font, and so on.
- Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.
- If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".
- More than one font family is specified in a comma-separated list:

Font Family - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.serif {font-family: "Times New Roman", Times, serif;}
p.sansserif {font-family: Arial, Helvetica, sans-serif;}
</style>
</head>
<body>
<h1>CSS font-family</h1>
<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>
<p class="sansserif">This is a paragraph, shown in the Arial font.</p>
</body>
</html>
```

CSS font-family

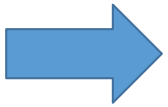
This is a paragraph, shown in the Times New Roman font.

This is a paragraph, shown in the Arial font.

Font Style

- The font-style property is mostly used to specify italic text.
- This property has three values:
  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {font-style: normal;}
p.italic {font-style: italic;}
p.oblique {font-style: oblique;}
</style>
</head>
<body>
<p class="normal">This is a paragraph in normal style.</p>
<p class="italic">This is a paragraph in italic style.</p>
<p class="oblique">This is a paragraph in oblique style.</p>
</body>
</html>
```



This is a paragraph in normal style.

This is a paragraph in italic style.

This is a paragraph in oblique style.

Font Size

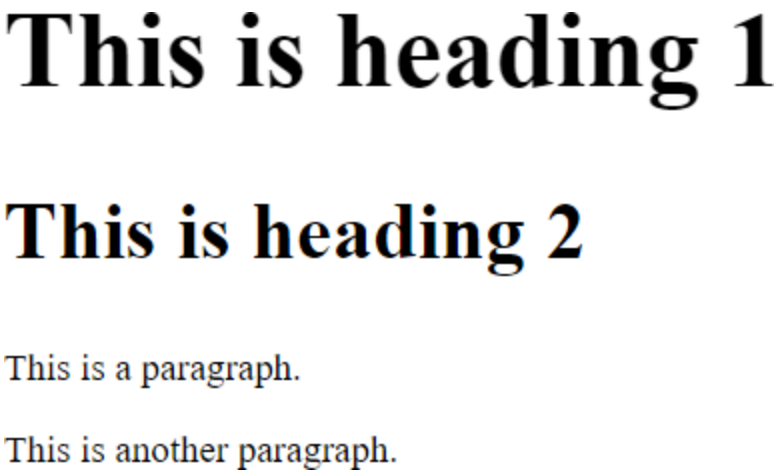
- font-size property sets the size of the text.
- should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.
- use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.
- font-size value can be an absolute, or relative size.
- **Absolute size:**
  - Sets the text to a specified size
  - Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
  - Absolute size is useful when the physical size of the output is known
- **Relative size:**
  - Sets the size relative to surrounding elements
  - Allows a user to change the text size in browsers

Font Size

Pixels

- Setting the text size with pixels gives you full control over the text size

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 { font-size: 40px;}
h2 { font-size: 30px;}
p {font-size: 14px;}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```



Font Size

em

- To allow users to resize the text (in the browser menu), many developers use em instead of pixels.
- The em size unit is recommended by the W3C.
- 1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.
- The size can be calculated from pixels to em using this formula: pixels/16=em

```
h1 {
    font-size: 2.5em; /* 40px/16=2.5em */
}
h2 {
    font-size: 1.875em; /* 30px/16=1.875em */
}
p {
    font-size: 0.875em; /* 14px/16=0.875em */
}
```

Font Weight

- The font-weight property specifies the weight of a font:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {font-weight:normal;}
p.light {font-weight: lighter;}
p.thick{font-weight: bold;}
p.thicker {font-weight:900;}
</style>
</head>
<body>
<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>
</body>
</html>
```

This is a paragraph.

This is a paragraph.

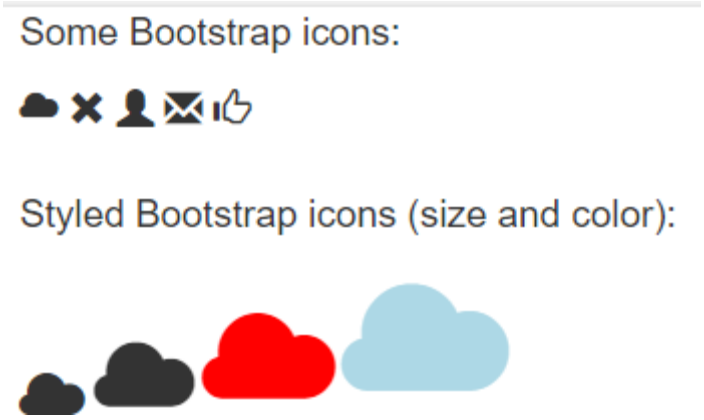
This is a paragraph.

This is a paragraph.

- The simplest way to add an icon to your HTML page, is with an icon library, such as Font Awesome.
- Add the name of the specified icon class to any inline HTML element (like <i> or).
- All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.)
- To use the Bootstrap glyphicons, add the following line inside the <head> section of your HTML page:
- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"

Icons – Bootstrap Example

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap Icons</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body class="container">

<p>Some Bootstrap icons:</p>
<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
<i class="glyphicon glyphicon-user"></i>
<i class="glyphicon glyphicon-envelope"></i>
<i class="glyphicon glyphicon-thumbs-up"></i>
<br><br>

<p>Styled Bootstrap icons (size and color):</p>
<i class="glyphicon glyphicon-cloud" style="font-size:24px;"></i>
<i class="glyphicon glyphicon-cloud" style="font-size:36px;"></i>
<i class="glyphicon glyphicon-cloud" style="font-size:48px;color:red;"></i>
<i class="glyphicon glyphicon-cloud" style="font-size:60px;color:lightblue;"></i>

</body>
</html>
```



Links

- Links can be styled with any CSS property (e.g. color, font-family, background, etc.).
- In addition, links can be styled differently depending on what state they are in.
- The four links states are:
  - a:link - a normal, unvisited link
  - a:visited - a link the user has visited
  - a:hover - a link when the user mouse over it
  - a:active - a link the moment it is clicked

Links - Example

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {color: red;}
a:visited {color: green;}
a:hover {color: hotpink;}
a:active {color: blue;}
</style>
</head>
<body>
<p><b><a href="default.asp"
target="_blank">This is a link</a></b></p>
</body>
</html>
```

This is a link

Links Text Decoration

- The text-decoration property is mostly used to remove underlines from links

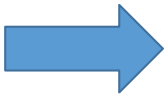
```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {text-decoration: none;}
a:visited {text-decoration: none;}
a:hover {text-decoration: underline;}
a:active {text-decoration:underline;}
</style>
</head>
<body>
<p><b><a href="default.asp"
target="_blank">This is a link</a></b></p>
</body>
</html>
```

This is a link

Links Background Color

- The background-color property can be used to specify a background color for links

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {background-color: yellow;}
a:visited {background-color: cyan;}
a:hover {background-color: lightgreen;}
a:active {background-color: hotpink;}
</style>
</head>
<body>
<p><b><a href="default.asp"
target="_blank">This is a link</a></b></p>
</body>
</html>
```



This is a link

CSS

Overflow

- The CSS overflow property specifies whether to clip content or to add scrollbars when the content of an element is too big to fit in a specified area.
- The overflow property has the following values:
  - visible - Default. The overflow is not clipped. It 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, but a scrollbar is added to see the rest of the content
  - auto - If overflow is clipped, a scrollbar should be added to see the rest of the content

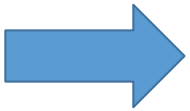
```
div {
    overflow: hidden;
}
```

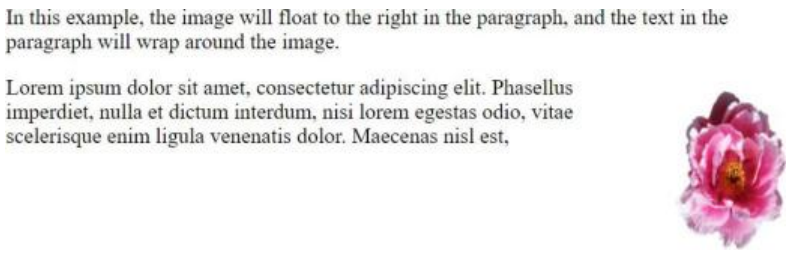
You can use the overflow property when you want to have better control of the

- The float property specifies whether or not an element should float.
- The clear property is used to control the behaviour of floating elements.
- In its simplest use, the float property can be used to wrap text around images.
- The clear property is used to control the behaviour of floating elements.
- Elements after a floating element will flow around it. To avoid this, use the clear property.
- The clear property specifies on which sides of an element floating elements are not allowed to float

Float Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    float: right;
    margin: 0 0 10px 10px;
}
</style>
</head>
<body>
<p>In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.</p>
<p><img src="flower.gif" alt="Lithan.com" width="100" height="140">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, </p>
</body>
</html>
```





- An image sprite is a collection of images put into a single image.
- A web page with many images can take a long time to load and generates multiple server requests.
- Using image sprites will reduce the number of server requests and save bandwidth.
- **Image Sprites - Simple Example**

- Instead of using three separate images, we use this single image ("img_navsprites.gif"):
- With CSS, we can show just the part of the image we need.

Exam
ple

```
<!DOCTYPE html>
<html>
<head>
<style>
#home {
    width: 46px;
    height: 44px;
    background: url(img_navsprites.gif) 0 0;
}

#next {
    width: 43px;
    height: 44px;
    background: url(img_navsprites.gif) -91px 0;
}
</style>
</head>
<body>

<img id="home"><br><br>
<img id="next">

</body>
</html>
```

Styling List in CSS

- **List**
  - Lists are very helpful in conveying a set of numbered or bullet points.
- **We have the following five CSS properties, which can be used to control lists:**
  - **list-style-type :** Used to control the shape or appearance of the marker.
  - **list-style-position :** Used to specify whether a long point that wraps to a second line should align with the first line or start underneath the start of the marker.
  - **list-style-image:** Used to specify an image for the marker rather than a bullet point or number.
  - **list-style:** Used to serve as shorthand for the preceding properties.
  - **marker-offset:** Used to specify the distance between a marker and the text in the list.

Styling List in CSS – list-style-type

```
<html>
  <head>
  </head>
  <body>
    <ul style="list-style-type:circle;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li> </ul>
    <ul style="list-style-type:square;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li></ul>
    <ol style="list-style-type:decimal;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li></ol>
    <ol style="list-style-type:lower-alpha;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li></ol>
    <ol style="list-style-type:lower-roman;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li></ol>
  </body>
</html>
```

o/p

- Maths
- Social Science
- Physics

- Maths
- Social Science
- Physics

1. Maths
2. Social Science
3. Physics

a. Maths
b. Social Science
c. Physics

i. Maths
ii. Social Science
iii. Physics

Styling List in CSS – list-style-position

```
<html>
  <head>
  </head>
  <body>
    <ul style="list-style-type:circle; list-style-position:outside;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>
    <ul style="list-style-type:square;list-style-position:inside;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>
    <ol style="list-style-type:decimal;list-style-position:outside;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
    <ol style="list-style-type:lower-alpha;list-style-position:inside;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
  </body>
</html>
```

- Maths
- Social Science
- Physics

- Maths
- Social Science
- Physics

1. Maths
2. Social Science
3. Physics

a. Maths
b. Social Science
c. Physics

Styling List in CSS - list-style-image

Example

```
<html>
  <head>
  </head>

  <body>
    <ul>
      <li style="list-style-image: url(/images/bullet.gif);">Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ol>
      <li style="list-style-image: url(/images/bullet.gif);">Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
  </body>

</html>
```

Result

- Maths
- Social Science
- Physics

1. Maths
2. Social Science
3. Physics

Styling List in CSS - list-style

❑ The list-style Property

- The *list-style* used to specify all the list properties into a single expression.

Example

```
<html>
  <head>
  </head>

  <body>
    <ul style="list-style: inside square;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ol style="list-style: outside upper-alpha;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
  </body>

</html>
```

Result

- Maths
- Social Science
- Physics

A. Maths
B. Social Science
C. Physics

Styling List in CSS - marker-offset

❑ The marker-offset Property

- The *marker-offset* property used to specify the distance between the marker and the text relating to that marker.

Example

```
<html>
  <head>
  </head>

  <body>
    <ul style="list-style: inside square; marker-offset:2em;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ol style="list-style: outsideupper-alpha; marker-offset:2cm;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
  </body>

</html>
```

Result

- Maths
- Social Science
- Physics

A. Maths
B. Social Science
C. Physics

Styling Tables in CSS

- **Styling Tables**

To set different properties of an HTML table using CSS you can set following properties of a table −

- **border-collapse:** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
- **border-spacing:** specifies the width that should appear between table cells.
- **caption-side**: captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
- **empty-cells** :specifies whether the border should be shown if a cell is empty.

Styling Tables in CSS

- **table-layout** :allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

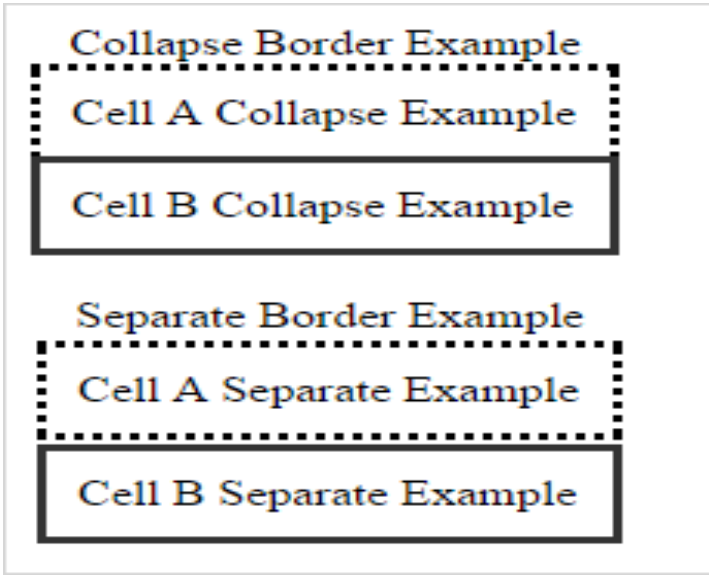
❑ The border-collapse Property:

- This property can have two values *collapse* and *separate*. The following example uses both the values:

Styling Tables in CSS

```
<html>
  <head>
    <style type="text/css">
      table.one   {border-collapse:collapse;}
      table.two  {border-collapse:separate;}
      td.a {
        border-style:dotted;
        border-width:3px;
        border-color:#000000;
        padding:  10px;
      }
      td.b {
        border-style:solid;
        border-width:3px;
        border-color:#333333;
        padding:10px;
      }
    </style>
  </head>
  <body>
    <table class="one">        <caption>Collapse Border  Example</caption>
      <tr><td class="a"> Cell A Collapse Example</td></tr>
      <tr><td class="b"> Cell B Collapse Example</td></tr></table><br />
    <table class="two">        <caption>Separate Border Example</caption>
      <tr><td class="a"> Cell A Separate Example</td></tr>
      <tr><td class="b"> Cell B Separate Example</td></tr> </table>
    </body>
</html>
```

o/p



Styling Tables in CSS

- **The border-spacing Property**
  - specifies the distance that separates adjacent cells' borders.
  - can take one or two values; these should be units of length.
  - provide one value, it will applies to both vertical and horizontal borders.
  - can specify two values, in which case, the first refers to the horizontal spacing and the second to the vertical spacing

Exam ple

```
<head>
  <style type="text/css">
    table.one {
      border-collapse:separate;
      width:400px;
      border-spacing:10px;
    }
    table.two {
      border-collapse:separate;
      width:400px;
      border-spacing:10px 50px;
    }
  </style>
</head>
<body>
  <table class="one" border="1">
    <caption>Separate Border Example with border-spacing</caption>
    <tr><td> Cell A Collapse Example</td></tr>
    <tr><td> Cell B Collapse Example</td></tr>
  </table>
  <br />
  <table class="two" border="1">
    <caption>Separate Border Example with border-spacing</caption>
    <tr><td> Cell A Separate Example</td></tr>
    <tr><td> Cell B Separate Example</td></tr> </table>
</body>
<
```

o/p

Separate Border Example with border-spacing

Cell A Collapse Example
Cell B Collapse Example

Separate Border Example with border-spacing

Cell A Separate Example
Cell B Separate Example

Styling Tables in CSS

- **The caption-side Property**
  - The caption-side property used to specify where the content of a <caption> element should be placed in relationship to the table. The table that follows lists the possible values.
  - This property can have one of the four values *top, bottom, right or left*.

Exam ple

```
<style type="text/css">
      caption.top {caption-side:top}
      caption.bottom {caption-side:bottom}
      caption.left {caption-side:left}
      caption.right {caption-side:right}</style></head>
 <body>
   <table style="width:400px; border:1px solid black;">
    <caption class="top">
      This caption will appear at the top</caption>
      <tr><td > Cell A</td></tr>
      <tr><td > Cell B</td></tr></table><br />
   <table style="width:400px; border:1px solid black;">
      <caption class="bottom">
      This caption will appear at the bottom </caption>
      <tr><td > Cell A</td></tr>
      <tr><td > Cell B</td></tr> </table> <br/>
   <table style="width:400px; border:1px solid black;">
      <caption class="left">
      This caption will appear at the left </caption>
      <tr><td > Cell A</td></tr>
      <tr><td > Cell B</td></tr></table><br />
   <table style="width:400px; border:1px solid black;">
      <caption class="right">
      This caption will appear at the right
      </caption>
      <tr><td > Cell A</td></tr>
      <tr><td > Cell B</td></tr></table></body>
```

o/p

This caption will appear at the top

Cell A
Cell B

Cell A
Cell B

This caption will appear at the bottom

This caption will appear at the left

Cell A
Cell B

This caption will appear at the right

Cell A
Cell B

Styling Tables in CSS

- **The empty-cells Property**
  - The empty-cells property indicates whether a cell without any content should have a border displayed.
  - This property can have one of the three values - *show, hide* or *inherit*.
  - Here is the empty-cells property used to hide borders of empty cells in the <table> element..

Exam ple

```
<style type="text/css">
      table.empty{
        width:350px;
        border-collapse:separate;
        empty-cells:hide;
      }
      td.empty{
        padding:5px;
        border-style:solid;
        border-width:1px;
        border-color:#999999;
      }
    </style>
  </head>
  <body>
    <table class="empty">
    <tr>
      <th></th>
      <th>Title one</th>
      <th>Title two</th></tr>
    <tr>
      <th>Row Title</th>
      <td class="empty">value</td>
      <td class="empty">value</td></tr>
    <tr>
      <th>Row Title</th>
      <td class="empty">value</td>
      <td class="empty"></td></tr></table>
```

	Title one	Title two
Row Title	value	value
Row Title	value	

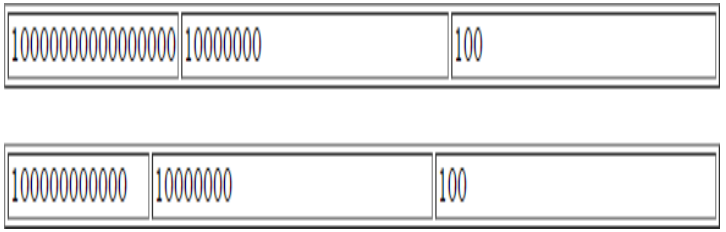
Styling Tables in CSS

- **The table-layout Property**
  - The table-layout property helps you control how a browser should render or layout a table.
  - This property can have one of the three values: *fixed, auto* or *inherit*.

Exam ple

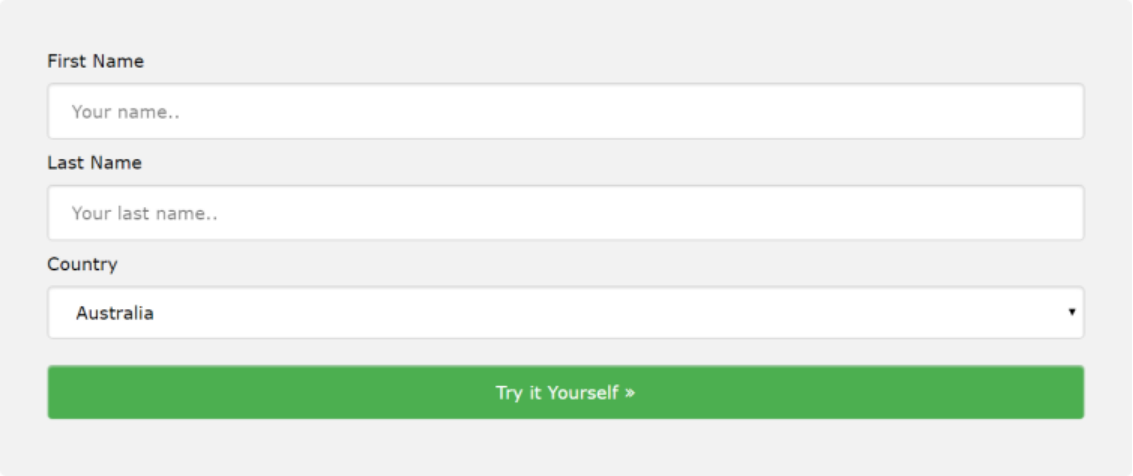
```
<html>
  <head>
    <style type="text/css">
      table.auto {
        table-layout: auto
      }
      table.fixed{
        table-layout: fixed
      }
    </style>
  </head>
  <body>
    <table class="auto" border="1"width="100%">
    <tr>
      <td width="20%">10000000000000000000000000000</td>
      <td width="40%">10000000</td>
      <td width="40%">100</td>
    </tr>
    </table><br />
    <table class="fixed" border="1"width="100%">
    <tr>
      <td width="20%">10000000000000000000000000000</td>
      <td width="40%">10000000</td>
      <td width="40%">100</td>
    </tr> </table>
  </body>
</html>
```

o/p



For ms

- The look of an HTML form can be greatly improved with CSS.



- The example applies to all <input> elements. If you only want to style a specific input type, you can use attribute selectors:
  - input[type=text] - will only select text fields
  - input[type=password] - will only select password fields
  - input[type=number] - will only select number fields

Styling Input Fields

- Use the width property to determine the width of the input field:

```
<!DOCTYPE html>
<html>
<head>
<style>
input {
   width: 100%;
}
</style>
</head>
<body>

<p>A full-width input field:</p>

<form>
  <label for="fname">First Name</label>
  <input type="text" id="fname" name="fname">
</form>

</body>
</html>
```

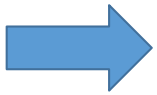
A full-width input field:

First Name

Padded Inputs

- Use the padding property to add space inside the text field.
- When you have many inputs after each other, you might also want to add some margin, to add more space outside of them:

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
}
</style>
</head>
<body>

<p>Padded text fields:</p>

<form>
 <label for="fname">First Name</label>
 <input type="text" id="fname" name="fname">
 <label for="lname">Last Name</label>
 <input type="text" id="lname" name="lname">
</form>

</body>
</html>
```



Bordered Inputs

- Use the border property to change the border size and color, and use the border-radius property to add rounded corners

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
   width: 100%;
   padding: 12px 20px;
   margin: 8px 0;
   box-sizing: border-box;
   border: 2px solid red;
   border-radius: 4px;
}
</style>
</head>
<body>
<p>Text fields with borders:</p>
<form>
  <label for="fname">First Name</label>
  <input type="text" id="fname" name="fname">
  <label for="lname">Last Name</label>
  <input type="text" id="lname" name="lname">
</form>
</body>
</html>
```

Text fields with borders:

First Name

Last Name

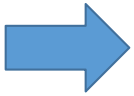
```
input[type=text] {
    border: none;
    border-bottom: 2px solid red;
}
```

- If you only want a bottom border, use the border-bottom property

Colored Inputs

- Use the background-color property to add a background color to the input, and the color property to change the text color:

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: none;
  background-color: #3CBC8D;
  color: white;
}
</style>
</head>
<body>
<p>Colored text fields:</p>
<form>
 <label for="fname">First Name</label>
 <input type="text" id="fname" name="fname" value="John">
 <label for="lname">Last Name</label>
 <input type="text" id="lname" name="lname" value="Doe">
</form>
</body>
</html>
```



Colored text fields:

First Name

John

Last Name

Doe

Focused Inputs

- By default, some browsers will add a blue outline around the input when it gets focus (clicked on).
- You can remove this behavior by adding outline: none; to the input.
- Use the :focus selector to do something with the input field when it gets focus:

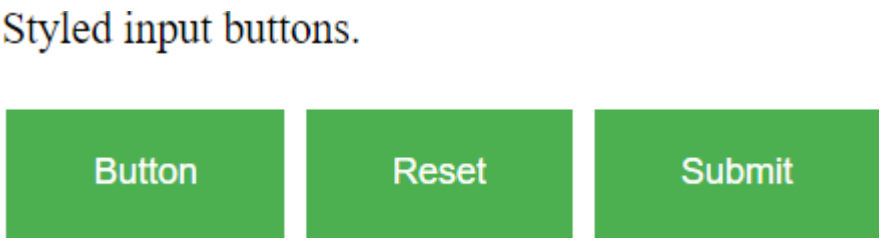
```
input[type=text]:focus {
    background-color: lightblue;
}
```

```
input[type=text]:focus {
    border: 3px solid #555;
}
```

Styling Input Buttons

- You can style the input buttons using CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=button], input[type=submit],
input[type=reset] {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 16px 32px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
}
</style>
</head>
<body>
<p>Styled input buttons.</p>
<input type="button" value="Button">
<input type="reset" value="Reset">
<input type="submit" value="Submit">
</body>
</html>
```



CSS Import

- The @import statement lets us import CSS from other style sheets. It shares some of the same advantages as linking a stylesheet, like browser caching and maintenance efficiency.

```
@import 'reset-styles.css';
```

CSS

Comments

- The stuff inside the /* */ marks are CSS comments.
- This allows you to enter notes into CSS that will not be interpreted.
- In this case, this comment lets someone reading the CSS file know that that particular line of CSS was intended to allow for using ems to set font size later in the CSS in a more intuitive base 10 way.

```
body  font-size 62.5%/* 1em = 10px*/
```

THANK YOU