بسم الله الرحمن الرحيم

تقنيات و أكواد سي اس اس

اللهم أعني على إتمام هذا الكتاب

الحمد لله المنفرد بالخلق والتدبير، الواحد في الحكم والتقدير، الملك الذي ليس له في ملكه وزير.

وأشهد ألا إله الا الله وحده لا شريك له شهادة عبد مفوض لقضائه، مستسلم في حكمه وامضائه

وأشهد أن محمد عبده ورسوله، المفضل على جميع أنبيائه، المخصوص بجزيل فضله وعطائه، الفاتح الخاتم وليس ذلك لسوائه، الشافع في كل العباد حين يجمعهم الحق لفصل قضائه.

صلى الله عليه وعلى آله وصحبه المستمسكين بولائه وسلم كثيرا.

أما بعد:

بفضل الله ، جمعت في هذا الكتاب أساسيات البرمجة بسي اس اس ، حيث أغلب الأكواد و التقنيات نقلتها من الموقع الشهير w3school ، حيث يمكنك العودة اليه في حال أردت تجربة الأكواد بشكل أسهل ، و كذا الاطلاع على بعض التفاصيل التي لم أذكرها بهذا الكتاب .

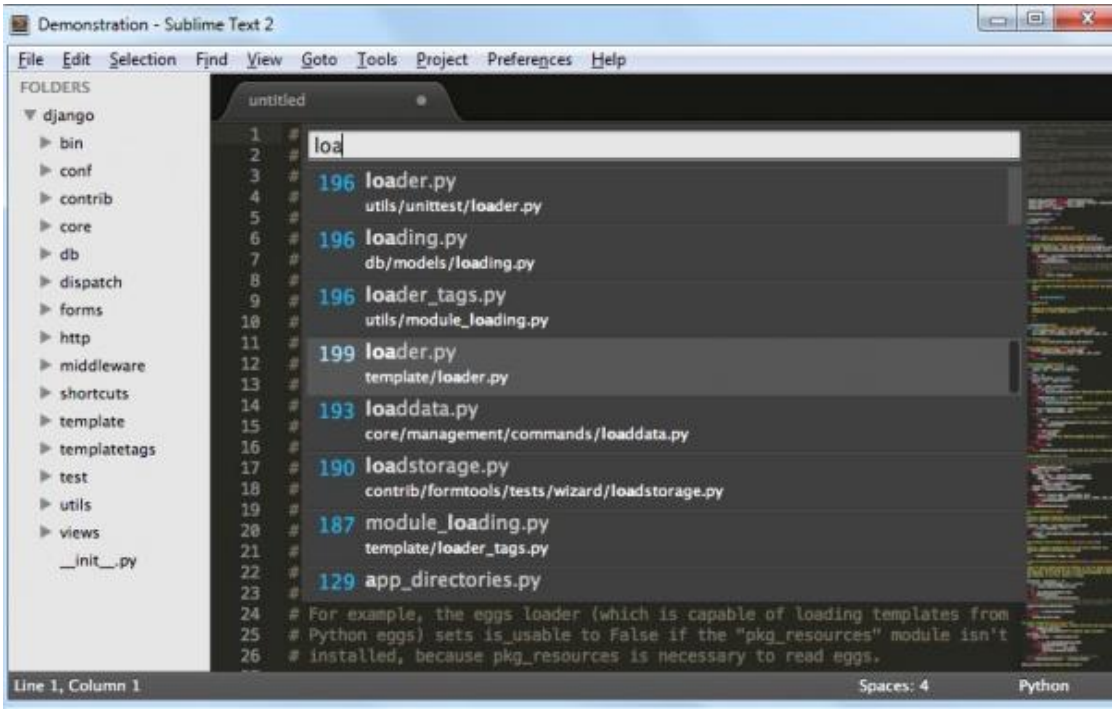
أريد أن أذكر يا صديقي أن الجزء الثاني من الكتاب لن يكون كالأول فسيشمل أمثلة وكذا مواقع لتحميل قوالب جاهزة واحترافية ان شاء الله .

فأسأل الله أن يفيدك الكتاب فهو يساعدك في الاطلاع على الأكواد دون نت عكس الموقع،

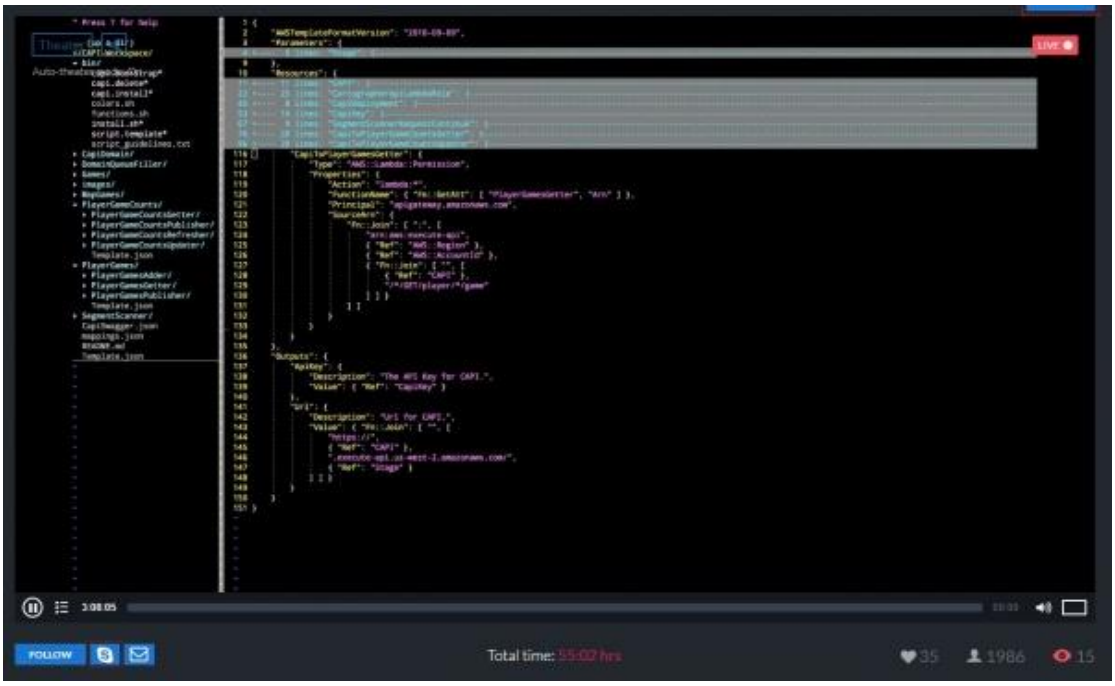
وأسأل الله قبل كل شيء أن يكون لوجهه الكريم، ولا حول وقوة الا بالله.

قائمة لبعض من المحررات الاحترافية

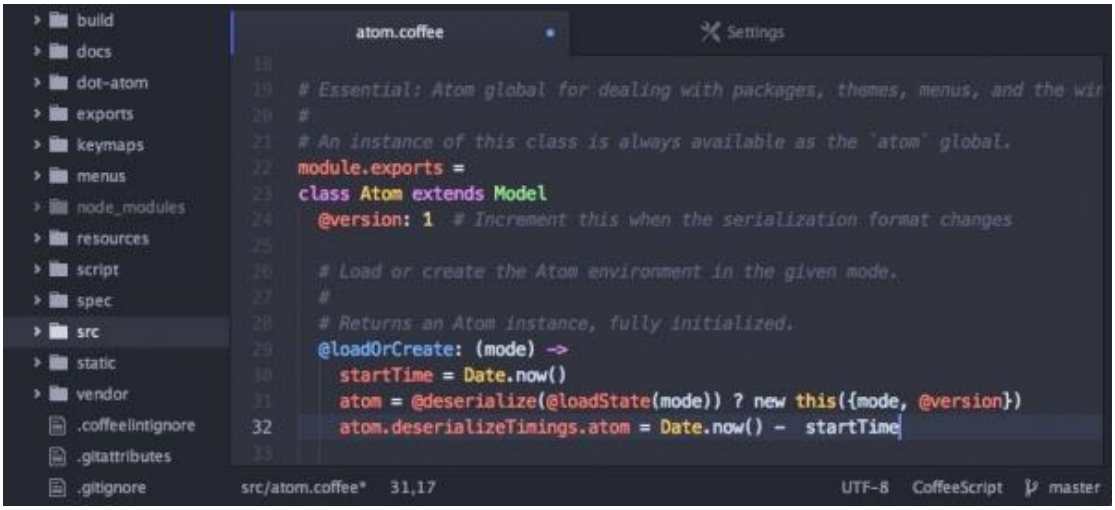
SUBLIME TEXT EDITOR



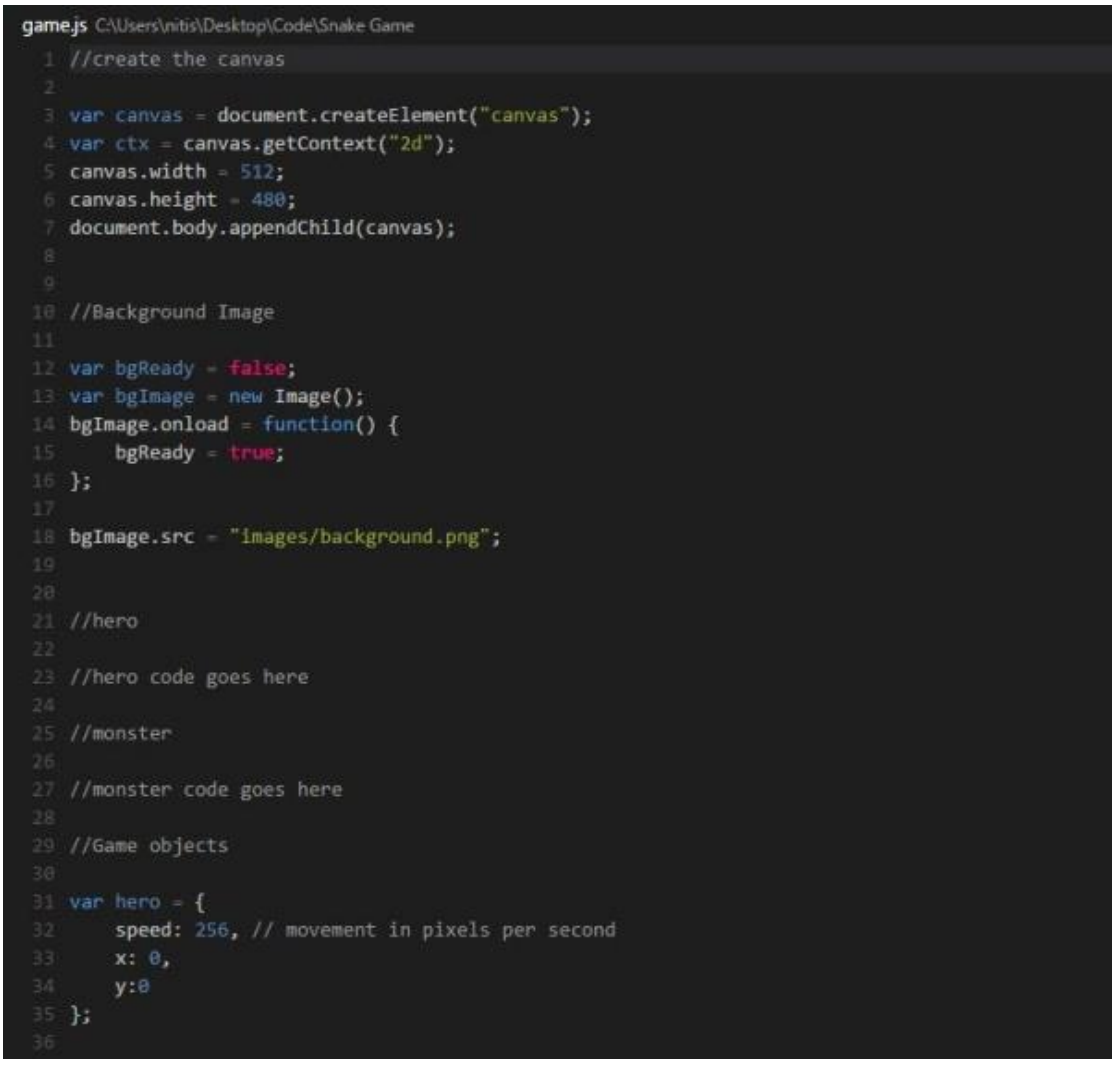
VIM



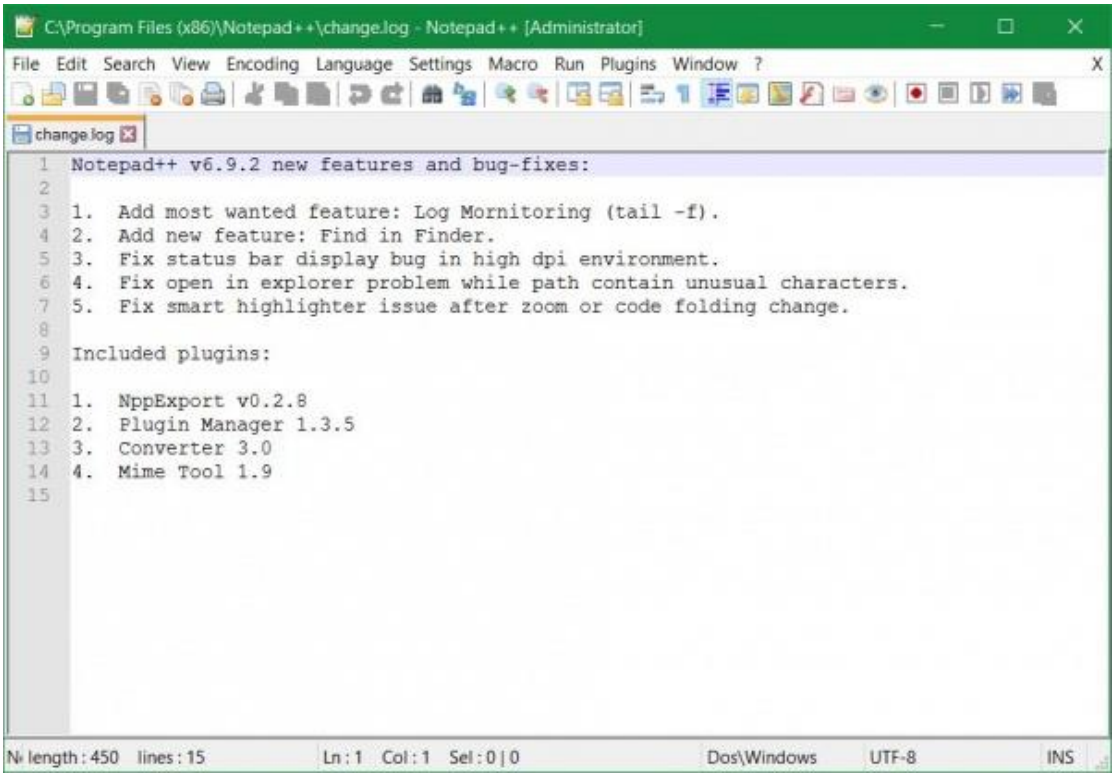
ATOM



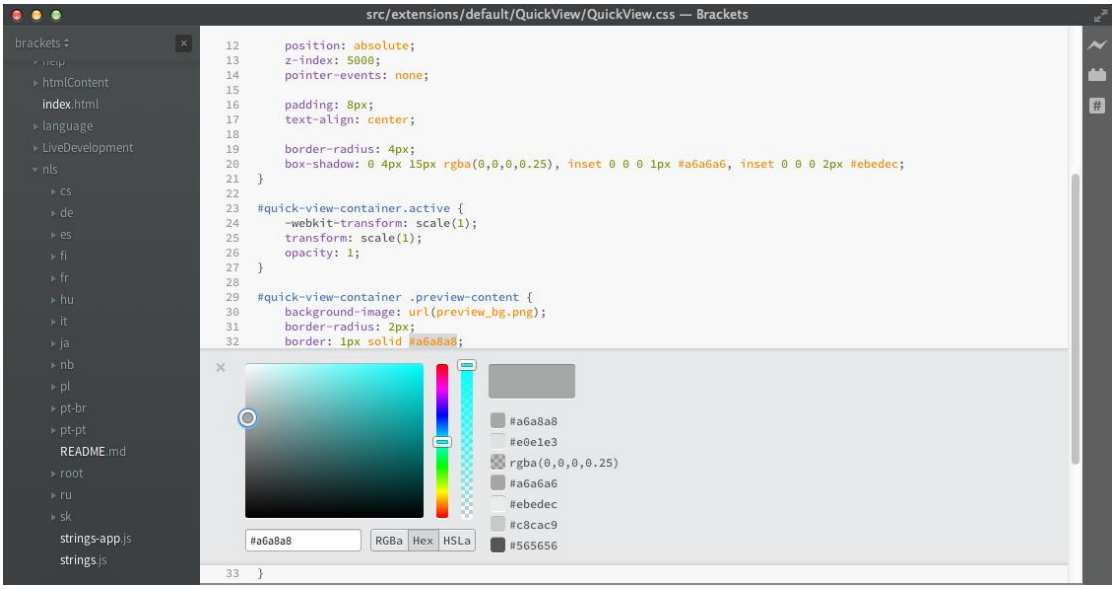
VISUAL STUDIO CODE



NOTEPAD++



BRACKETS



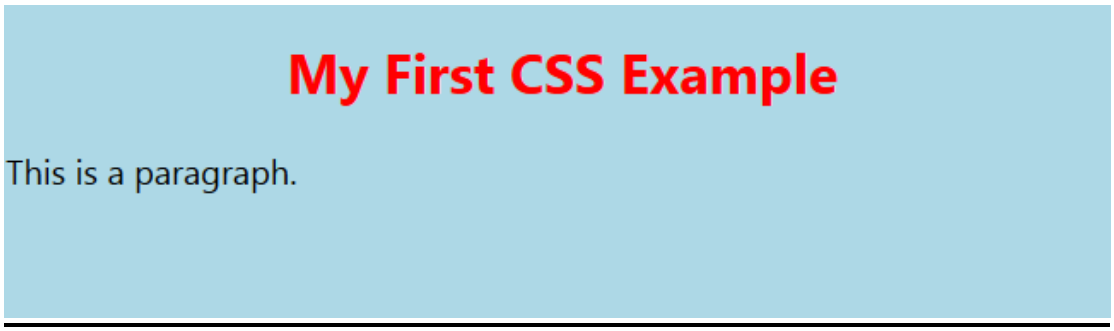
المصدر

/https://blog.liveedu.tv/10-best-text-editors-programming-2016

أكود سي اس اس

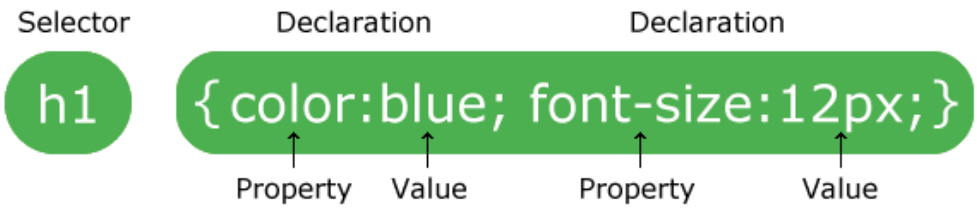
Syntax and Selectors

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}

h1 {
    color: red;
    text-align: center;
}

p {
    font-family: aria;
    font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



CSS SYNTAX

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

In the following example all <p> elements will be center-aligned, with a red text color:

EXTERNAL STYLE SHEET

External Style Sheet

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:

Example

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

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;
}
```

Note: Do not add a space between the property value and the unit (such as `margin-left: 20 px;`). The correct way is: `margin-left: 20px;`

Inline Styles

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a <h1> element:

Example

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

CSS COLORS

Color Names

In HTML, a color can be specified by using a color name:

Tomato	Orange	DodgerBlue	MediumSeaGreen
Gray	SlateBlue	Violet	LightGray

BACKGROUND COLOR

You can set the background color for HTML elements:

Hello World

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.

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

وَقُل رَّبِّ زِدْنِي عِلْمًا

TEXT COLOR

You can set the color of text:

Hello World

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.

Example

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

BORDER COLOR

You can set the color of borders:

Hello World

Hello World

Hello World

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

COLOR VALUES

rgb(255, 99, 71)

#ff6347

hsl(9, 100%, 64%)

Same as color name "Tomato", but 50% transparent:

rgba(255, 99, 71, 0.5)

hsla(9, 100%, 64%, 0.5)

Example

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>
```

RGBA VALUE

RGBA color values are an extension of RGB color values with an alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with:

rgba(*red, green, blue, alpha*)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example

rgba(255, 99, 71, 0)

rgba(255, 99, 71, 0.2)

rgba(255, 99, 71, 0.4)

rgba(255, 99, 71, 0.6)

rgba(255, 99, 71, 0.8)

rgba(255, 99, 71, 1)

HSLA VALUE

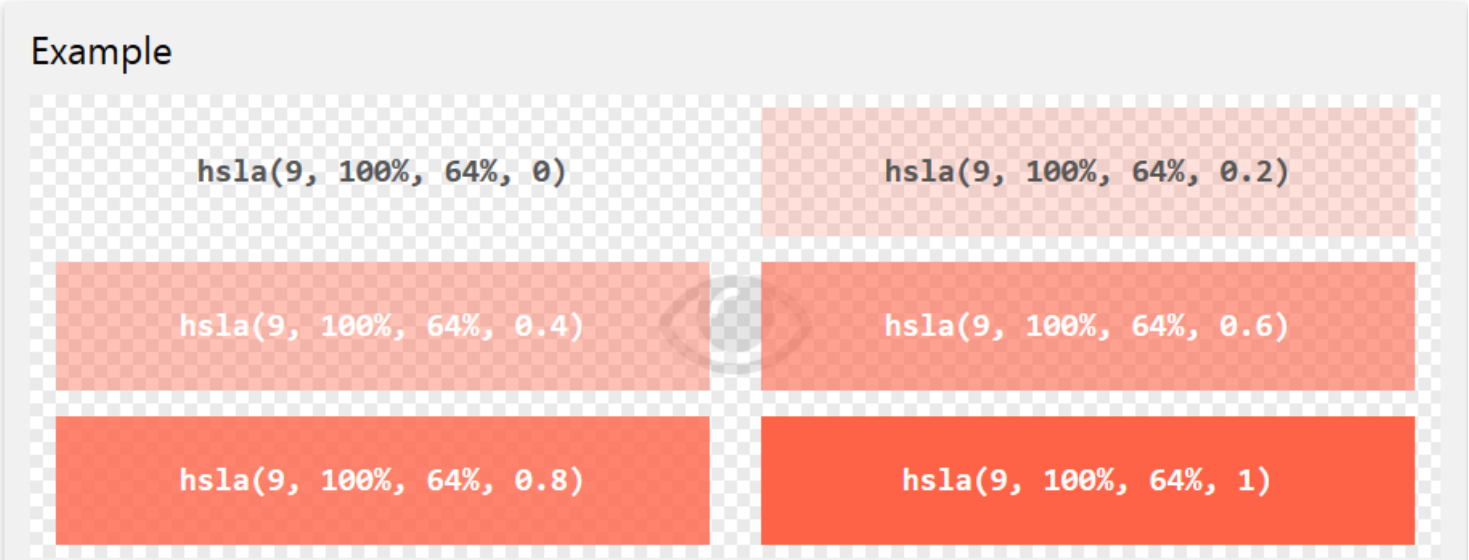
HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

An HSLA color value is specified with:

hsla(*hue, saturation, lightness, alpha*)

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

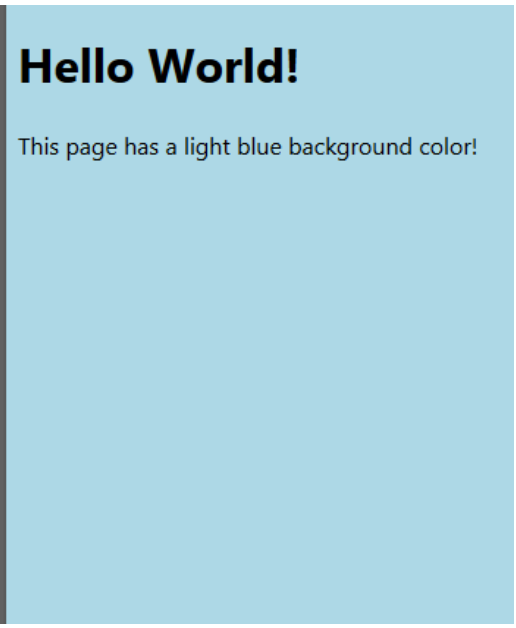
Example



CSS BACKGROUNDS

BACKGROUND COLOR

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}
</style>
</head>
<body>

<h1>Hello World!</h1>

<p>This page has a light blue background color!</p>

</body>
</html>
```



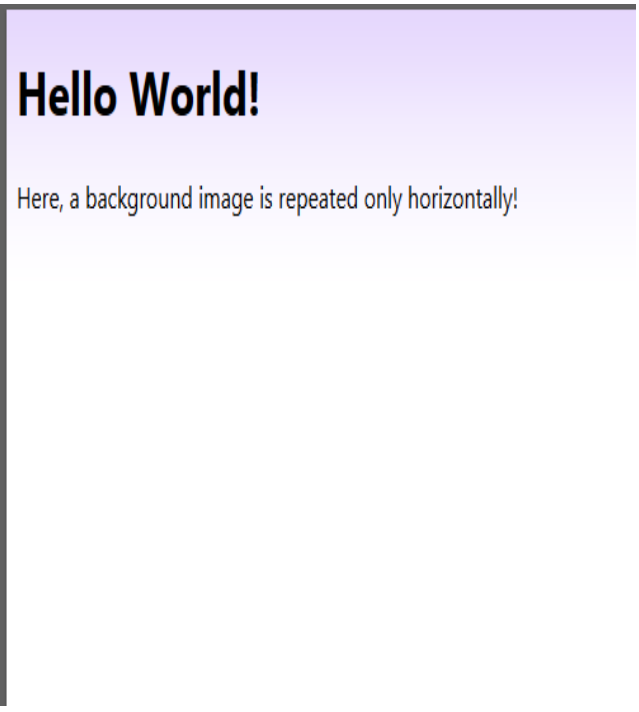
BACKGROUND IMAGE

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

<h1>Hello World!</h1>
```

Note: When using a background image, use an image that does not disturb the text.

BACKGROUND IMAGE - REPEAT HORIZONTALLY OR VERTICALLY

```
<!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 background image is repeated only horizontally!</p>

</body>
</html>
```



Tip: To repeat an image vertically, set `background-repeat: repeat-y;`

background-image: url("img/gradientbg.png");

background-repeat: repeat-x;

(GRADIENT – IMAGE)

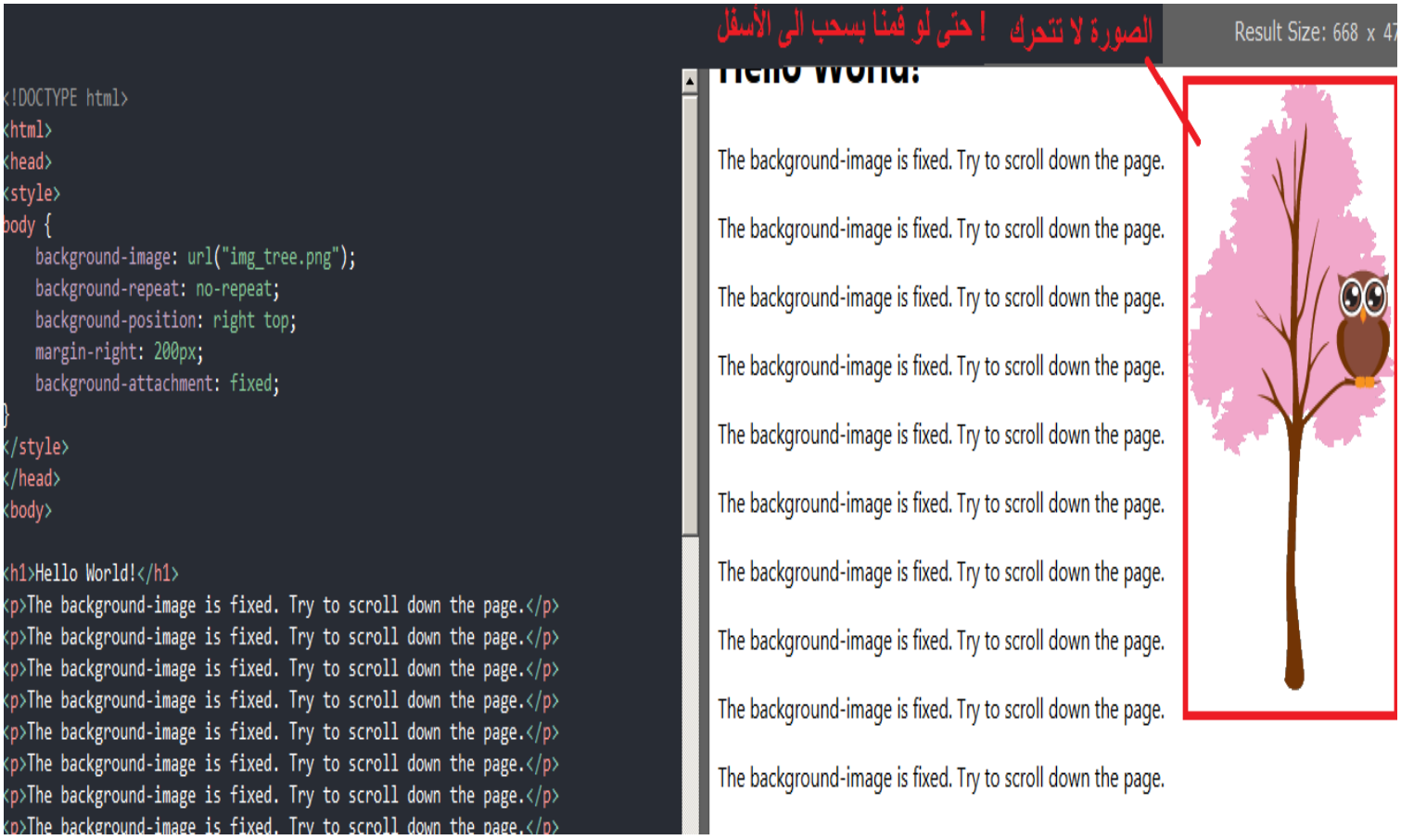
GRADIENTBACKGROUND IMAGE - SET POSITION AND NO-REPEAT

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    margin-right: 200px; /* أستغفر الله العظيم و أتوب اليه */
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>W3Schools background no-repeat, set position example.</p>
<p>Now the background image is only shown once, and positioned away from the
text.</p>
<p>In this example we have also added a margin on the right side, so the
background image will never disturb the text.</p>

</body>
</html>
```

background-repeat: no-repeat;
background-position: right top;

BACKGROUND IMAGE - FIXED POSITION

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
    background-position: right top;
    margin-right: 200px;
    background-attachment: fixed;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<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>
<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>
```



background-attachment: fixed;

BACKGROUND - SHORTHAND PROPERTY

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background: #ff2200 url("img_tree.png") 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>

</body>
</html>
```

When using the shorthand property the order of the property values is:

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`

background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-clip	Specifies the painting area of the background
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-origin	Specifies where the background image(s) is/are positioned
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated
background-size	Specifies the size of the background image(s)

CSS BORDERS

CSS BORDER PROPERTIES

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

I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

BORDER STYLE

The `border-style` property specifies what kind of border to display.

The following values are allowed:

- `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
- `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

The `border-style` property can have from one to four values (for the top border, right border, bottom border, and the left border).

وَقُل رَّبِّ زِدْنِي عِلْمًا

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;}
```

Result:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

Note: None of the OTHER CSS border properties described below will have ANY effect unless the `border-style` property is set!

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).

Example

```
p.one {
    border-style: solid;
    border-width: 5px;
}
```

BORDER COLOR

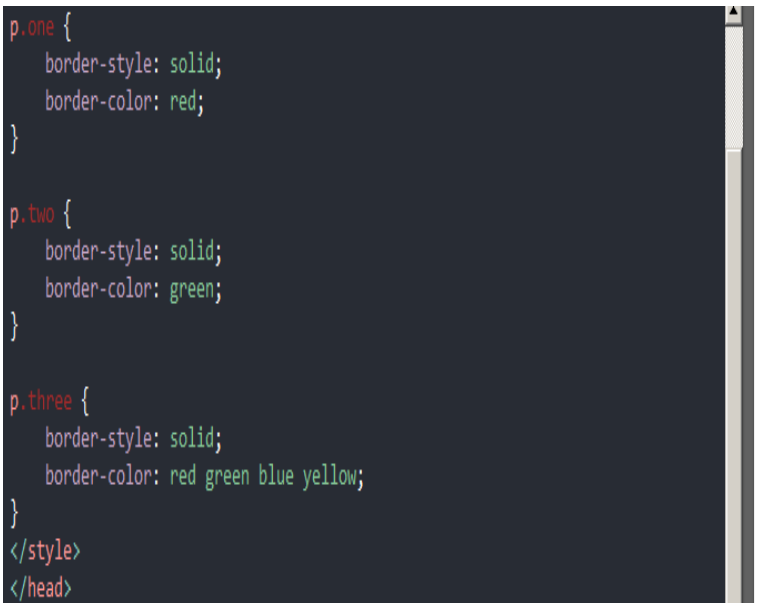
The `border-color` property is used to set the color of the four borders.

The color can be set by:

- name - specify a color name, like "red"
- Hex - specify a hex value, like "#ff0000"
- RGB - specify a RGB value, like "rgb(255,0,0)"
- transparent

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.



```
p.one {
    border-style: solid;
    border-color: red;
}

p.two {
    border-style: solid;
    border-color: green;
}

p.three {
    border-style: solid;
    border-color: red green blue yellow;
}
</style>
</head>
```

The border-color Property

This property specifies the color of the four borders:

A solid red border

A solid green border

A solid multicolor border

Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.

BORDER - INDIVIDUAL SIDES

From the examples above you have seen that it is possible to specify a different border for each side.

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

Different Border Styles

Example

```
p {
    border-top-style: dotted;
    border-right-style: solid;
    border-bottom-style: dotted;
    border-left-style: solid;
}
```

The example above gives the same result as this:

Example

```
p {
    border-style: dotted solid;
}
```

If the `border-style` property has four values:

- **border-style: dotted solid double dashed;**
  - top border is dotted
  - right border is solid

- bottom border is double
- left border is dashed

If the `border-style` property has three values:

- **border-style: dotted solid double;**
  - top border is dotted
  - right and left borders are solid
  - bottom border is double

If the `border-style` property has two values:

- **border-style: dotted solid;**
  - top and bottom borders are dotted
  - right and left borders are solid

If the `border-style` property has one value:

- **border-style: dotted;**
  - all four borders are dotted

The `border-style` property is used in the example above. However, it also works with `border-width` and `border-color`.

BORDER - SHORTHAND PROPERTY

As you can see from the examples above, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property.

The `border` property is a shorthand property for the following individual border properties:

- `border-width`
- `border-style` (required)
- `border-color`

Example

```
p {
    border: 5px solid red;
}
```

Result:

Some text

You can also specify all the individual border properties for just one side:

Left Border

```
p {
    border-left: 6px solid red;
    background-color: lightgrey;
}
```

Result:

Some text

وَقُل رَّبِّ زِدْنِي عِلْمًا

Bottom Border

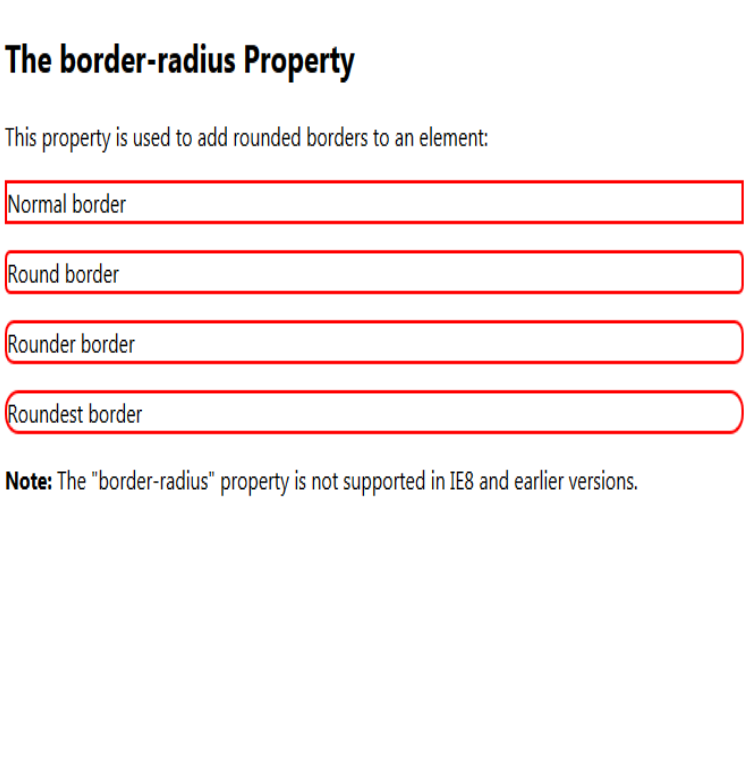
```
p {
    border-bottom: 6px solid red;
    background-color: lightgrey;
}
```

Result:

Some text

ROUNDED BORDERS

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
    border: 2px solid red;
}

p.round1 {
    border: 2px solid red;
    border-radius: 5px;
}

p.round2 {
    border: 2px solid red;
    border-radius: 8px;
}

p.round3 {
    border: 2px solid red;
    border-radius: 12px;
}
</style>
</head>
<body>
```



Note: The `border-radius` property is not supported in IE8 and earlier versions

ALL CSS BORDER PROPERTIES

Property	Description
border	Sets all the border properties in one declaration
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-color	Sets the color of the four borders
border-left	Sets all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border

border-radius	Sets all the four border-*-radius properties for rounded corners
border-right	Sets all the right border properties in one declaration
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-style	Sets the style of the four borders
border-top	Sets all the top border properties in one declaration
border-top-color	Sets the color of the top border
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders

CSS MARGINS

CSS MARGINS

The CSS `margin` properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are 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:

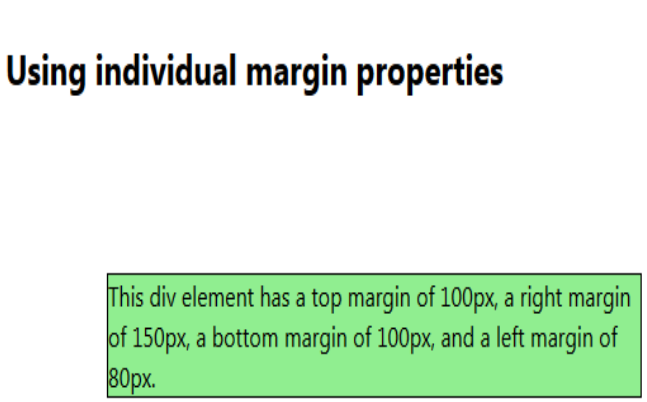
- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

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

Tip: Negative values are allowed.

```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
    background-color: lightgreen;
}
</style>
</head>
<body>

<h2>Using individual margin properties</h2>

<div>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.</div>

</body>
</html>
```



MARGIN - SHORTHAND PROPERTY

To shorten the code, it is possible to specify all the margin properties in one property.

The `margin` property is a shorthand property for the following individual margin properties:

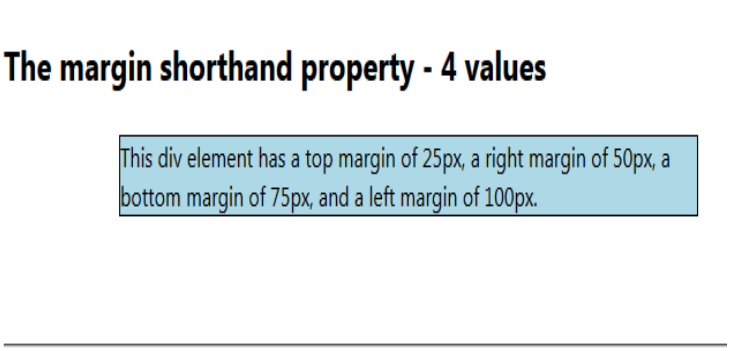
- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

So, here is how it works:

If the `margin` property has four values:

- **margin: 25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px

```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    margin: 25px 50px 75px 100px;
    background-color: lightblue;
}
</style>
</head>
<body>

<h2>The margin shorthand property - 4 values</h2>

<div>This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.</div>

<hr>

</body>
</html>
```



If the `margin` property has two values:

- **margin: 25px 50px;**
  - top and bottom margins are 25px
  - right and left margins are 50px

If the `margin` property has one value:

- **margin: 25px;**
  - all four margins are 25px

THE AUTO VALUE

You can set the margin property to `auto` to horizontally center the element within its container.

The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:

```html
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width:300px;
    margin: auto;
    border: 1px solid red;
}
</style>
</head>
<body>

<h2>Use of margin:auto</h2>
<p>You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:</p>

<div>
This div will be horizontally centered because it has margin: auto;
</div>

</body>
</html>
```

Use of margin:auto

You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins:

This div will be horizontally centered because it has margin: auto;

THE INHERIT VALUE

```
p.ex1 {
margin-left: inherit;
}
```

MARGIN COLLAPSE

```html
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    margin: 0 0 50px 0;
}

h2 {
    margin: 20px 0 0 0;
}
</style>
</head>
<body>

<p>In this example the h1 element has a bottom margin of 50px and the h2 element has a top margin of 20px. Then, the vertical margin between h1 and h2 should have been 70px (50px + 20px). However, due to margin collapse, the actual margin ends up being 50px.</p>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

</body>
</html>
```

In this example the h1 element has a bottom margin of 50px and the h2 element has a top margin of 20px. Then, the vertical margin between h1 and h2 should have been 70px (50px + 20px). However, due to margin collapse, the actual margin ends up being 50px.

Heading 1

Heading 2

CSS PADDING

CSS PADDING

The CSS `padding` properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are 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 of an element:

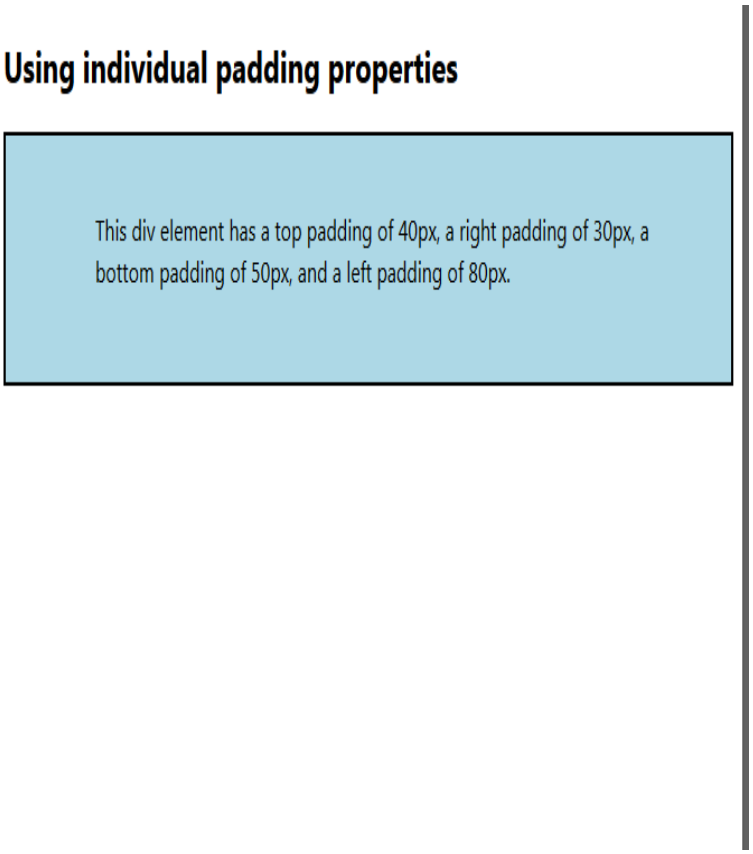
- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

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

Note: Negative values are not allowed.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 2px solid black;
    background-color: lightblue;
    padding-top: 40px;
    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 40px, a right padding of 30px, a
bottom padding of 50px, and a left padding of 80px.</div>

</body>
</html>
```



PADDING - SHORTHAND PROPERTY

To shorten the code, it is possible to specify all the padding properties in one property.

The `padding` property is a shorthand property for the following individual padding properties:

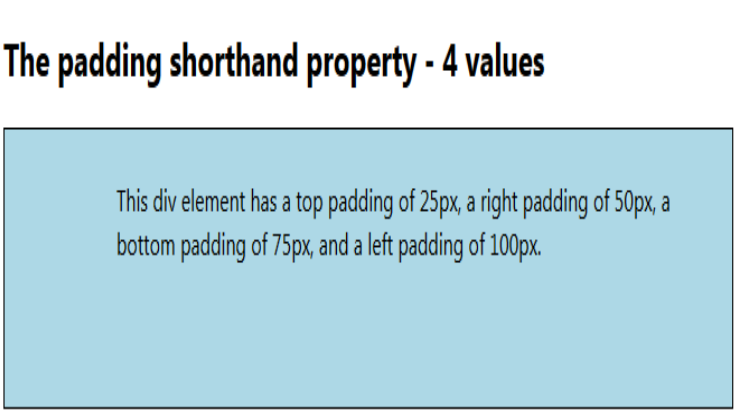
- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

So, here is how it works:

If the `padding` property has four values:

- **padding: 25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    padding: 25px 50px 75px 100px;
    background-color: lightblue;
}
</style>
</head>
<body>

<h2>The padding shorthand property - 4 values</h2>

<div>This div element has a top padding of 25px, a right padding of 50px, a
bottom padding of 75px, and a left padding of 100px.</div>

</body>
</html>
```



If the `padding` property has three values:

- **padding: 25px 50px 75px;**
  - top padding is 25px
  - right and left paddings are 50px
  - bottom padding is 75px

If the `padding` property has two values:

- **padding: 25px 50px;**
  - top and bottom paddings are 25px
  - right and left paddings are 50px

If the `padding` property has one value:

- **padding: 25px;**
  - all four paddings are 25px

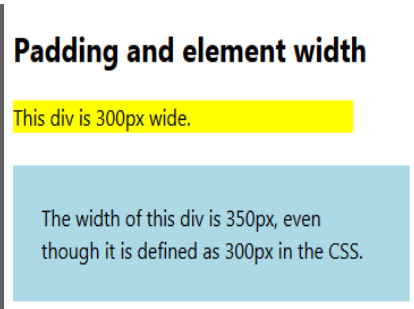
PADDING AND ELEMENT WIDTH

The CSS `width` property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element (the box model).

So, if an element has a specified width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

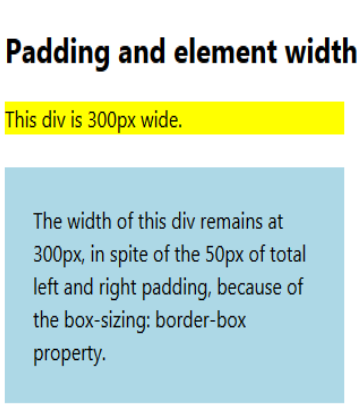
In the following example, the element is given a width of 300px. However, the actual rendered width of the element will be 350px (300px + 25px of left padding + 25px of right padding):

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
    width: 300px;
    background-color: yellow;
}

div.ex2 {
    width: 300px;
    padding: 25px;
    background-color: lightblue;
}
</style>
</head>
<body>

<h2>Padding and element width</h2>

<div class="ex1">This div is 300px wide.</div>
<br>

<div class="ex2">The width of this div is 350px, even though it is defined as
300px in the CSS.</div>
```



To keep the width at 300px, no matter the amount of padding, you can use the **`box-sizing`** property. This causes the element to maintain its width; if you increase the padding, the available content space will decrease. Here is an example:

```
<style>
div.ex1 {
    width: 300px;
    background-color: yellow;
}

div.ex2 {
    width: 300px;
    padding: 25px;
    box-sizing: border-box;
    background-color: lightblue;
}
</style>
</head>
<body>

<h2>Padding and element width</h2>

<div class="ex1">This div is 300px wide.</div>
<br>

<div class="ex2">The width of this div remains at 300px, in spite of the 50px
of total left and right padding, because of the box-sizing: border-box
property.
</div>
```



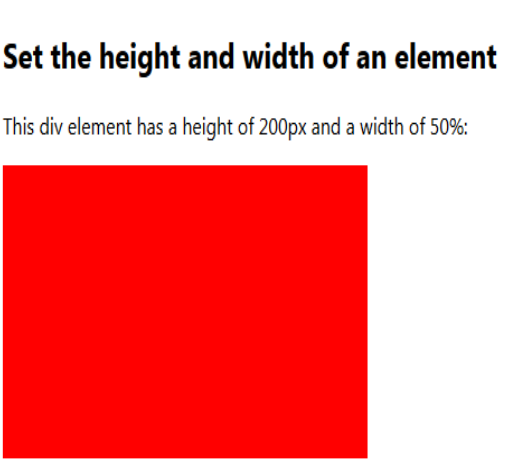
CSS HEIGHT AND WIDTH

SETTING HEIGHT AND 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.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    height: 200px;
    width: 50%;
    background-color: red;
}
</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>
```



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).

The problem with the `<div>` above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.

Using `max-width` instead, in this situation, will improve the browser's handling of small windows.

```
div {
max-width: 500px;
height: 100px;
background-color: powderblue;
}
```

ALL CSS DIMENSION PROPERTIES

Property	Description
height	Sets the height of an element
max-	Sets the maximum height of an element

height	
max-width	Sets the maximum width of an element
min-height	Sets the minimum height of an element
min-width	Sets the minimum width of an element
width	Sets the width of an element

CSS 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. The image below illustrates the box model:


```
<html>
<head>
<style>
div {
    background-color: lightgrey;
    width: 300px;
    border: 25px solid lightgreen;
    padding: 25px;
    margin: 25px;
}
</style>
</head>
<body>

<p>The CSS box model is essentially a box that wraps around every HTML element.
It consists of: borders, padding, margins, and the actual content.</p>

<div>This text is the actual content of the box. We have added a 25px padding,
25px margin and a 25px green border. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum.</div>

</body>
</html>
```

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

WIDTH AND HEIGHT OF AN ELEMENT

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

Important: When you set the width and height properties of an element with CSS, you just set the width and height of the **content area**. To calculate the full size of an element, you must also add padding, borders and margins.

```
<!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>
```

Calculate the total width:

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

Here is the calculation:

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

The total width of an element should be calculated like this:

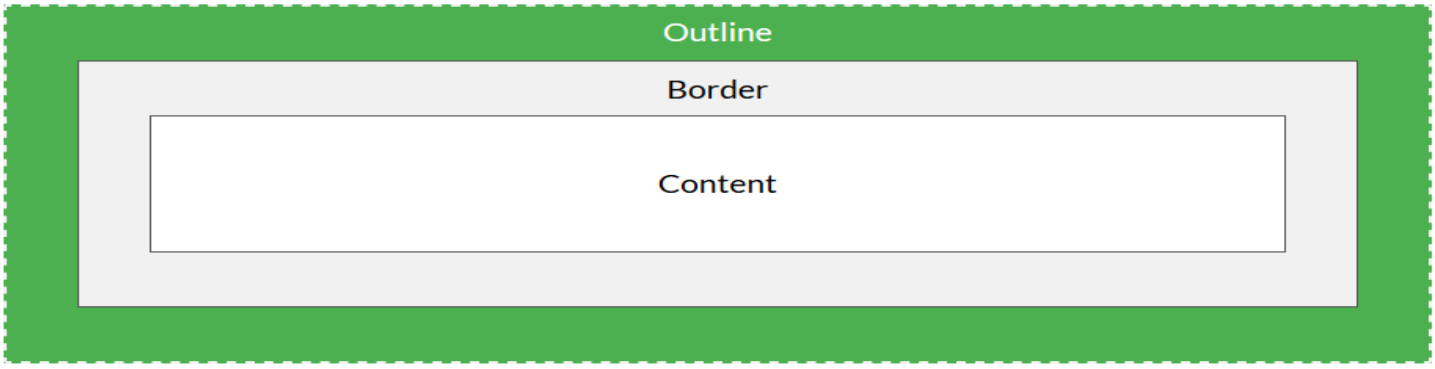
Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS OUTLINE

An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".



CSS has the following outline properties:

- `outline-style`
- `outline-color`
- `outline-width`
- `outline-offset`
- `outline`

Note: Outline differs from borders! Unlike border, the outline is drawn outside the element's border, and may overlap other content. Also, the outline is NOT a part of the element's dimensions; the element's total width and height is not affected by the width of the outline.

OUTLINE STYLE

The `outline-style` property specifies the style of the outline, and 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
- `ridge` - Defines a 3D ridged outline
- `inset` - Defines a 3D inset outline
- `outset` - Defines a 3D outset outline
- `none` - Defines no outline
- `hidden` - Defines a hidden outline

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
    border: 4px solid red;
    outline-style: solid;
    outline-color:  #000000;
}
</style>
</head>
<body>
```

Using outline-color:invert

A solid invert outline.

CSS TEXT

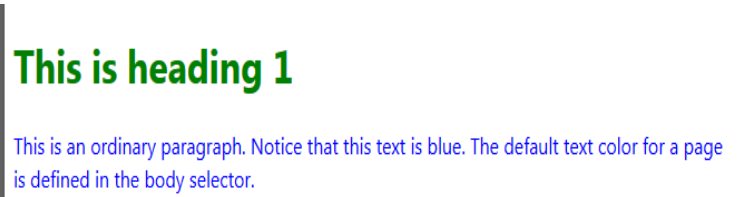
The `color` property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

Look at CSS Color Values for a complete list of possible color values.

The default text color for a page is defined in the body selector.

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

h1 {
    color: green;
}
</style>
</head>
<body>
```



Note: For W3C compliant CSS: If you define the `color` property, you must also define the `background-color`.

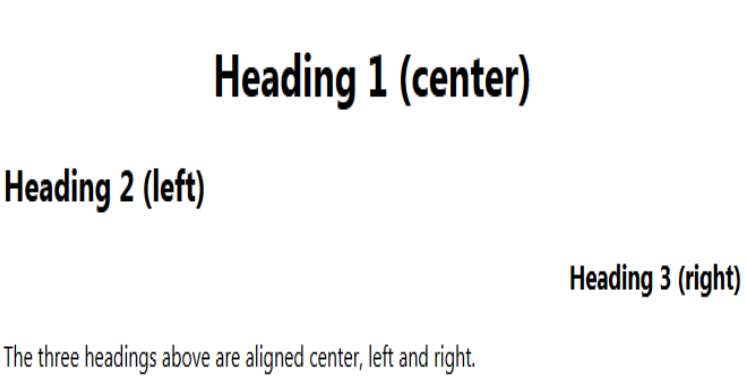
TEXT ALIGNMENT

The `text-align` property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    text-align: center;
}

h2 {
    text-align: left;
}

h3 {
    text-align: right;
}
</style>
</head>
<body>

<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>

<p>The three headings above are aligned center, left and right.</p>
```



When the `text-align` property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers):

```
div {
text-align: justify;
}
```

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>
a {
    text-decoration: none;
}
</style>
</head>
<body>

<p>A link with no underline: <a href="https://www.quran.com">Quran</a></p>

</body>
</html>
```

A link with no underline: Quran

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    text-decoration: overline;
}

h2 {
    text-decoration: line-through;
}

h3 {
    text-decoration: underline;
}
</style>
```

This is heading 1

~~This is heading 2~~

This is heading 3

Note: It is not recommended to underline text that is not a link, as this often confuses the reader.

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>
```

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: 60px;
}
</style>
</head>
<body>

<p>In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'</p>

</body>
</html>
```

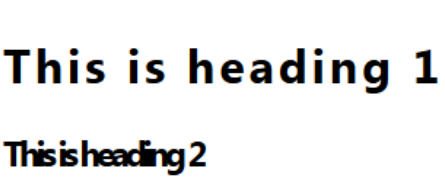
In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'

LETTER SPACING

The `letter-spacing` property is used to specify the space between the characters in a text.

The following 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>
```



LINE HEIGHT

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

```
<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>
```

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.

TEXT DIRECTION

The `direction` property is used to change the text direction of an element:

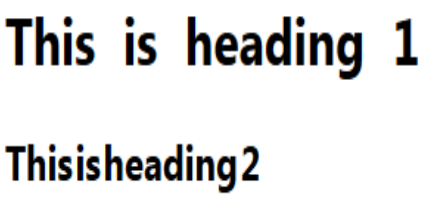
```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
    direction: rtl;
}
</style>
</head>
<body>

<p>This is the default text direction.</p>
<p class="ex1"><bdo dir="rtl">This is right-to-left text direction.</bdo></p>

</body>
</html>
```

WORD SPACING

The `word-spacing` property is used to specify the space between the words in a text.

The following example demonstrates how to increase or decrease the space between words:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    word-spacing: 10px;
}

h2 {
    word-spacing: -5px;
}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>

</body>
</html>
```

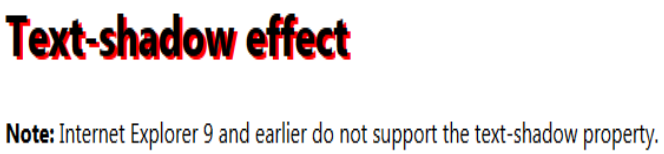


TEXT SHADOW

The `text-shadow` property adds shadow to text.

The following example specifies the position of the horizontal shadow (3px), the position of the vertical shadow (2px) and the color of the shadow (red):

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    text-shadow: 3px 2px red;
}
</style>
</head>
<body>

<h1>Text-shadow effect</h1>
<p><b>Note:</b> Internet Explorer 9 and earlier do not support the text-shadow
property.</p>

</body>
</html>
```



Note: Internet Explorer 9 and earlier do not support the text-shadow property.

All CSS Text Properties

Property	Description
color	Sets the color of text
direction	Specifies the text direction/writing direction
letter-spacing	Increases or decreases the space between characters in a text
line-height	Sets the line height
text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block
text-shadow	Specifies the shadow effect added to text
text-transform	Controls the capitalization of text
text-overflow	Specifies how overflowed content that is not displayed should be signaled to the user
unicode-bidi	Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document
vertical-align	Sets the vertical alignment of an element
white-space	Specifies how white-space inside an element is handled
word-spacing	Increases or decreases the space between words in 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")

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

Note: On computer screens, sans-serif fonts are considered easier to read than serif fonts.

FONT FAMILY

The font family of a text is set with the `font-family` property.

The `font-family` property should hold several font names as a "fallback" 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.

Note: 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:

```
<!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>
```

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)

FONT SIZE

The `font-size` property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

The 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

Note: If you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).

FONT WEIGHT

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

```
<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>
```

This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

RESPONSIVE FONT SIZE

The text size can be set with a `vw` unit, which means the "viewport width".

That way the text size will follow the size of the browser window:

SALAM 3alaykom!

Resize the browser window to see how the font size scales.

EXAMPLE

```
<h1 style="font-size:10vw">SALAM 3alaykom!</h1>
```

FONT VARIANT

The `font-variant` property specifies whether or not a text should be displayed in a small-caps font.

In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appears in a smaller font size than the original uppercase letters in the text.

EXAMPLE

```
p.normal {
  font-variant: normal;
}

p.small {
  font-variant: small-caps;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
    font-variant: normal;
}

p.small {
    font-variant: small-caps;
}
</style>
</head>
<body>

<p class="normal">My name is Souhail.</p>
<p class="small">My name is Souhail.</p>

</body>
</html>
```

My name is Souhail.

MY NAME IS SOUHAIL.

ALL CSS FONT PROPERTIES

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font

CSS ICONS

HOW TO ADD ICONS

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 `<span>`).

All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.)

FONT AWESOME ICONS

To use the Font Awesome icons, add the following line inside the `<head>` section of your HTML page:

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
```

Note: No downloading or installation is required!

```html
<!DOCTYPE html>
<html>
<head>
<title>Font Awesome Icons</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>

<p>Some Font Awesome icons:</p>
<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>

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

</body>
</html>
```

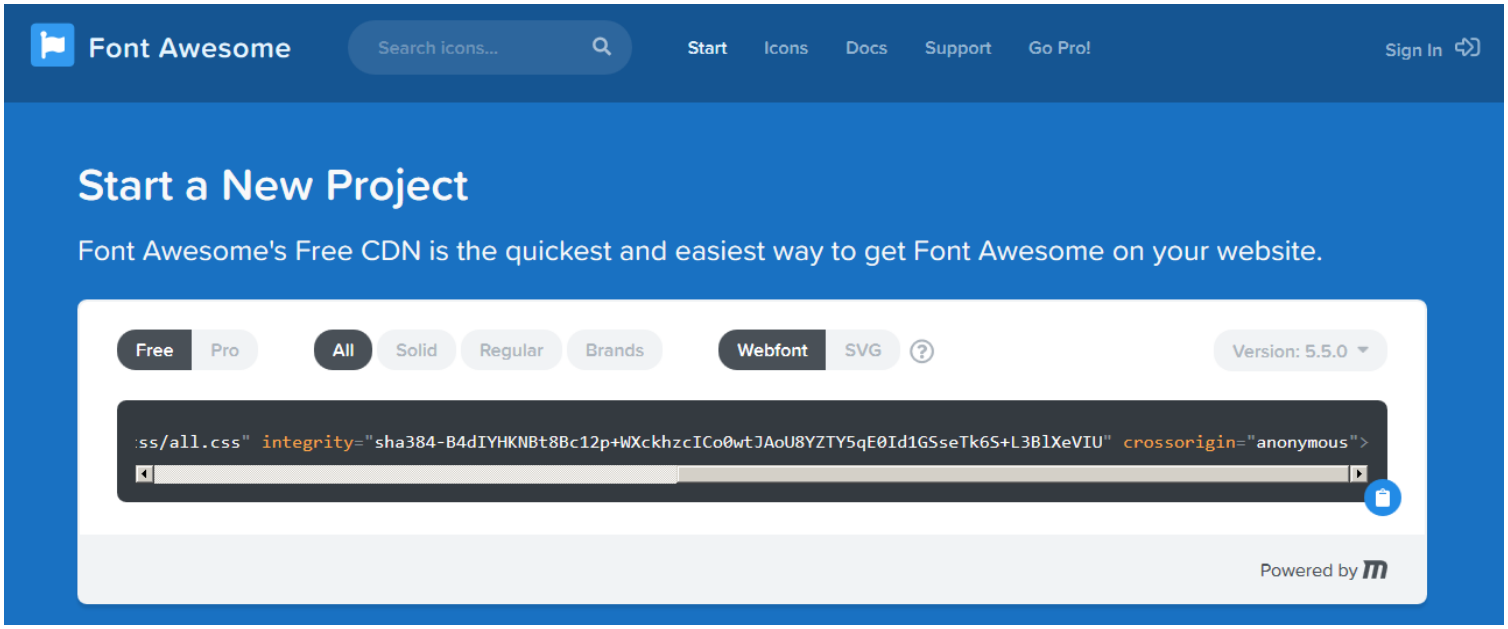
Some Font Awesome icons:

Styled Font Awesome icons (size and color):

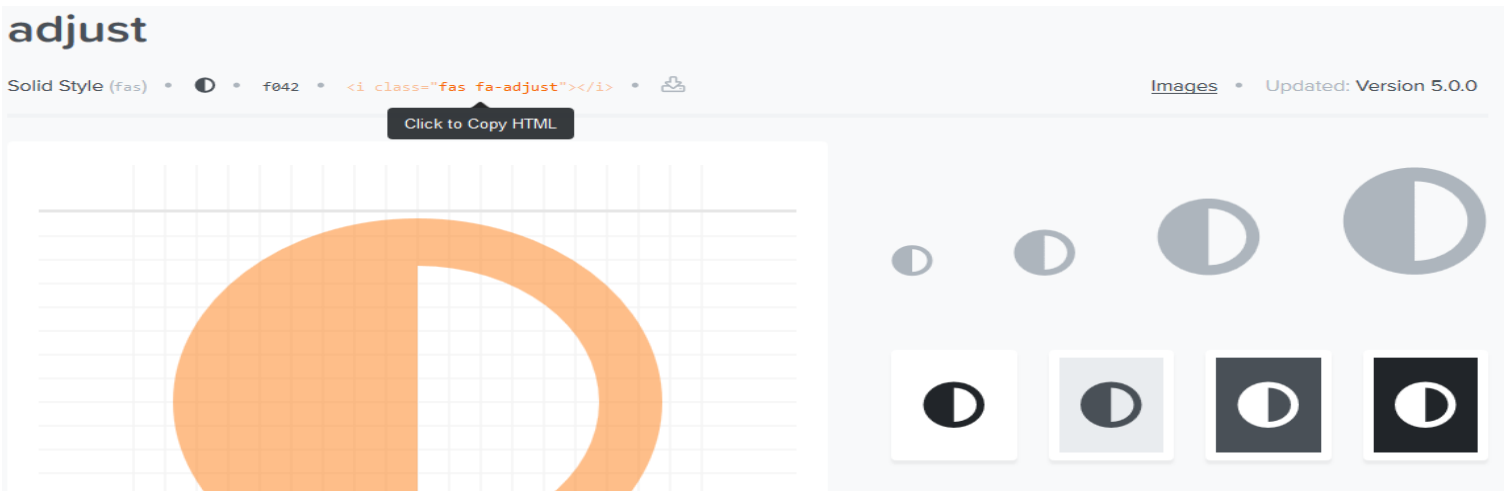
موقع جميل لتحميل اكونز

/https://fontawesome.com

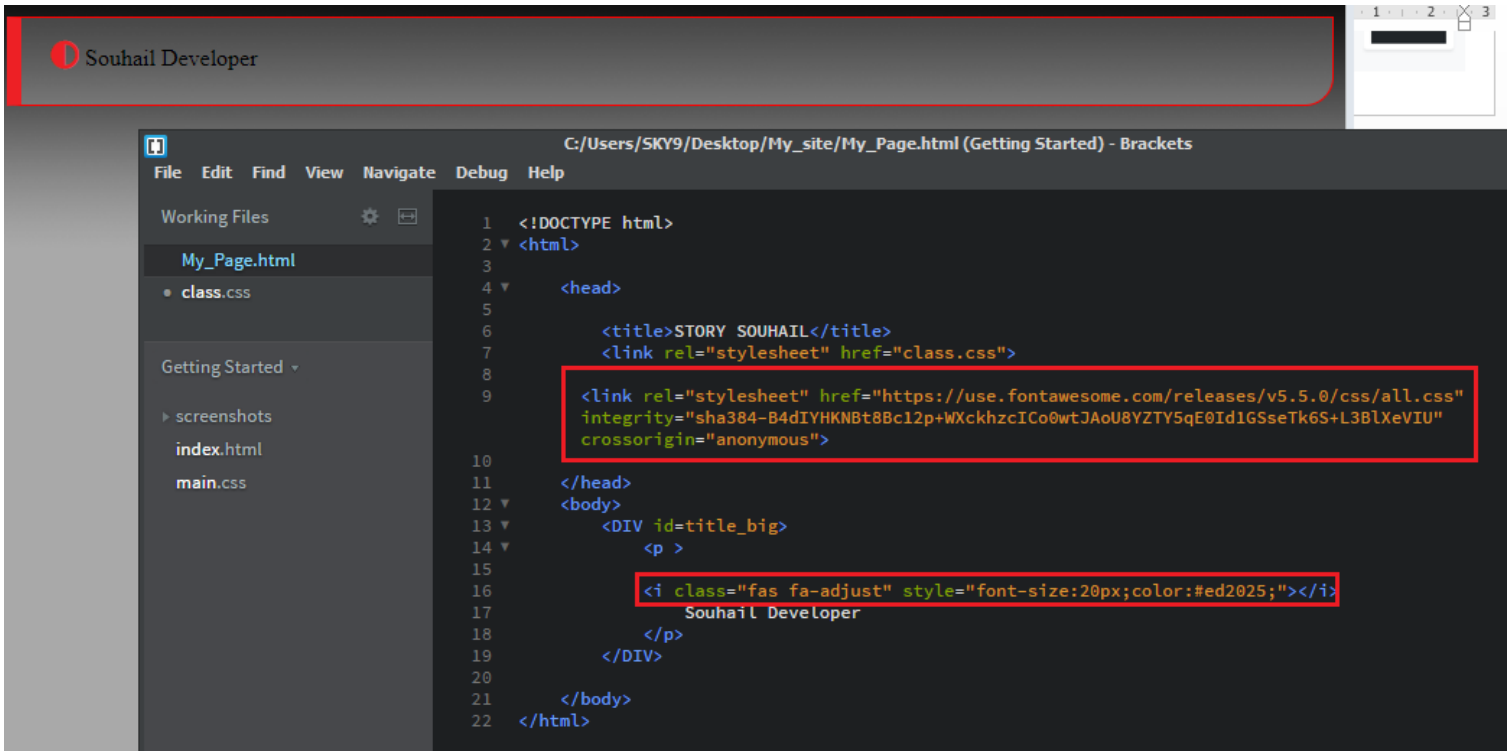
انسخ الرابط الذي يعطيه لك الموقع



ثم اختر أيقونة ونسخ اسمها



وتم ادخال الأيقونة بنجاح و الحمد لله



link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-><"B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeVIU" crossorigin="anonymous

BOOTSTRAP ICONS

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.m
in.css">
```

Note: No downloading or installation is required!

```
<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>
```

Some Bootstrap icons:

Styled Bootstrap icons (size and color):

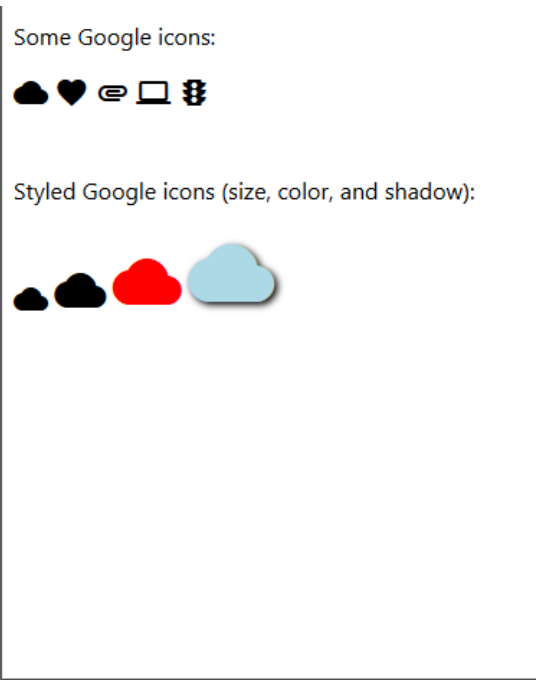
GOOGLE ICONS

To use the Google icons, add the following line inside the `<head>` section of your HTML page:

```
<link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
```

Note: No downloading or installation is required!

```
<html>
<head>
<title>Google Icons</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://fonts.googleapis.com
/icon?family=Material+Icons">
</head>
<body>

<p>Some Google icons:</p>
<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>
<br><br>

<p>Styled Google icons (size, color, and shadow):</p>
<i class="material-icons" style="font-size:24px;">cloud</i>
<i class="material-icons" style="font-size:36px;">cloud</i>
<i class="material-icons" style="font-size:48px;color:red;">cloud</i>
<i class="material-icons" style="font-size:60px;color:lightblue;text-shadow:2px
2px 4px #000000;">cloud</i>

</body>
</html>
```



هنا ستجد مجموعة من أسماء أيقونات جاهزة من مواقع مختلفة

https://www.w3schools.com/icons/icons_reference.asp

موقع آخر قد تحتاجه

/http://fontello.com

CSS LINKS

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 mouses over it
- `a:active` - a link the moment it is clicked

```css
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
```

When setting the style for several link states, there are some order rules:

- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

TEXT DECORATION

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

```css
a:link {
text-decoration: none;
}

a:visited {
text-decoration: none;
}

a:hover {
text-decoration: underline;
}

a:active {
text-decoration: underline;
}
```

BACKGROUND COLOR

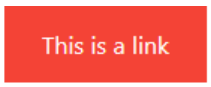
The `background-color` property can be used to specify a background color for links:

```
a:link {
background-color: yellow;
}

a:visited {
background-color: cyan;
}

a:hover {
background-color: lightgreen;
}

a:active {
background-color: hotpink;
}
```

ADVANCED - LINK BUTTONS

This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes/buttons:

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link, a:visited {
    background-color: #f44336;
    color: white;
    padding: 14px 25px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
}


a:hover, a:active {
    background-color: red;
}
</style>
</head>
<body>

<a href="default.asp" target="_blank">This is a link</a>

</body>
</html>
```



CSS LISTS

DIFFERENT LIST ITEM MARKERS

The `list-style-type` property specifies the type of list item marker.

The following example shows some of the available list item markers:

```
ul.a {
    list-style-type: circle;
}

ul.b {
    list-style-type: square;
}

ol.c {
    list-style-type: upper-roman;
}

ol.d {
    list-style-type: lower-alpha;
}
</style>
</head>
<body>

<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

Example of unordered lists:

- Coffee
- Tea
- Coca Cola

- Coffee
- Tea
- Coca Cola

Example of ordered lists:

I. Coffee
II. Tea
III. Coca Cola

a. Coffee
b. Tea
c. Coca Cola

AN IMAGE AS THE LIST ITEM MARKER

The `list-style-image` property specifies an image as the list item marker:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-image: url('sqpurple.gif');
}
</style>
</head>
<body>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

</body>
</html>
```

- Coffee
- Tea
- Coca Cola

POSITION THE LIST ITEM MARKERS

The `list-style-position` property specifies the position of the list-item markers (bullet points).

"list-style-position: outside;" means that the bullet points will be outside the list item. The start of each line of a list item will be aligned vertically. This is default:

```
<html>
<head>
<style>
ul.a {
    list-style-position: outside;
}

ul.b {
    list-style-position: inside;
}
</style>
</head>
<body>
<h1>The list-style-position Property</h1>

<h2>list-style-position: outside (default):</h2>
<ul class="a">
  <li>Coffee - A brewed drink prepared from roasted coffee beans, which are the seeds of berries from the Coffea plant</li>
  <li>Tea - An aromatic beverage commonly prepared by pouring hot or boiling water over cured leaves of the Camellia sinensis, an evergreen shrub (bush) native to Asia</li>
  <li>Coca Cola - A carbonated soft drink produced by The Coca-Cola Company. The drink's name refers to two of its original ingredients, which were kola nuts (a source of caffeine) and coca leaves</li>
```

list-style-position: outside (default):

- Coffee - A brewed drink prepared from roasted coffee beans, which are the seeds of berries from the Coffea plant
- Tea - An aromatic beverage commonly prepared by pouring hot or boiling water over cured leaves of the Camellia sinensis, an evergreen shrub (bush) native to Asia
- Coca Cola - A carbonated soft drink produced by The Coca-Cola Company. The drink's name refers to two of its original ingredients, which were kola nuts (a source of caffeine) and coca leaves

list-style-position: inside:

- Coffee - A brewed drink prepared from roasted coffee beans, which are the seeds of berries from the Coffea plant
- Tea - An aromatic beverage commonly prepared by pouring hot or boiling water over cured leaves of the Camellia sinensis, an evergreen shrub (bush) native to Asia
- Coca Cola - A carbonated soft drink produced by The Coca-Cola Company. The drink's name refers to two of its original ingredients, which were kola nuts (a source of caffeine) and coca leaves

REMOVE DEFAULT SETTINGS

The `list-style-type:none` property can also be used to remove the markers/bullets. Note that the list also has default margin and padding. To remove this, add `margin:0` and `padding:0` to <ul> or <ol>:

```
<html>
<head>
<style>
ul.demo {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
</style>
</head>
<body>

<p>Default list:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>

<p>Remove bullets, margin and padding:</p>
<ul class="demo">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

Default list:

- Coffee
- Tea
- Coca Cola

Remove bullets, margin and padding:

Coffee
Tea
Coca Cola

STYLING LIST WITH COLORS

We can also style lists with colors, to make them look a little more interesting.

Anything added to the <ol> or <ul> tag, affects the entire list, while properties added to the <li> tag will affect the individual list items:

```
ol {
    background: #ff9999;
    padding: 20px;
}

ul {
    background: #3399ff;
    padding: 20px;
}

ol li {
    background: #ffe5e5;
    padding: 5px;
    margin-left: 35px;
}

ul li {
    background: #cce5ff;
    margin: 5px;
}
```

Result:

1. Coffee
2. Tea
3. Coca Cola

- Coffee
- Tea
- Coca Cola

CSS TABLES

TABLE BORDERS

To specify table borders in CSS, use the `border` property.

The example below specifies a black border for <table>, <th>, and <td> elements:

EXAMPLE

```
table, th, td {
  border: 1px solid black;
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

CSS TABLE PROPERTIES

Property	Description
border	Sets all the border properties in one declaration
border-collapse	Specifies whether or not table borders should be collapsed
border-spacing	Specifies the distance between the borders of adjacent cells
caption-side	Specifies the placement of a table caption
empty-cells	Specifies whether or not to display borders and background on empty cells in a table
table-layout	Sets the layout algorithm to be used for a table

https://www.w3schools.com/css/css_table.asp

CSS LAYOUT - THE DISPLAY PROPERTY

DISPLAY: NONE;

`display: none;` is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.

The `<script>` element uses `display: none;` as default.

OVERRIDE THE DEFAULT DISPLAY VALUE

As mentioned, every element has a default display value. However, you can override this.

Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow the web standards.

A common example is making inline `<li>` elements for horizontal menus:

```
<html>
<head>
<style>
li {
    display: inline;

    /********** سبحان الله وبحمده ****/
    border:1px solid red; /* أستغفر الله و أتوب اليه */
    padding:10px;
    background-color:lightgreen;
    font-weight:bold;

}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<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>
```

Display a list of links as a horizontal menu:

HTML CSS JavaScript

Note: Setting the display property of an element only changes **how the element is displayed**, NOT what kind of element it is. So, an inline element with `display: block;` is not allowed to have other block elements inside it.

The following example displays elements as block elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
#salam_3alykom{
    display: block;
}
</style>
</head>
<body>

example1:
<span id="salam_3alykom">A display property with a value of "block" results
in</span>
<span id="salam_3alykom">a line break between the two elements.</span>

<hr>

example2:
<span>A display property with a value of "block" results in</span>
<span>a line break between the two elements.</span>

</body>
</html>
```

example1:
A display property with a value of "block" results in
a line break between the two elements.

---

example2: A display property with a value of "block" results in a line break between the two elements.

HIDE AN ELEMENT - DISPLAY:NONE OR VISIBILITY:HIDDEN?

Hiding an element can be done by setting the `display` property to `none`. The element will be hidden, and the page will be displayed as if the element is not there:

EXAMPLE

```
h1.hidden {
  display: none;
}
```

`visibility:hidden;` also hides an element.

However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

EXAMPLE

```
h1.hidden {
  visibility: hidden;
}
```

https://www.w3schools.com/css/css_display_visibility.asp

CSS LAYOUT - WIDTH AND MAX-WIDTH

USING WIDTH, MAX-WIDTH AND MARGIN: AUTO;

As mentioned in the previous chapter; a block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Setting the `width` of a block-level element will prevent it from stretching out to the edges of its container. Then, you can set the margins to auto, to horizontally center the element within its container. The element will take up the specified width, and the remaining space will be split equally between the two margins:

Note: The problem with the `<div>` above occurs when the browser window is smaller than the width of the element. The browser then adds a horizontal scrollbar to the page.

Using `max-width` instead, in this situation, will improve the browser's handling of small windows. This is important when making a site usable on small devices:

Tip: Resize the browser window to less than 500px wide, to see the difference between the two divs!

Here is an example of the two divs above:

EXAMPLE

```
div.ex1 {
  width: 500px;
  margin: auto;
  border: 3px solid #73AD21;
}

div.ex2 {
  max-width: 500px;
  margin: auto;
  border: 3px solid #73AD21;
}
```

```
<style>
div.ex1 {
    width:500px;
    margin: auto;
    border: 3px solid #73AD21;
}

div.ex2 {
    max-width:500px;
    margin: auto;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>

<div class="ex1">This div element has width: 500px;</div>
<br>

<div class="ex2">This div element has max-width: 500px;</div>

<p><strong>Tip:</strong> Drag the browser window to smaller than 500px wide, to
see the difference between
the two divs!</p>

</body>
```

This div element has width: 500px;

This div element has max-width: 500px;

Tip: Drag the browser window to smaller than 500px wide, to see the difference between the two divs!

CSS LAYOUT - THE POSITION PROPERTY

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

THE POSITION PROPERTY

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

There are five different position values:

- `static`
- `relative`
- `fixed`
- `absolute`
- `sticky`

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the `position` property is set first. They also work differently depending on the position value.

POSITION: STATIC;

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

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

```
div.static {
position: static;
border: 3px solid #73AD21;
}
```

POSITION: RELATIVE;

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

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

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
    position: relative;
    left: 10px;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: relative;</h2>

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

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

position: relative;

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

This div element has position: relative;

POSITION: FIXED;

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

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

<h2>position: fixed;</h2>

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

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

</body>
</html>
```

position: fixed;

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

This div element has position: fixed;

POSITION: ABSOLUTE;

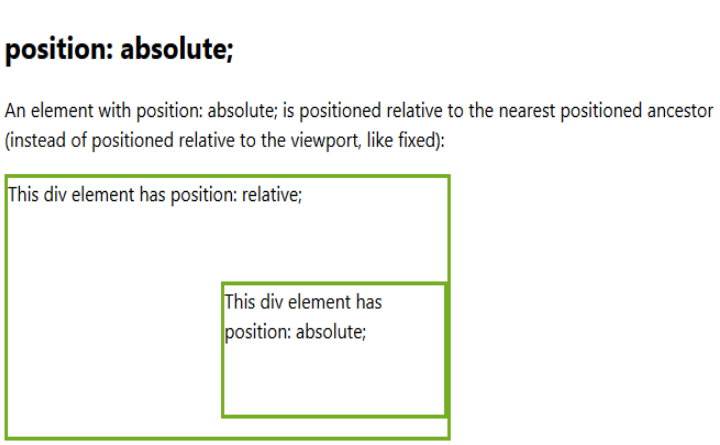
An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Note: A "positioned" element is one whose position is anything except `static`.

Here is a simple example:

```
<style>
div.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>position: absolute;</h2>
<p>An element with position: absolute; is positioned relative to the nearest
positioned ancestor (instead of positioned relative to the viewport, like
fixed):</p>
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
```



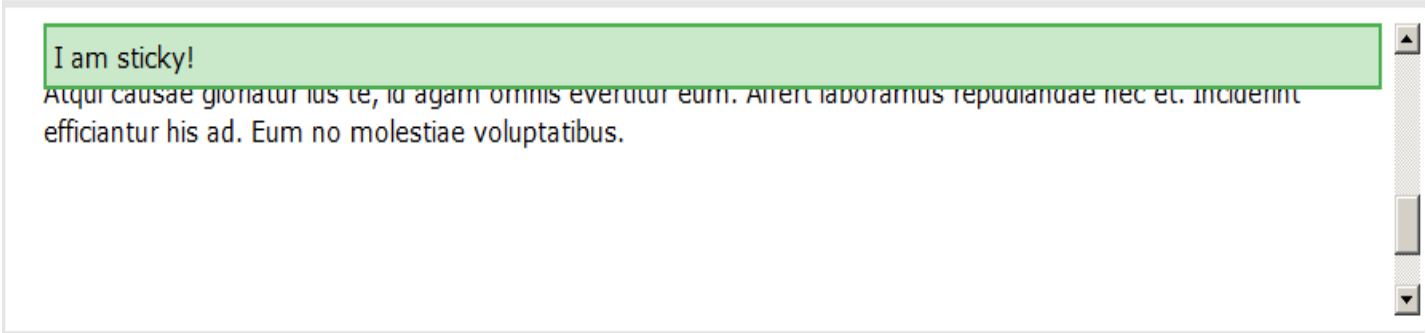
POSITION: STICKY;

An element with `position: sticky;` is positioned based on the user's scroll position.

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

Note: Internet Explorer, Edge 15 and earlier versions do not support sticky positioning. Safari requires a -webkit- prefix (see example below). You must also specify at least one of `top`, `right`, `bottom` or `left` for sticky positioning to work.

```
div.sticky {
position: -webkit-sticky; /* Safari */
position: sticky;
top: 0;
background-color: green;
border: 2px solid #4CAF50;
}
```



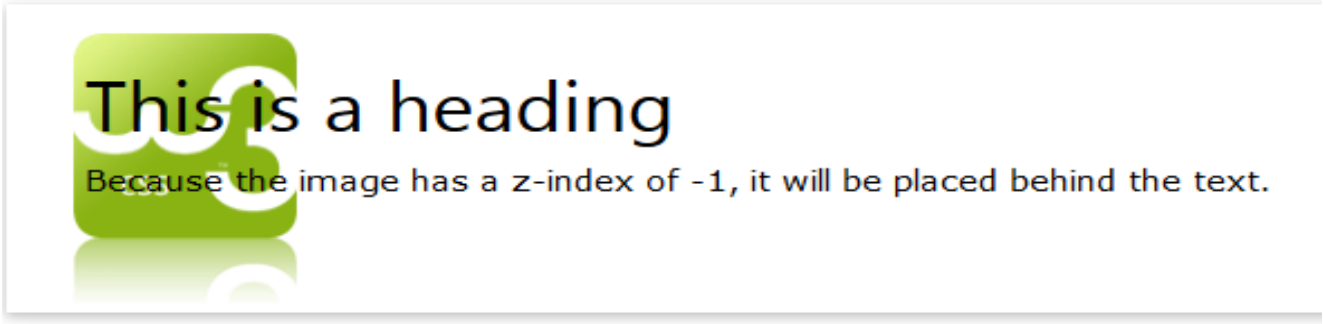
OVERLAPPING ELEMENTS

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:

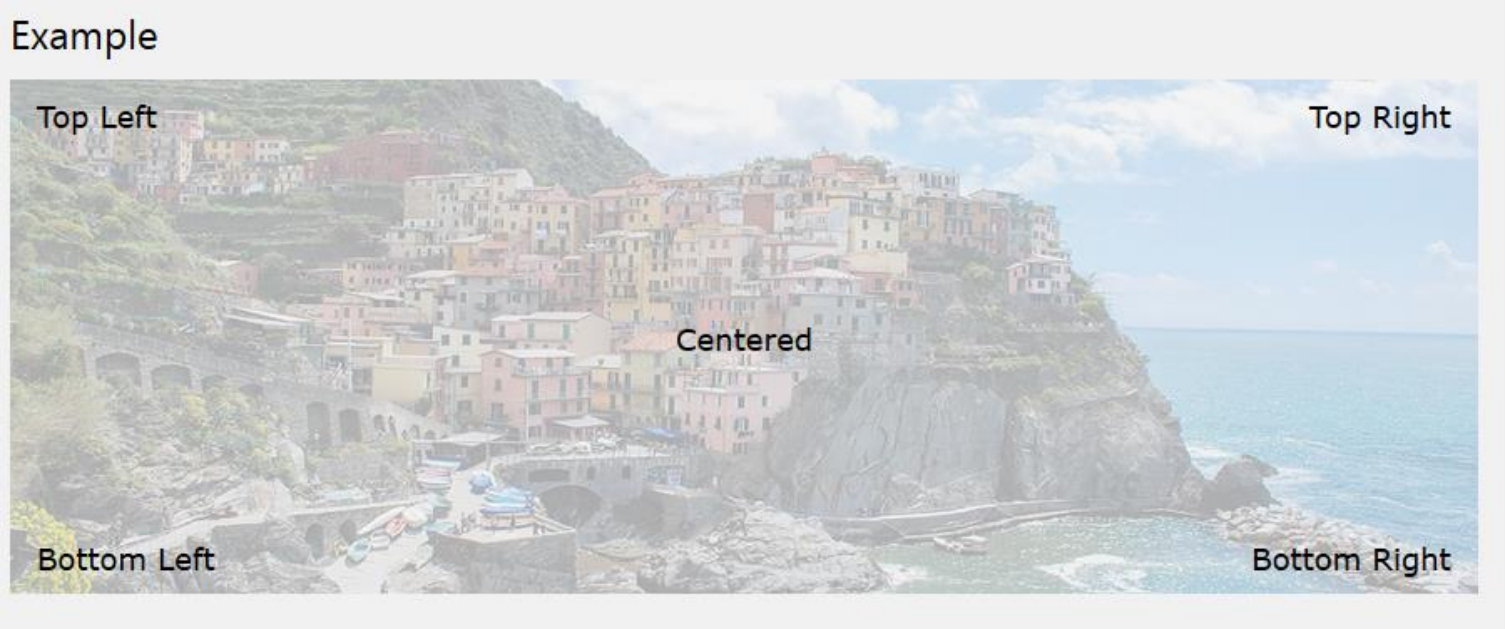
```
img {
position: absolute;
left: 0px;
top: 0px;
z-index: -1;
}
```



Note: If two positioned elements overlap without a `z-index` specified, the element positioned last in the HTML code will be shown on top.

POSITIONING TEXT IN AN IMAGE

How to position text over an image:



الى هنا ينتهي الجزء الأول

نلتقي في الجزء الثاني ان شاء الله

Souhail.developer@gmail.com

https://ebook-sl-d3v.blogspot.com

souhail developer – مع خالص تحياتي

صل على الحبيب المصطفى .