

LAB 2: INTRODUCTION TO CSS

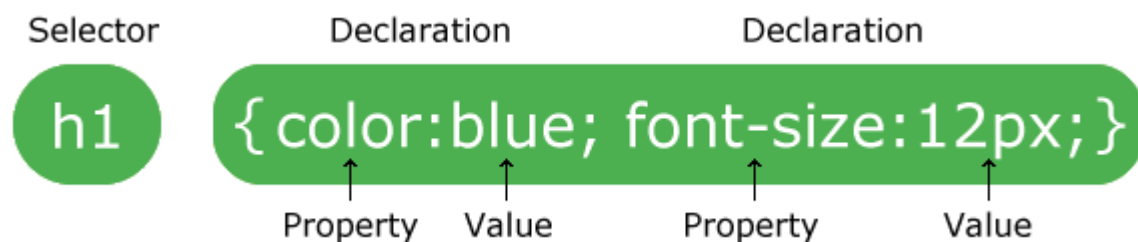
What is CSS?

CSS stands for **C**ascading **S**tyle **S**heets

- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.
- The style definitions are normally saved in external .css files. With an external stylesheet file, you can change the look of an entire website by changing just one file!

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.

Example

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

```
p {  
  color: red;  
}
```

```
text-align: center;
}
```

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the `<link>` element, inside the head section.

Example

External styles are defined within the `<link>` element, inside the `<head>` section of an HTML page:

```
<!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>
```

An external style sheet can be written in any text editor, and must be saved with a `.css` extension. The external `.css` file should not contain any HTML tags. Here is how the "mystyle.css" file looks like:

"mystyle.css"

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

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

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the `<style>` element, inside the head section.

Example

Internal styles are defined within the `<style>` element, inside the `<head>` section of an HTML page:

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

Inline CSS

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.

Example

Inline styles are defined within the "style" attribute of the relevant element:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
</body>
</html>
```

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser defaults.

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers.

```
p {
  color: red;
  /* This is a single-line comment */
  text-align: center;
}

/* This is
a multi-line
comment */
```

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)

- [Combinator selectors](#) (select elements based on a specific relationship between them)
- [Pseudo-class selectors](#) (select elements based on a certain state)
- [Pseudo-elements selectors](#) (select and style a part of an element)
- [Attribute selectors](#) (select elements based on an attribute or attribute value)

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p {  
  text-align: center;  
  color: red;  
}
```

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

Example

In this example all HTML elements with class="center" will be red and center-aligned:

```
. center {  
  text-align: center;  
  color: red;  
}
```

HTML elements can also refer to more than one class.

Example

In this example the <p> element will be styled according to class="center" and to class="large":

```
<p class="center large">This paragraph refers to two classes.</p>
```

he CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Example

The CSS rule below will affect every HTML element on the page:

```
* {  
  text-align: center;  
  color: blue;  
}
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 {  
  text-align: center;  
  color: red;  
}  
  
h2 {  
  text-align: center;  
  color: red;  
}  
  
p {  
  text-align: center;  
  color: red;  
}
```

It will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

CSS Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values. You can set the background color for HTML elements:

Example

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

You can set the color of text:

Example

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

You can set the color of borders:

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>
```

colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

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

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

In CSS, a color can be specified as an RGB value, using this formula:

rgb(*red*, *green*, *blue*)

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.

For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255) and the others are set to 0.

To display the color black, all color parameters must be set to 0, like this: rgb(0, 0, 0).

To display the color white, all color parameters must be set to 255, like this: rgb(255, 255, 255).

In CSS, a color can be specified using a hexadecimal value in the form: **#rrggbb**, Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255). For example, #ff0000 is displayed as red, because red is set to its highest value (ff) and the others are set to the lowest value (00).

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

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

Example

The background color of a page is set like this:

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

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.

Example

The background image for a page can be set like this:

```
body {  
  background-image: url("YourImage1.gif");  
}
```

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:

Example

```
body {  
  background-image: url("YourImage2.png");  
}
```

If the image above is repeated only horizontally (`background-repeat: repeat-x;`), the background will look better:

Example

```
body {  
  background-image: url("YourImage2.png");  
  background-repeat: repeat-x;  
}
```

Showing the background image only once is also specified by the `background-repeat` property:

Example

Show the background image only once:

```
body {  
  background-image: url("YourImage3.png");  
  background-repeat: no-repeat;  
}
```

The `background-position` property is used to specify the position of the background image.

Example

Position the background image in the top-right corner:

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

The `background-attachment` property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page):

Example

Specify that the background image should be fixed:

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

If we specify that the background image should scroll with the rest of the page:

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

CSS Borders

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

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).

Example

Demonstration of the different border styles:

```
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;}
```

The `border-width` property specifies the width of the four borders.

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;  
}  
  
p.two {  
  border-style: solid;  
  border-width: medium;  
}  
  
p.three {  
  border-style: solid;  
  border-width: 2px 10px 4px 20px;  
}
```

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

Example

```
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;  
}
```

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

Left Border

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

Bottom Border

```
p {  
  border-bottom: 6px solid red;
```

```
background-color: lightgrey;
}
```

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).

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.

Example

Set different margins for all four sides of a `<p>` element:

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

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

Example

Use the margin shorthand property with four values:

```
p {  
  margin: 25px 50px 75px 100px;  
}
```

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.

Example

Use margin: auto:

```
div {  
  width: 300px;  
  margin: auto;  
  border: 1px solid red;  
}
```

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!

Look at the following example:

Example

Demonstration of margin collapse:

```
h1 {  
  margin: 0 0 50px 0;  
}  
  
h2 {  
  margin: 20px 0 0 0;  
}
```

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.

Example

Set different padding for all four sides of a `<div>` element:

```
div {  
  padding-top: 50px;  
  padding-right: 30px;  
  padding-bottom: 50px;  
  padding-left: 80px;  
}
```

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

Example

Use the padding shorthand property with four values:

```
div {  
  padding: 25px 50px 75px 100px;  
}
```

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

adding 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.

Example

Here, the `<div>` element is given a width of 300px. However, the actual width of the `<div>` element will be 350px (300px + 25px of left padding + 25px of right padding):

```
div {  
  width: 300px;  
  padding: 25px;  
}
```

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.

Example

Use the `box-sizing` property to keep the width at 300px, no matter the amount of padding:

```
div {  
  width: 300px;  
  padding: 25px;  
  box-sizing: border-box;  
}
```

CSS Setting height and width

The `height` and `width` properties are used to set the height and width of an element.

The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

CSS height/width Values

The `height` and `width` properties may have the following values:

- `auto` - This is default. The browser calculates the height and width
- `length` - Defines the height/width in px, cm etc.
- `%` - Defines the height/width in percent of the containing block
- `initial` - Sets the height/width to its default value
- `inherit` - The height/width will be inherited from its parent value

CSS height/width Examples

Example

Set the height and width of a `<div>` element:

```
div {  
  height: 200px;  
  width: 50%;  
  background-color: powderblue;}
```

Example

This `<div>` element has a height of 100 pixels and a max-width of 500 pixels:

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

Setting max-width

Example

Set the height and width of another `<div>` element:

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

The 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:

```
<!DOCTYPE html>

<html>

<head>

<style>

div {

    background-color: lightgrey;

    width: 300px;

    border: 15px solid green;

    padding: 50px;

    margin: 20px;

}

</style>

</head>

<body>

<h2>Demonstrating the Box Model</h2>

<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 content of the box. We have added a 50px padding, 20px margin and a 15px 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>
```

Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

CSS 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

The following example shows the different `outline-style` values:

A dotted outline.

A dashed outline.

A solid outline.

A double outline.

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

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

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

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

Example

Demonstration of the different outline styles:

```
p.dotted {outline-style: dotted;}  
p.dashed {outline-style: dashed;}  
p.solid {outline-style: solid;}
```

```
p.double {outline-style: double;}
p.groove {outline-style: groove;}
p.ridge {outline-style: ridge;}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
```

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):

Example

```
h1 {
  text-align: center;
}

h2 {
  text-align: left;
}

h3 {
  text-align: 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:

Example

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

The other `text-decoration` values are used to decorate text:

Example

```
h1 {  
  text-decoration: underline;  
}  
  
h2 {  
  text-decoration: line-through;  
}  
  
h3 {  
  text-decoration: underline;  
}
```

Text Indentation

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

Example

```
p {  
  text-indent: 50px;  
}
```

CSS Font Families

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
Sans-serif	Arial Verdana	"Sans" means without - the characters

Monospace

Courier New
Lucida Console

All monospace characters h

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

Example

```
p {  
  font-family: "Times New Roman", Times, serif;  
}  
  
p.normal {  
  font-style: normal;  
}  
  
p.italic {  
  font-style: italic;  
}  
  
p.oblique {  
  font-style: oblique;  
}  
  
h1 {  
  font-size: 40px;  
}  
  
h2 {  
  font-size: 30px;  
}  
  
p {  
  font-size: 14px;  
}
```

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 ``).

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, go to fontawesome.com, sign in, and get a code to add in the `<head>` section of your HTML page:

```
<script src="https://kit.fontawesome.com/yourcode.js"></script>
```

Read more about how to get started with Font Awesome in our [Font Awesome 5 tutorial](#).

Note: No downloading or installation is required!

Example

```
<!DOCTYPE html>
<html>
<head>
<script src="https://kit.fontawesome.com/a076d05399.js"></script>
</head>
<body>

<i class="fas fa-cloud"></i>
<i class="fas fa-heart"></i>
<i class="fas fa-car"></i>
<i class="fas fa-file"></i>
<i class="fas fa-bars"></i>

</body>
</html>
```

Styling Links

Links can be styled with any CSS property (e.g. `color`, `font-family`, `background`, etc.).

Example

```
a {
  color: hotpink;
}
```

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

Example

```
/* 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;
}
```

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:

Example

```
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;  
}
```

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;  
}
```

Collapse Table Borders

The `border-collapse` property sets whether the table borders should be collapsed into a single border:

Example

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

Table Width and Height

Width and height of a table are defined by the `width` and `height` properties.

The example below sets the width of the table to 100%, and the height of the `<th>` elements to 50px:

Example

```
table {  
  width: 100%;  
}
```

```
}  
  
th {  
    height: 50px;  
}
```

Horizontal Alignment

The `text-align` property sets the horizontal alignment (like left, right, or center) of the content in `<th>` or `<td>`.

By default, the content of `<th>` elements are center-aligned and the content of `<td>` elements are left-aligned.

The following example left-aligns the text in `<th>` elements:

```
th {  
    text-align: left;  
}
```

Vertical Alignment

The `vertical-align` property sets the vertical alignment (like top, bottom, or middle) of the content in `<th>` or `<td>`.

By default, the vertical alignment of the content in a table is middle (for both `<th>` and `<td>` elements).

The following example sets the vertical text alignment to bottom for `<td>` elements:

Example

```
td {  
    height: 50px;  
    vertical-align: bottom;  
}
```

Table Padding

To control the space between the border and the content in a table, use the `padding` property on `<td>` and `<th>` elements:

Example

```
th, td {  
  padding: 15px;  
  text-align: left;  
}
```

Responsive Table

A responsive table will display a horizontal scroll bar if the screen is too small to display the full content:

Add a container element (like <div>) with `overflow-x:auto` around the <table> element to make it responsive:

Example

```
<div style="overflow-x:auto;">  
  
<table>  
... table content ...  
</table>  
  
</div>
```

CSS border-image Property

The CSS `border-image` property allows you to specify an image to be used instead of the normal border around an element.

The property has three parts:

1. The image to use as the border
2. Where to slice the image
3. Define whether the middle sections should be repeated or stretched

The `border-image` property takes the image and slices it into nine sections, like a tic-tac-toe board. It then places the corners at the corners, and the middle sections are repeated or stretched as you specify.

Note: For `border-image` to work, the element also needs the `border` property set!

Here, the middle sections of the image are repeated to create the border:

Example

```
#borderimg {  
  border: 10px solid transparent;  
  padding: 15px;  
  border-image: url(YourImage.png) 30 round;  
}
```

CSS Multiple Backgrounds

CSS allows you to add multiple background images for an element, through the `background-image` property.

The different background images are separated by commas, and the images are stacked on top of each other, where the first image is closest to the viewer.

The following example has two background images, the first image is a flower (aligned to the bottom and right) and the second image is a paper background (aligned to the top-left corner):

Example

```
#example1 {  
  background-image: url(YourImage1.gif), url(YourImage2.gif);  
  background-position: right bottom, left top;  
  background-repeat: no-repeat, repeat;  
}
```

he following example uses the `background` shorthand property (same result as example above):

Example

```
#example1 {  
  background: url(YourImage1.gif) right bottom no-repeat,  
  url(Yourimage2.gif) left top repeat;  
}
```

CSS Background Size

The CSS `background-size` property allows you to specify the size of background images.

The size can be specified in lengths, percentages, or by using one of the two keywords: contain or cover.

The following example resizes a background image to much smaller than the original image (using pixels):

Example

```
#div1 {  
    background: url(Yourimage.jpg);  
    background-size: 100px 80px;  
    background-repeat: no-repeat;  
}
```

What is Responsive Web Design?

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

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

It is called responsive web design when you use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen. Don't worry if you don't understand the example below, we will break down the code, step-by-step.

Setting The Viewport

HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag.

You should include the following `<meta>` viewport element in all your web pages:

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

A `<meta>` viewport element gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

What is a Media Query?

Media query is a CSS technique introduced in CSS3.

It uses the `@media` rule to include a block of CSS properties only if a certain condition is true.

Example

If the browser window is 600px or smaller, the background color will be lightblue:

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Background Images

Background images can also respond to resizing and scaling.

Here we will show three different methods:

1. If the `background-size` property is set to "contain", the background image will scale, and try to fit the content area. However, the image will keep its aspect ratio (the proportional relationship between the image's width and height):

Example

```
div {  
  width: 100%;  
  height: 400px;  
  background-image: url('img_flowers.jpg');  
  background-repeat: no-repeat;  
  background-size: contain;  
  border: 1px solid red;  
}
```

EXO 01:

- [Accueil](#)
- [Page 1](#)
- [Page 2](#)
- [Page 3](#)

```
<div class="menu">  
  
  <ul>  
  
    <li><a href="#">Accueil</a></li>  
  
    <li><a href="#">Page 1</a></li>  
  
    <li><a href="#">Page 2</a></li>  
  
    <li><a href="#">Page 3</a></li>  
  
  </ul> </div>
```

Use CSS in order that the page looks like this:



Exo2:

The purpose of this exercise is to carry out the horizontal menu described below and improve the menu design, while ensuring that the code is well structured.



Exo3:

Reproduce the site below using and aligning DIVs:



HTML Links and Images

Links are found in nearly all web pages. Links allow users to click their way from page to page. You can click on a link and jump to another document. When you move the mouse over a link, the mouse arrow will turn into a little hand.

Example

```
<a href="https://www.w3schools.com/javascript/">Visit Our javascript tutorial</a>
```

The example above used an absolute URL (a full web address). A local link (link to the same web site) is specified with a relative URL:

```
<a href="html_images.asp">HTML Images</a>  
<a href="html_images.asp">HTML Images</a>
```

HTML Links - The target Attribute

The `target` attribute specifies where to open the linked document.

The `target` attribute can have one of the following values:

- `_blank` - Opens the linked document in a new window or tab

- `_self` - Opens the linked document in the same window/tab as it was clicked (this is default)
- `_parent` - Opens the linked document in the parent frame
- `_top` - Opens the linked document in the full body of the window
- `framename` - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

Example

```
<a href="https://www.w3schools.com/" target="_blank">Visit  
W3Schools!</a>
```

External pages can be referenced with a full URL or with a path relative to the current web page.

This example uses a full URL to link to a web page:

Example

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

You can change the default colors of a link, by using CSS:

Example

```
<style>  
a:link {  
    color: green;  
    background-color: transparent;  
    text-decoration: none;  
}  
  
a:visited {  
    color: pink;  
    background-color: transparent;  
    text-decoration: none;  
}  
  
a:hover {  
    color: red;  
    background-color: transparent;  
    text-decoration: underline;  
}  
  
a:active {  
    color: yellow;
```

```
background-color: transparent;
text-decoration: underline;
}
</style>
```

A links can also be styled as a button, by using CSS:

Example

```
<style>
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 15px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

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

Images can improve the design and the appearance of a web page.

In HTML, images are defined with the `` tag.

The `` tag is empty, it contains attributes only, and does not have a closing tag.

The `src` attribute specifies the URL (web address) of the image:

```

```

The `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader). The value of the `alt` attribute should describe the image.

Width and Height, or Style?

The `width`, `height`, and `style` attributes are valid in HTML.

However, we suggest using the `style` attribute. It prevents styles sheets from changing the size of images:

Example

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




</body>
</html>
```

If not specified, the browser expects to find the image in the same folder as the web page.

However, it is common to store images in a sub-folder. You must then include the folder name in the `src` attribute:

Example

```

```

Image as a Link

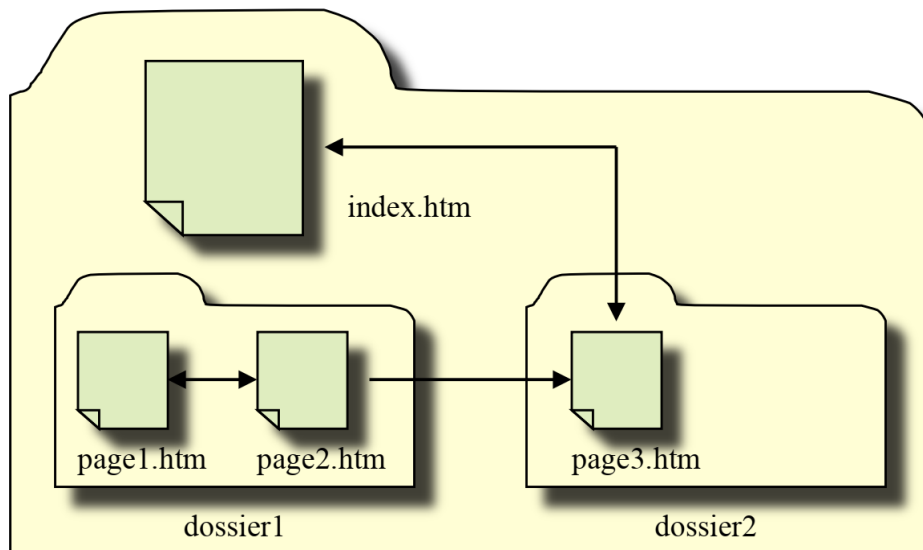
To use an image as a link, put the `` tag inside the `<a>` tag:

Example

```
<a href="default.asp">
  
</a>
```

EX0 5: Create an Exo 5 folder which contains the file index.htm and two folders "folder1" and "folder2".

These contain the files page1.htm, page2.htm and page3.htm respectively, as follows:



Reproduce the pages index.htm, page1.htm, page2.htm and page3.htm as below by creating the following internal links:

1. from page1.htm to page2.htm and a back link,
2. from page2.htm to page3.htm
3. from page index.htm to page page3.htm and a back link

Liens internes en Hypertexte

Le langage HTML permet la navigation sur le Web, à l'aide de liens hypertextes.

Un lien est affecté à un morceau de texte, une image...

Un clic sur [Lien Page 3](#) permet d'aller vers le fichier page3.htm

Index.htm

Liens en Hypertexte

Le langage HTML permet la navigation sur le Web, à l'aide de liens hypertextes.

Un lien est affecté à un morceau de texte, une image...

Un clic sur [ce lien](#) permet de faire un lien vers la page page2.htm située dans le même dossier.

Page 1.htm

Liens internes en Hypertexte

Bien, maintenant que vous y êtes.

Vous pouvez retourner :

- ♦ soit en cliquant sur "back" sur votre navigateur,
- ♦ soit en faisant [un lien](#) dans ce texte-ci.

Un clic sur [ce lien](#) permet de faire un lien vers la page page3.htm située dans l'autre dossier.

Page2.htm

Liens en Hypertexte

Bien, maintenant que vous y êtes.

Vous pouvez retourner :

- ♦ soit en cliquant sur "back" sur votre navigateur,
- ♦ soit en faisant [un lien index](#) dans ce texte-ci.

Page3.htm

Exo 6:

Link to the google search engine and another link to your email address like below:

Les liens sont toujours soulignés, ils servent à naviguer dans votre site mais l'on peut aussi faire un lien vers un autre site web : [Cliquez ici pour accéder au moteur de recherche google](#)

On peut aussi faire un lien vers une adresse électronique comme [moi même](#)

Exo 7: Reproduce this page



Exo 8: Reproduce this page



Ceci est le logo de dauphine avec du texte aligné à gauche du logo. Pour réussir cet effet, utilisez un tableau avec la cellule de gauche de 300 pixels, le texte est en Arial et son alignement est justifié



Ceci est un lien sur le logo de dauphine



Ceci est un lien sans la bordure ajoutée par défaut par le navigateur aux images servant de lien

