CSS3

Contents

- Fundamentals
- Colors, Backgrounds
- Borders, Margins and Padding
- Height and Width
- Box Model

- Text
- Fonts
- Links
- List
- Tables
- Forms

Contents

- Combinators
- Pseudo class
- Pseudo element
- Attribute selectors

- Opacity
- Display
- Visibility
- Specificity
- Max width
- Position
- Float
- Inline-block
- Overflow

Contents

- Gradients
- Shadows
- Text effects
- 2D/3D transforms
- Tool Tips
- Style images

- CSS Button
- Navigation Bar
- Drop Downs
- Image Gallery
- Website Layout
- Flex Box
- CSS Grid
- CSS Responsive

Cascading Style Sheets - Importance

- Let's do a simple experiment…
- Add **pendule**/**web developer toolbar** extensions to your Chrome browser before you start the experiment
- Load www.facebook.com or any other website of your choice
- Try disabling all the CSS styles from the web page you are viewing
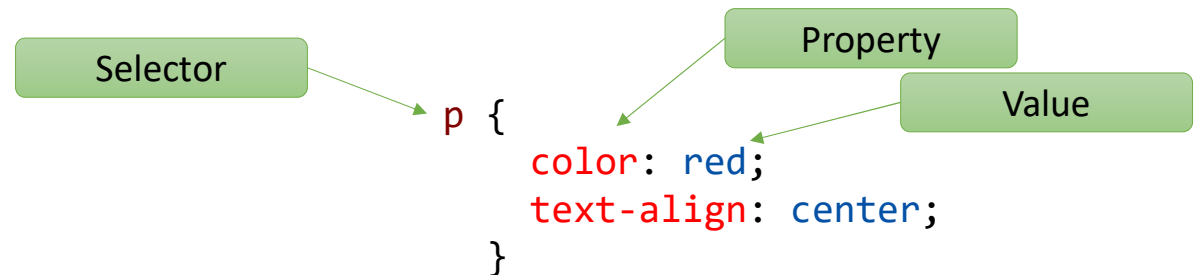- What did you learn??

CSS describes how the HTML elements should be displayed

03_tables.html visit this

Fundamentals

- CSS Syntax
- CSS Comment

```
/* This is a comment */
```

Selector

Property

Value

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

- CSS selectors are used to select the HTML elements to be styled

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element,..	div, p	Selects all <div> elements and all <p> elements

Ways to Add Styles

- Inline CSS
  - Can be used to apply a unique style to a single 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>
```

Ways to Add Styles

- Internal CSS
  - Can be used if one single HTML page has a unique style

```
<!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>
```

Way to Add Styles

- An eternal .css file is maintained

- Every HTML file should reference it

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

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

</body>
</html>
```

style.css

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

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

Specifying Colors

Tomato	Orange	DodgerBlue	MediumSeaGreen
Gray	SlateBlue	Violet	LightGray

Specifying rgb values

rgb(255, 0, 0)
rgb(60, 179, 113)
rgb(255, 165, 0)

Specifying hex values

#0000ff
#ee82ee
#6a5acd

Specifying hsl values

hsl(0, 100%, 50%)
hsl(0, 60%, 50%)
hsl(0, 20%, 50%)

Additionally, **alpha** values can be specified for rgb and hsl values to indicate transparency

Styling Text

- CSS Background Color

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

- CSS Text Color

```
<h1 style="color:Tomato;">Hello World</h1>
```

Backgrounds

- Background color

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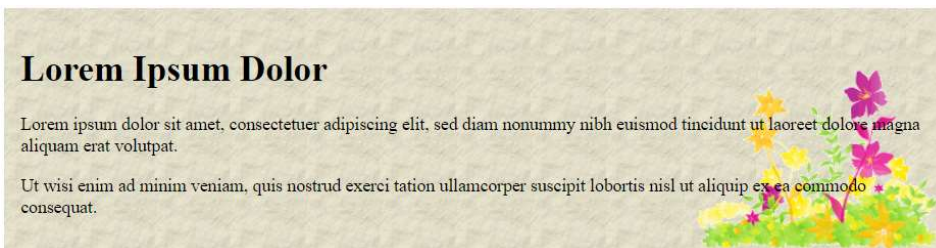
- Background Image

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

Specific values can be found using the in-code help menus

Shorthand → background: #ffffff url("img_tree.png") no-repeat fixed right top;

Multiple Backgrounds

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


Lorem Ipsum Dolor

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.

Borders

- Border Style

    `border-style`: `dotted solid double dashed`;

    `Try with 1, 2 and 3 values only`

- Border Width

    `border-width`: `20px 5px`;

- Border Color

    `border-color`: `red`;
    `border-color`: `red green yellow`;

- Rounded Borders

    `border-radius`: `5px`;

- top border is dotted
- right border is solid
- bottom border is double
- left border is dashed

- 20 px Top and Bottom
- 5 px Left and Right

Margins

- Margins are used to create space around elements
- All the margin and following few properties can have the following values:
    - auto - the browser calculates the margin
    - *length* - margin in px, pt, cm, etc.
    - *%* - margin in % of the width of the containing element
    - inherit - margin should be inherited from the parent element

```
p {
    margin-top: 100px;          Shorthand → margin: 25px 50px 75px 100px;
    margin-bottom: 100px;
    margin-right: 150px;
    margin-left: 80px;
}
```

Padding

- The CSS Padding properties are used to generate space around elements content inside of any defined borders

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

Height and Width

- Height and Width properties are used to set the height and width of an element

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

The Box Model

- All HTML elements can considered as boxes.
- Box models refers to specifying margins, border and padding within which actual content is placed

```
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

Text

- Several properties can be applied to text:

```css
h1 {
    background-color: black;
    color: white;
    text-align: center;
    direction: rtl;
    vertical-align: top;
    text-decoration: underline;
    text-transform: capitalize;
    text-indent: 50px;
    letter-spacing: 3px;
    line-height: 0.8;
    word-spacing: 10px;
    white-space: nowrap;
    text-shadow: 2px 2px 5px red;
}
```

In-line help in IDE gives you more options

Font

- Font family, boldness, size and style can be specified using the following attributes
- The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems.
- If the browser does not support the first font, it tries the next font.

```css
h1{
    font-family: "Times New Roman", Times, serif;
    font-style: normal;
    font-weight: normal;
    font-variant: small-caps;
    font-size: 40px;
}
```

Google Fonts

```html
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<style>
body {
  font-family: "Sofia";
  font-size: 22px;
}
</style>
</head>
<body>

<h1>Sofia Font</h1>
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.</p>

</body>
</html>
```

Google fonts has 900+ choices. Create a stylesheet link and refer to a font family of your choice

Links

- Links can be styled using **color**, **font-family**, **background**, etc.,

- Also, they can be styled differently depending on their state

- 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 hovers over
  - **a:active** - a link the moment it is clicked

Follow the same sequence to be effective

```css
/* unvisited link */
a:link {
  color: red;
  text-decoration: none;
}
```

Removes Underline

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

```css
/* mouse over link */
a:hover {
  color: hotpink;
  background-color: lightgreen;
}
```

Changes background on hover

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

Other link text properties can be changed as well

Link Button

Link Button

A link styled as a button:

This is a link

```
<!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>

<h2>Link Button</h2>
<p>A link styled as a button:</p>
<a href="default.asp" target="_blank">This is a link</a>

</body>
</html>
```

Lists

- The table shows all the list properties that can be set

Property	Description
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies the position of the list-item markers (bullet points)
list-style-type	Specifies the type of list-item marker; none value with remove default settings

- 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;
}
```

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

Tables

```
table {
    border: 1px solid black;
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

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

Firstname	Lastname
Peter	Griffin
Lois	Griffin

```
table {
    border-collapse: collapse;
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Width, Height and Alignment of Lists

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

th {
    height: 50px;
}




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

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




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

Specifying horizontal dividers

```css
th, td {
    border-bottom: 1px solid #ddd;
}
```

Hover-able and Striped Lists

- Use the :hover selector on <tr> to highlight table rows on mouse over

```
tr:hover {background-color: #f5f5f5;}
```

- For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

- Table colors can be set using the regular attributes for text and color discussed earlier

```
th {
    background-color: #4CAF50;
    color: white;
}
```

Responsive Table

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

```
<div style="overflow-x:auto;">

<table>
... table content ...
</table>

</div>
```

Forms – Styling Input Fields

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

```css
input { width: 100%; }
```

- For specific input types, you can use attribute selectors

```css
input[type=text][type=password] { width: 100%; }
```

- Padding and margins can also be used

```css
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
}
```

Styling Input Fields

- Inputs can be styles with borders along with style, size and color

```css
input[type=text] {
  border: 2px solid red;
  border-radius: 4px;
}
```

- If only a bottom border is required, use the border-bottom property

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

Styling Input Fields

- Text and background color

```css
input[type=text] {
   background-color: #3CBC8D;
   color: white;
}
```

- Use the :focus selector to do something with the input field when it gets focus

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

- **outline:none** can be used to remove the automatic highlighting by some browsers

Styling Input Fields

- Input with icon/image can be done using the `background-image` property positioned using the `background-position` property

```css
input[type=text] {
    background-color: white;
    background-image: url('searchicon.png');
    background-position: 10px 10px;
    background-repeat: no-repeat;
    padding-left: 40px;
}
```

Styling Input Fields

- Animated search input field can be created using the **transition** property

Before Focus

Q Search..

After Focus

```
input[type=text] {
    transition: width 0.4s ease-in-out;
}

input[type=text]:focus {
    width: 100%;
}
```

Animated search input:

Q Search..

bhavishya puran

Styling Text Areas, Select Menus, Buttons

```css
textarea {
    width: 100%;
    height: 150px;
    padding: 12px 20px;
    box-sizing: border-box;
    border: 2px solid #ccc;
    border-radius: 4px;
    background-color: #f8f8f8;
    resize: none;
}
```

Prevents textarea from being resized

```css
select {
    width: 100%;
    padding: 16px 20px;
    border: none;
    border-radius: 4px;
    background-color: #f1f1f1;
}

input[type=button], input[type=submit],
input[type=reset] {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 16px 32px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
}
```

4

Combinators

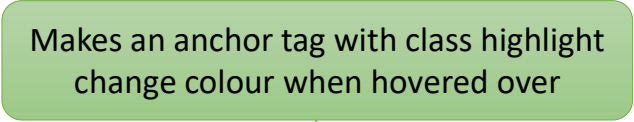| Selector	Example	Example description
element element	div p	Selects all <p> elements inside <div> elements
element>element	div > p	Selects all <p> elements where the parent is a <div> element
element+element	div + p	Selects all <p> elements that are placed immediately after <div> elements
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element

```css
/* example */
div ~ p {
  background-color: yellow;
}
```

Pseudo-Class

- A pseudo-class is used to define a special state of an element.

- For example, it can be used to:
  - Style an element when a user mouses over it
  - Style visited and unvisited links differently
  - Style an element when it gets focus

```
/* example */
selector:pseudo-class {
  property: value;
}
```

Makes an anchor tag with class highlight change colour when hovered over

```html
<!DOCTYPE html>
<html>
<head>
<style>
a.highlight:hover {color: #ff0000;}
</style>
</head>
<body>
<p><a class="highlight" href="css_syntax.asp">
CSS Syntax</a></p>
<p><a href="default.asp">CSS Tutorial</a></p>
</body>
</html>
```

CSS Pseudo Class Summary

Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent

CSS Pseudo Class Summary

Selector	Example	Example Description
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects <input> elements with no "required" attribute
:out-of-range	input:out-of-range	Selects <input> elements with a value outside a specified range
:read-only	input:read-only	Selects <input> elements with a "readonly" attribute specified
:read-write	input:read-write	Selects <input> elements with no "readonly" attribute
:required	input:required	Selects <input> elements with a "required" attribute specified
:root	root	Selects the document's root element

CSS Pseudo Class Summary

Selector	Example	Example Description
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all <input> elements with a valid value
:visited	a:visited	Selects all visited links

```
<style>
#news1:target {
  border: 2px solid #D4D4D4;
  background-color: #e5eecc;
}
</style>

<body>
<h1>This is a heading</h1>
<p><a href="#news1">Jump to New content 1</a></p>
<p><a href="#news2">Jump to New content 2</a></p>

<p>Click on the links above and the :target selector highlight the current active HT
ML anchor.</p>

<p id="news1"><b>New content 1...</b></p>
<p id="news2"><b>New content 2...</b></p>
</body>
```

```
:target {
  border: 2px solid #D4D4D4;
  background-color: #e5eecc;
}
```

This will target both anchors

Pseudo-Elements

- A CSS pseudo-element is used to style specified parts of an element.
- For example, it can be used to:
  - Style the first letter, or line, of an element
  - Insert content before, or after, the content of an element

Pseudo-Elements

Selector	Example	Example description
::after	p::after	Insert content after every <p> element
::before	p::before	Insert content before every <p> element
::first-letter	p::first-letter	Selects the first letter of every <p> element, only block level elements
::first-line	p::first-line	Selects the first line of every <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user

```
p::after {
  content: " - Remember this";
}

<p>My name is Donald</p>
<p>I live in  Johannesburg </p>
…
```

My name is Donald – Remember This
I live in Johannesburg – Remember This

```
<style>

::selection {
  color: red;
  background: yellow;
}
</style>
<body>

<h1>Select some text on this page:</h1>

<p>This is a paragraph.</p>
<div>This is some text in a div element.</div>

</body>
```

Select some text on this page:

This is a paragraph.

This is some text in a div element.

Attribute Selectors

Selector	Example	Example description
[attribute]	[target]	Selects all elements with a target attribute
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[attribute$=value]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[attribute*=value]	a[href*="select"]	Selects every <a> element whose href attribute value contains the substring "select"

```css
a[target] {
  background-color: yellow;
}
```

```css
a[target="_blank"] {
  background-color: yellow;
}
```

5

Opacity

- Opacity defines the transparency of the element
- Opacity can take value from 0.0 to 1.0
- Opacity is often used together with :hover selector to change opacity on mouse-over

```css
img {
    opacity: 0.5;
}

img:hover {
    opacity: 1.0;
}
```

Display

- Block level elements starts on a new line and takes up the full width
  - <div>, <h1>..<h2>, <p>, <form>, <header>, <footer>, <section>
- Inline elements does not start on a new line and only takes up as much width as necessary
  - <span>,<a>,<img>
- Display property specifies if/how an element is displayed

```
li { display: inline; }
```
Causes <li> element to be displayed horizontally

```
span { display: block; }
```
Causes <span> element to be a block

```
h1.hidden { display: none; }
```
Causes <h1> with hidden class to be invisible

Visibility

- **visibility:hidden** hides the element, but it still takes up space in the layout.
- **display:none** removes the element from the document. It does not take up any space.

Max Width

- Using **max-width** will improve the browser's handling of small windows. This is important when making a site usable on small devices

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

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

This div element has width: 500p

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!

Float

- The CSS float property specifies how an element should float.

```
<style>
img {
  float: right;
}
</style>

<p><img src="pineapple.jpg" alt="Pineapple" style="width:170px;height:170px;
margin-left:15px;">
The text in this example will wrap around the image. The image will float to
 the right. Float can be used to arrange elements in a block.
</p>
```

The text in this example will wrap around the image. The image will float to the right. Float can be used to arrange elements in a block.

Style Images

```css
img {
    border: 1px solid #ddd;
    border-radius: 4px;
    padding: 5px;
    width: 150px;
}

img:hover {
    box-shadow: 0 0 2px 1px rgba(0, 140, 186, 0.5);
}
```

```html
<a href="paris.jpg">
    <img src="paris.jpg" alt="Paris">
</a>
```

```css
img {
    width: 200px;
    height: 400px;
    object-fit: cover;
}
```

Cuts off the sides of image, preserves aspect ratio and fits the space

```css
img {
    border-radius: 50%;
}
```

Style Images - Polaroids

```css
div.polaroid {
    width: 80%;                                          Makes it responsive
    background-color: white;
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
    margin-bottom: 25px;
}


div.container {
    text-align: center;
    padding: 10px 20px;
}
```

```html
<div class="polaroid">
  <img src="img_5terre.jpg" alt="5 Terre" style="width:100%">
  <div class="container">
  <p>Cinque Terre</p>
  </div>
</div>
```

Navigation Bar

- Navigation Bar is a list of links

```
<ul>
  <li><a href="default.asp">Home</a></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
  <li><a href="about.asp">About</a></li>
</ul>
```

```css
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #555;
  color: white;
}
```
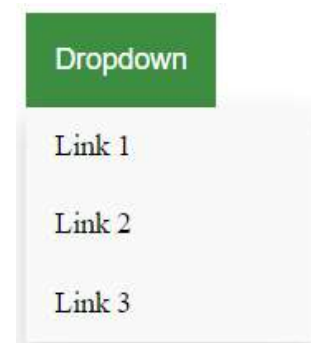
```css
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: Black;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

li a:hover {
  background-color: Green;
}
```

| Home | News | Contact | About |

Drop downs

- Dropdown menus can be created employing some of the CSS styling

```html
<div class="dropdown">
  <button class="dropbtn">Dropdown</button>
  <div class="dropdown-content">
  <a href="#">Link 1</a>
  <a href="#">Link 2</a>
  <a href="#">Link 3</a>
  </div>
</div>
```

```css
.dropbtn {
  background-color: #4CAF50;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;
}

.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content {
  display: block;
}

.dropdown:hover .dropbtn {
  background-color: #3e8e41;
}
```

Dropdown Image

```html
<div class="dropdown">
  <img src="img_5terre.jpg" alt="Cinque Terre" width="100" height="50">
  <div class="dropdown-content">
  <img src="img_5terre.jpg" alt="Cinque Terre" width="300" height="200">
  <div class="desc">Beautiful Cinque Terre</div>
  </div>
</div>
```

```css
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown:hover .dropdown-content {
  display: block;
}

.desc {
  padding: 15px;
  text-align: center;
}
```

Image Gallery

Image 1

Image 2

Image 3

Image Gallery - HTML

```html
<div class="gallery">
  <a target="_blank" href="img_5terre.jpg">
    <img src="img_5terre.jpg" alt="Cinque Terre" width="600" height="400">
  </a>
  <div class="desc">Image 1</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_forest.jpg">
    <img src="img_forest.jpg" alt="Forest" width="600" height="400">
  </a>
  <div class="desc">Image 2</div>
</div>

<div class="gallery">
  <a target="_blank" href="img_lights.jpg">
    <img src="img_lights.jpg" alt="Northern Lights" width="600" height="400">
  </a>
  <div class="desc">Image 3</div>
</div>
```

Image Gallery - CSS

```css
div.gallery {
  margin: 5px;
  border: 1px solid #ccc;
  float: left;
  width: 180px;
}

div.gallery:hover {
  border: 1px solid #777;
}

div.gallery img {
  width: 100%;
  height: auto;
}

div.desc {
  padding: 15px;
  text-align: center;
}
```

Website Layout

- A website is generally divided into headers, menus, content and a footer:

Header
Navigation Menu

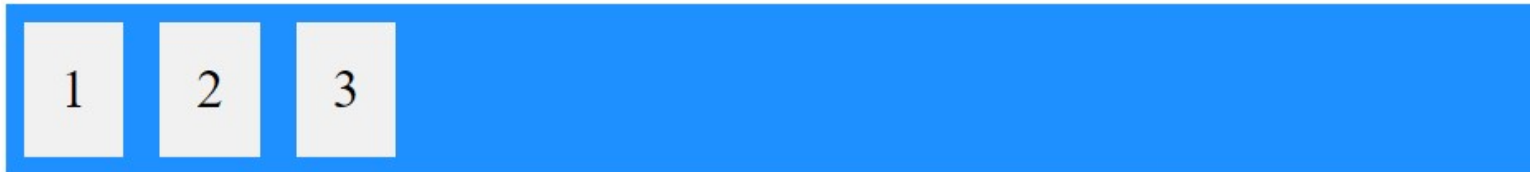| Content	Main Content	Content

Footer

Flex Box

- The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning

- First, define a flex container

- Next, set the flex parameters

```html
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

```css
.flex-container {
    display: flex;
    background-color: DodgerBlue;
}

.flex-container > div {
    background-color: #f1f1f1;
    margin: 10px;
    padding: 20px;
    font-size: 30px;
}
```

CSS Grid

- The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning
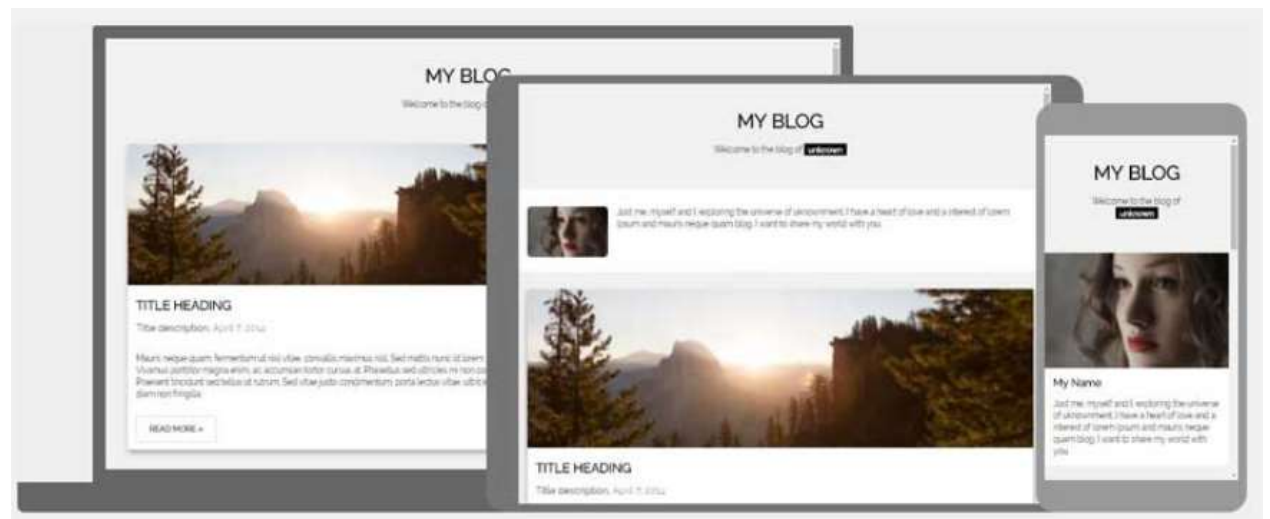
Header		
Menu	Main	Right
	Footer	

```html
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

```css
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
```

1	2	3
4	5	6
7	8	9

CSS Responsive

- Needed to create the best experience for all users
  - 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.

CSS Responsive

- Set the viewport
  - Viewport is the users visible area of the webpage

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

- Use the grid view
  - Pages are divided into columns
  - A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window

- Use media queries

- Adopt the "Mobile First" approach

https://www.w3schools.com/css/css_rwd_mediaqueries.asp : use example under Design for Mobile First