



# Website Design (Manual) CSS

Developed by Alabian Solutions Ltd

---

# Chapter 1

---

## Introduction to CSS

CSS stands for Cascading Style Sheets, CSS allows you to create rules that specify how the content of an element should appear. For example, you can specify that the background of the page is red, all paragraphs should appear in green using the Arial typeface, or that all level one headings should be in a blue, italic, Times typeface.

### CSS Solved a Big Problem

HTML was created to *describe the content* of a web page, like:

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

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

When tags like `<font>`, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

### CSS Saves a Lot of Work!

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!

## Anatomy of CSS

CSS consists of "**style rules**". Each style rule consists of a "**selector**" and "**declarations**" of property-value pairs:



CSS declarations sit inside curly brackets and each is made up of two parts: a property and a value, separated by a colon. You can specify several properties in one declaration, each separated by a semi-colon.

## Where to put CSS

- Internal CSS (embedded)
- Inline CSS
- External CSS (link)

## Internal CSS

An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the **<style>** element, inside the **<head>** section of an HTML page:

**code:**

```
<head>
  <meta charset="UTF-8">
  <title>Internal CSS</title>
  <style type="text/css">
    body{
      background-color:black;
      color:white;
    }

    h1{
      color:red;
      text-transform: uppercase;
      font-weight:bold;
    }

    h2{
      color:yellow;
      font-style:italic;
    }
  </style>
</head>
<body>

  <h1>Welcome to our Website</h1>

  <h2>Lionel Messi is the World's best player</h2>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quas
    asperiores unde ratione accusantium voluptatibus molestiae ut
    doloribus tempora eveniet perferendis, incidunt, modi ipsa omnis
    delectus vitae. Omnis iste alias suscipit.
  </p>

</body>
```

result:

# WELCOME TO OUR WEBSITE

*Lionel Messi is the World's best player*

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quas asperiores unde ratione accusantium voluptatibus molestiae ut doloribus tempora eveniet perferendis, incidunt, modi ipsa omnis delectus vitae. Omnis iste alias suscipit.

## Inline CSS

CSS styles can be assigned to individual HTML elements via the style attribute:

code:

```
<body style="background-color:black; color:white">

    <h1 style="color:red; text-transform:uppercase;">Welcome to our
    Website
    </h1>

    <h2 style="color:yellow; font-style:italic;">Lionel Messi is the
    World's best player</h2>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quas
        asperiores unde ratione accusantium
    </p>
</body>
```

result:

# WELCOME TO OUR WEBSITE

*Lionel Messi is the World's best player*

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quas asperiores unde ratione accusantium

## External

The style sheet is stored in another file, and the file is usually saved with **.css** file extension. This file is incorporated into your html document via **<link>** element. This makes it possible to change the look of an entire website by changing just one file. The **<link>** element has some attribute that is worth mentioning

### href

This specifies the path to the CSS file (which is often placed in a folder called css or styles).

### type

This attribute specifies the type of document being linked to. The value should be **text/css**.

### rel

This specifies the relationship between the HTML page and the file it is linked to. The value should be **stylesheet** when linking to a CSS file.

### html:

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>External</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>Welcome to our Website</h1>

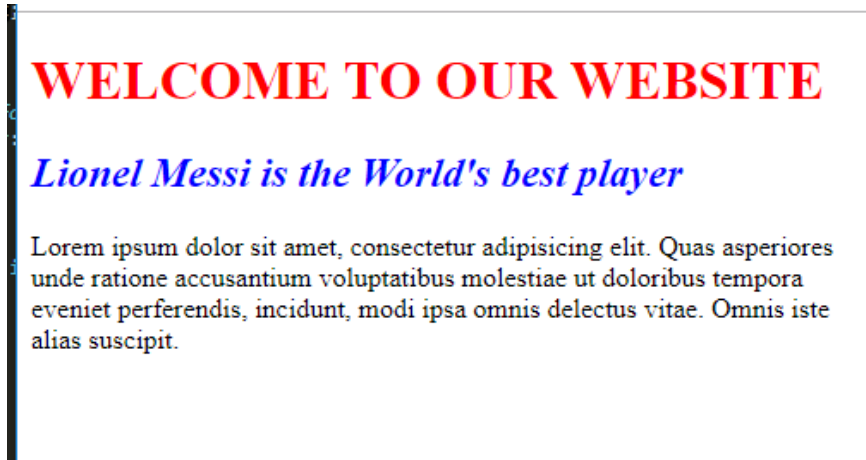
  <h2>Lionel Messi is the World's best player</h2>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quas
    asperiores unde ratione accusantium voluptatibus molestiae ut
    doloribus tempora eveniet perferendis, incidunt, modi ipsa omnis
    delectus vitae. Omnis iste alias suscipit.
  </p>
</body>
</html>
```

### CSS

```
body{
  background-color:black;
  color:white;
}
h1{
  color:red;
  text-transform: uppercase;
  font-weight:bold;
}
h2{
```

```
color:yellow;
font-style:italic;
}
```

**result:**



## CSS Selectors

These are ways of selecting HTML element so you can apply a CSS styles to them.

### Types of CSS selectors

1. Type Selector
2. Class Selector
3. Id Selector
4. Child Selector
5. Descendant Selector
6. Adjacent Sibling Selector
7. General Sibling Selector
8. Add attribute selector
9. Attribute value selector
10. Universal Selector

### Type Selector

The type selector is way of targeting an element based on the element type.

**html:**

```
<body>
  <h2>Heading 1</h2>
  <h2>Heading 2</h2>
  <h2>Heading 3</h2>
  <h2>Heading 4</h2>
  <h2>Heading 5</h2>
```

```
</body>
```

#### CSS:

```
body{
    background-color:black;
    color:white;
}

h2{
    color:yellow;
    text-transform:uppercase;
}
```

#### result:

---

**HEADING 1**

**HEADING 2**

**HEADING 3**

**HEADING 4**

**HEADING 5**

## Class Selector

The class selector is way of grouping elements that are related with the same value of their class attribute so you can style them in CSS. We use period symbol (.) to target the value of a class in css.

#### html:

```
<body>

    <h2 class="group1">Heading 1</h2>
    <h2 class="group1">Heading 2</h2>
    <h2 class="group1">Heading 3</h2>
    <h2 class="group2">Heading 4</h2>
    <h2 class="group2">Heading 5</h2>

</body>
```

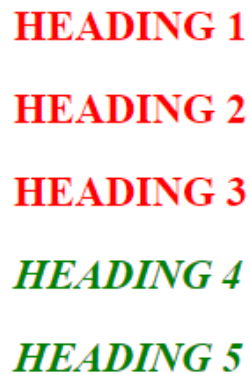
#### code:

```
.group1{
    color:red;
    text-transform:uppercase;
}
```



```
}
```

result:



## Id Selector

The Id selector is a way of giving an element a unique style from every other element in the page. It is important to note that no two element should have the same value of an Id attribute in a page. We use harsh or pound (#) symbol to target an Id in CSS.

html:

```
<body>

    <h2 class="group1">Heading 1</h2>
    <h2 class="group1">Heading 2</h2>
    <h2 class="group1">Heading 3</h2>
    <h2 id="group2">Heading 4</h2>
    <h2 class="group1">Heading 5</h2>

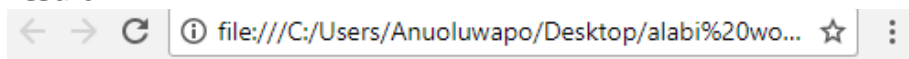
</body>
```

CSS:

```
#group2{
    height:25px;
    width:50px;
    text-transform:uppercase;
}
```



result:



**HEADING 1**

**HEADING 2**

**HEADING 3**

*HEADING 4*

**HEADING 5**

## Child Selector

The Child Selector is way of targeting an element that is the direct child of another element using greater than sign (>).

html:

```
<body>

  <div class="post">
    <p>
      Lorem Ipsum is simply dummy text of the printing and
      typesetting industry. Lorem Ipsum has been the
      industry's standard dummy text ever since the 1500s, when
      an unknown printer took a galley of type and scrambled it
      to make a type specimen book.
    </p>
  </div>
</body>
```

CSS:

```
.post > p{
  background-color: blue;
  color:white;
}
```

## result:

---

*Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.*

## Descendant Selector

Matches an element that is a descendent of another specified element (not just a direct child of that element).

### html:

```
<body>

  <div class="content">
    <p>
      Lorem Ipsum is simply dummy text of the printing and
      typesetting industry. Lorem Ipsum <span>alabian</span>
      has been the industry's standard dummy text ever since the
      1500s, when an unknown printer took a galley of type and
      scrambled it to make a type specimen book.
    </p>
  </div>

</body>
```

### CSS:

```
.content span{
  color:blue;
  text-transform:uppercase;
}
```

## result:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum **ALABIAN** has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

## Adjacent Sibling Selector

Matches an element that is the next sibling of another using the plus symbol (+). The code below will target the first element after the `<h1>` element (but not other `<p>` element).

### html:

```
<body>
```

```
  <h1>Heading 1</h1>
```

```
  <p>
```

```
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio  
    soluta explicabo officia corrupti rerum facilis cumque ea iure  
    architecto libero incidunt nulla similique tempore aperiam, odio  
    possimus labore esse, aspernatur.
```

```
  </p>
```

```
  <p>
```

```
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio  
    soluta explicabo officia corrupti rerum facilis cumque ea iure  
    architecto libero incidunt nulla similique tempore aperiam, odio  
    possimus labore esse, aspernatur.
```

```
  </p>
```

```
  <p>
```

```
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio  
    soluta explicabo officia corrupti rerum facilis cumque ea iure  
    architecto libero incidunt nulla similique tempore aperiam, odio  
    possimus labore esse, aspernatur.
```

```
  </p>
```

```
  <p>
```

```
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio  
    soluta explicabo officia corrupti rerum facilis cumque ea iure  
    architecto libero incidunt nulla similique tempore aperiam, odio  
    possimus labore esse, aspernatur.
```

```
  </p>
```

```
</body>
```

### CSS:

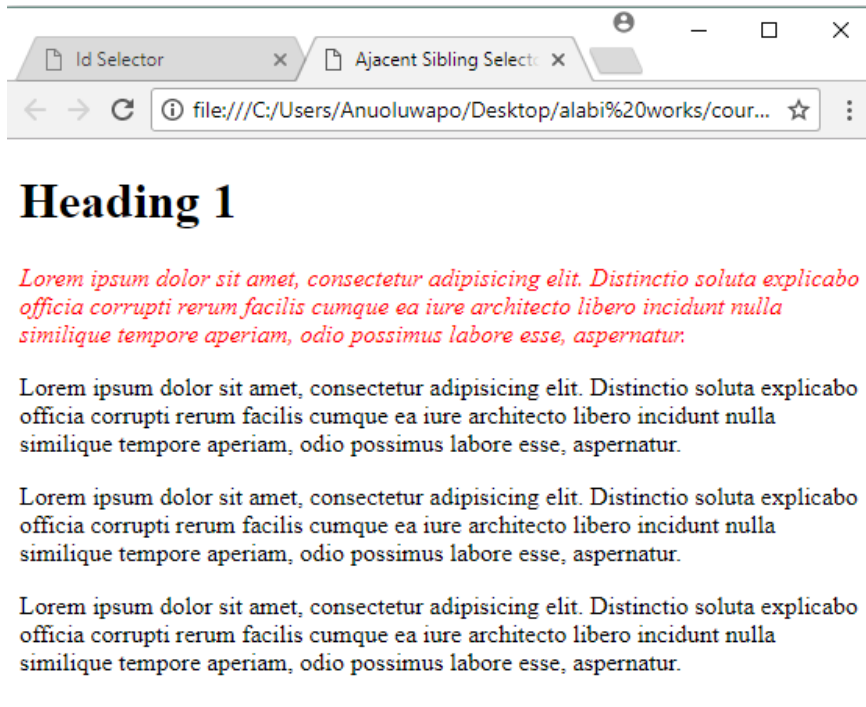
```
h1 + p{
```

```
    color:red;
```

```
    font-style:italic;
```

```
}
```

### result:



## General Sibling Selector

This type of selector matches an element that is a sibling of another using the tilde symbol (~), although it does not have to be the directly preceding element.

### html:

<body>

<h1>Heading 1</h1>

<p>

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio soluta explicabo officia corrupti rerum facilis cumque ea iure architecto libero incidunt nulla similique tempore aperiam, odio possimus labore esse, aspernatur.

</p>

<p>

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio soluta explicabo officia corrupti rerum facilis cumque ea iure architecto libero incidunt nulla similique tempore aperiam, odio possimus labore esse, aspernatur.

</p>

<p>

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio soluta explicabo officia corrupti rerum facilis cumque ea iure architecto libero incidunt nulla similique tempore aperiam, odio possimus labore esse, aspernatur.

```

</p>
<p>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio
  soluta explicabo officia corrupti rerum facilis cumque ea iure
  architecto libero incidunt nulla similique tempore aperiam, odio
  possimus labore esse, aspernatur.
</p>

```

</body>

**CSS:**

```

h1 ~ p{
    color:red;
    font-style:italic;
    width:500px;
}

```

**result:**

## Heading 1

*Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio soluta explicabo officia corrupti rerum facilis cumque ea iure architecto libero incidunt nulla similique tempore aperiam, odio possimus labore esse, aspernatur.*

*Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio soluta explicabo officia corrupti rerum facilis cumque ea iure architecto libero incidunt nulla similique tempore aperiam, odio possimus labore esse, aspernatur.*

*Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio soluta explicabo officia corrupti rerum facilis cumque ea iure architecto libero incidunt nulla similique tempore aperiam, odio possimus labore esse, aspernatur.*

*Lorem ipsum dolor sit amet, consectetur adipisicing elit. Distinctio soluta explicabo officia corrupti rerum facilis cumque ea iure architecto libero incidunt nulla similique tempore aperiam, odio possimus labore esse, aspernatur.*

## Attribute Selector

The **[attribute]** selector is used to select elements with a specified attribute. The following example selects all <a> elements with a target attribute:

**html:**

```

<body>
  <a href="#" target="_blank">Facebook</a>
</body>

```

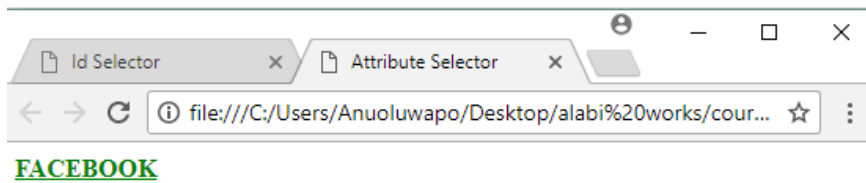
**CSS:**

```

a[target]{
    color:green;
    text-transform: uppercase;
    font-weight: bold;
}

```

result:



## Attribute Value Selector

The `[attribute="value"]` selector is used to select elements with a specified attribute and value.

The following example will select all input tag whose type attribute have the value of text:

html:

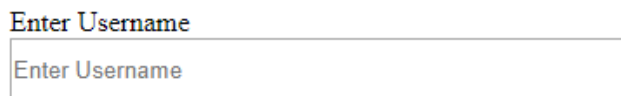
```
<body>
    Enter Username<br>
    <input type="text" name="username" placeholder="Enter Username">

</body>
```

CSS:

```
input[type="text"] {
    width: 350px;
    height: 30px;
}
```

result:



## Universal Selector

The Universal Selector is a way of targeting every single element in a page, using the star selector (\*)

html:

```
<body>
    <h1>Welcome to our Website</h1>

    <h2>Here are the things we do</h2>

    <ul>
        <li>Web Design</li>
        <li>Web Development</li>
        <li>Web Online and Digital Marketing</li>
        <li>Computer Graphics</li>
    </ul>

    <p>
```

```

        Lorem ipsum dolor sit amet, consectetur adipisicing elit.
        eius ducimus sint enim aut quos odio. Explicabo,
    </p>
</body>

```

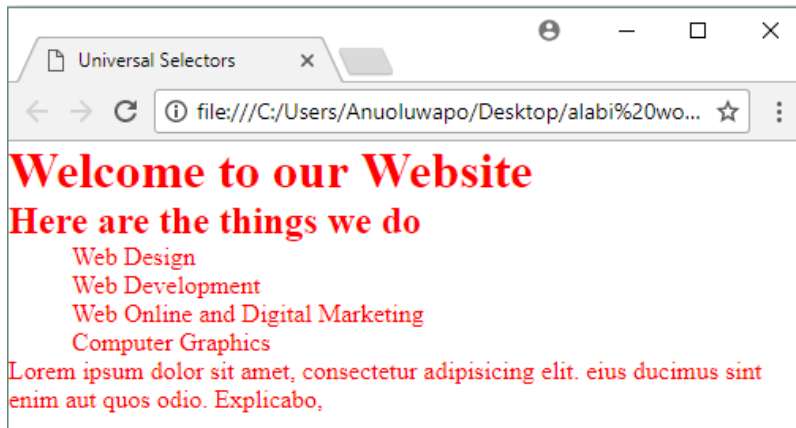
#### CSS:

```

*{
    color:red;
}

```

#### result:



## How CSS Rule

If there are two or more rules that apply to the same element, it is important to understand which will take precedence.

### Last Rule

If the two selectors are identical, the latter of the two will take precedence.

#### html:

```

<body>
    <h1>Heading 1</h1>
    <h1>Heading 2</h1>
</body>

```

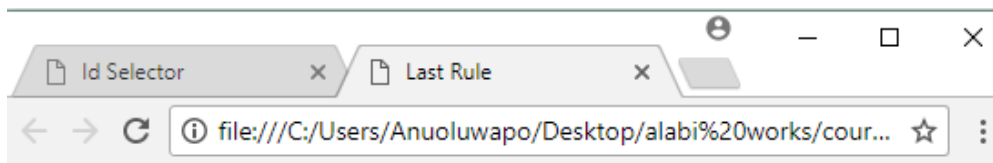
#### CSS:

```

h1{
    color:green;
}
h1{
    color:red;
}

```

#### result:



# Heading 1

## Heading 2

### Specificity

If one selector is more specific than the others, the more specific rule will take precedence over more general ones.

#### html:

```
<body>
  <div id="main">
    <div class="post">
      <h2>Heading </h2>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit.
        Quaerat dolore, alias, neque et est necessitatibus ducimus
        officiis <span>repellendus praesentium doloremque eius in
        earum</span> placeat adipisci qui fugit incidunt! Vitae,
        quia!
      </p>
    </div>
  </div>
</body>
```

#### CSS:

```
.post span{
  color:green;
  font-weight: bold;
}

#main .post span{
  color:red;
  font-weight:bold;
}
```

#### result:

### Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quaerat dolore, alias, neque et est necessitatibus ducimus officiis **repellendus praesentium doloremque eius in earum** placeat adipisci qui fugit incidunt! Vitae, quia!



## Important

You can add **!important** after any property value to indicate that it should be considered more important than other rules that apply to the same element.

### html:

```
<div id="main">
  <div class="post">
    <h2>Heading </h2>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit.
      Quaerat dolore, alias, neque et est necessitatibus ducimus
      officiis
      <span>repellendus praesentium doloremque eius in
      earum</span>
      placeat adipisci qui fugit incidunt! Vitae, quia!
    </p>
  </div>
</div>
```

### CSS:

```
.post span{
    color:red !important;
    font-weight: bold;
}

.post span{
    color:green;
    font-weight:bold;
}
```

### result:

---

## Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quaerat dolore, alias, neque et est necessitatibus ducimus officiis **repellendus praesentium doloremque eius in earum** placeat adipisci qui fugit incidunt! Vitae, quia!

## Naming convention

Some rules to follow when making IDs and class names:

- Describe the content, not the presentation ("warning", not "redbox").
- Use all lowercase, and hyphens when needed for readability ("header-info", not "headerInfo").
- Use hyphens to show that a class or ID is part of something else. (e.g. "footer ", "footer -copyright", and "footer - logo").

## Comments

Comments will be ignored by the browser and are useful for documenting your styles to other humans or commenting out rules. The example below shows how to make a comment in CSS.

**code:**

```
/* This is single line comment in css */  
/*  
    This is how to make  
    multiline line comment  
    in CSS  
*/
```

# Chapter 2

## Pseudo classes and Pseudo elements

### Pseudo classes

Pseudo class selectors are CSS selectors with a colon preceding them. A pseudo-class is similar to a class in HTML, but it's not specified explicitly in the markup. Examples of Pseudo classes are **:link**, **:visited**, **:hover**, **:focus**, **:active**, **:first-child**, **:last-child**, **nth-child(even)**, **nth-child(odd)**.

#### **:link**

This allows you to set styles for links that have not yet been visited.

##### html:

```
<body>
  <ul class="menu">
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Blog</a></li>
    <li><a href="#">Services</a></li>
  </ul>
</body>
```

##### CSS:

```
.menu li a:link{
    color:red;
    font-weight:bold;
    text-decoration:none;
}
```

##### result:

- 
- **Home**
  - **About**
  - **Blog**
  - **Services**

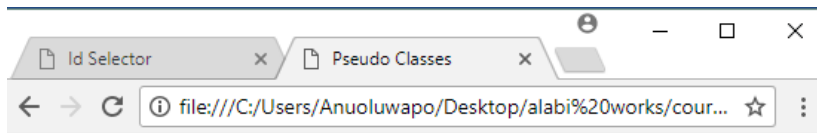
#### **:visited**

This allows you to set styles for links that have been clicked on. They are commonly used to control colors of the links and also whether they are to appear underlined or not. Using the above html code.

##### CSS:

```
.menu li a:visited{
    color:red;
}
```

## result:



- Home
- About
- Blog
- Services

## :hover

This is applied when a user hovers over an element with a pointing device such as a mouse. This has commonly been used to change the appearance of links and buttons when a user places their cursor over them. Using the above html code.

### CSS:

```
.menu li a:hover{
    color:blue;
    font-style:italic;
}
```

## Result:

- Home
- *About*
- Blog
- Services

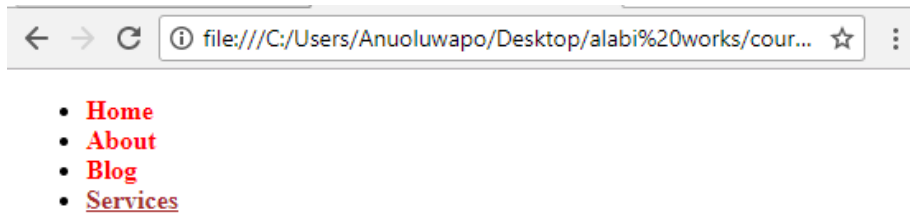
## :focus

This is applied when an element has focus. Any element that you can interact with, such as a link you can click on or any form control can have focus. Using the above html code.

### CSS:

```
.menu li a:focus{
    text-decoration:underline;
    color:brown;
}
```

result:



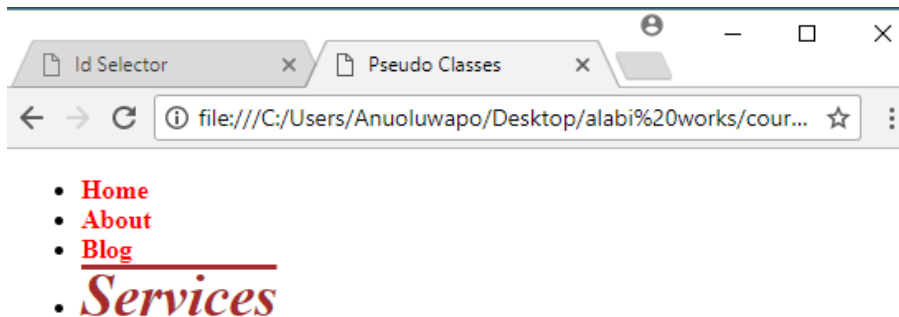
## :active

This is applied when an element is being activated by a user; for example, when a button is being pressed or a link being clicked. Using the above html code.

css:

```
.menu li a:active{
    font-size:35px;
    text-decoration:underline;
}
```

result:



## :first-child

The :first-child pseudo-class matches a specified element that is the first child of another element.

code:

```
.menu li:first-child{
    color:red;
    font-weight:bold;
}
```

result:



## :last-child

The :last-child pseudo-class matches a specified element that is the last child of another element.

### code:

```
.menu li:last-child{
    color:red;
    font-weight:bold;
}
```

### result:

---

- Home
- About
- Blog
- **Contact**

## :nth-child

This has the ability to do things like select even/odd elements.

### code:

```
.menu li:nth-child(even){
    color:red;
    font-weight:bold;
}
```

### result:

- Home
- **About**
- Blog
- **Services**
- Products
- **Contact**

## Pseudo-elements

Pseudo elements are CSS selectors with a two colon preceding them. They are used to style specified parts of an element. Just as pseudo-class is not explicitly defined in the HTML markup.

### ::first-letter

The *::first-letter* pseudo-element is used to add a special style to the first letter of a text. The *::first-letter* pseudo-element can only be applied to block-level elements.

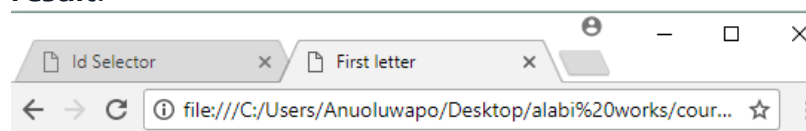
#### html:

```
<body>
  <div class="post">
    <h2>Heading 1</h2>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit.
      Quisquam expedita ut officiis repudiandae sint, earum numquam
      magnam vel architecto autem, sequi omnis facere, consequuntur
      vero non iusto. Architecto, est, accusamus?Lorem ipsum dolor sit
      amet, consectetur adipisicing elit. Quisquam expedita ut officiis
      repudiandae sint, earum numquam magnam vel architecto autem,
      sequi omnis facere, consequuntur vero non iusto. Architecto, est,
      accusamus?
    </p>
  </div>
</body>
```

#### CSS:

```
.post p::first-letter{
  font-size:35px;
  color:red;
  font-style:italic;
}
```

#### result:



### Heading 1

*L*orem ipsum dolor sit amet, consectetur adipisicing elit. Quisquam expedita ut officiis repudiandae sint, earum numquam magnam vel architecto autem, sequi omni facere, consequuntur vero non iusto. Architecto, est, accusamus?Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quisquam expedita ut officiis repudiandae sint earum numquam magnam vel architecto autem, sequi omnis facere, consequuntur vero non iusto. Architecto, est, accusamus?

### ::first-line

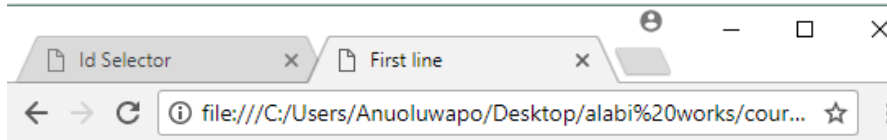
The *::first-line* pseudo-element is used to add a special style to the first line of a text.

#### CSS:

```
.post p::first-line{
  font-size:35px;
```

```
color:red;  
font-style:italic;  
}
```

result:



## Heading 1

*Lorem ipsum dolor sit amet,*

consectetur adipisicing elit. Quisquam expedita ut officiis repudiandae sint, earum numquam magnam vel architecto autem, sequi omnis facere, consequuntur vero non iusto. Architecto, est, accusamus? Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quisquam expedita ut officiis repudiandae sint, earum numquam magnam vel architecto autem, sequi omnis facere, consequuntur vero non iusto. Architecto, est, accusamus?

## ::before

The `::before` is able to add content before a certain element. For example, adding an opening quote before a blockquote or perhaps an preceding image to set apart a particular paragraph.

html:

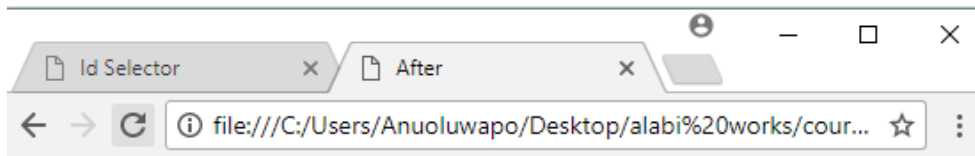
```
<body>  
  
  <div class="post">  
    <h2>Heading 1</h2>  
    <p>  
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam  
      provident consectetur dignissimos in totam, culpa, doloremque  
      vero amet mollitia earum tempore soluta perspiciatis maxime dolor  
      iste asperiores impedit voluptates dolores?  
    </p>  
  </div>  
  <div class="post">  
    <h2>Heading 1</h2>  
    <p>  
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam  
      provident consectetur dignissimos in totam, culpa, doloremque  
      vero amet mollitia earum tempore soluta perspiciatis maxime dolor  
      iste asperiores impedit voluptates dolores?  
    </p>  
  </div>  
</body>
```



#### CSS:

```
.post h2::before{
    content:"Chapter 1";
    font-size:13px;
    font-style:italic;
    color:red;
}
```

#### result:



### *Chapter 1* **Heading 1**

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam provident consectetur dignissimos in totam, culpa, doloremque vero amet mollitia earum tempore soluta perspiciatis maxime dolor iste asperiores impedit voluptates dolores?

### *Chapter 1* **Heading 1**

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam provident consectetur dignissimos in totam, culpa, doloremque vero amet mollitia earum tempore soluta perspiciatis maxime dolor iste asperiores impedit voluptates dolores?

#### ::after

The ::after pseudo element used to add content after a certain element. For example, a closing quote to a blockquote.

#### html:

#### Using the above html code

#### CSS:

```
.post h2::after{
    content:"Chapter 1";
    font-size:13px;
    font-style:italic;
    color:red;
}
```

#### result:

## Heading 1 *Chapter 1*

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam provident consectetur dignissimos in totam, culpa, doloremque vero amet mollitia earum tempore soluta perspiciatis maxime dolor iste asperiores impedit voluptates dolores?

## Heading 1 *Chapter 1*

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam provident consectetur dignissimos in totam, culpa, doloremque vero amet mollitia earum tempore soluta perspiciatis maxime dolor iste asperiores impedit voluptates dolores?

# Chapter 3

## Text

### Text transform

The **text-transform** property is used to change the case of text giving it one of the following values: **none**, **uppercase**, **lowercase**, and **capitalize**.

html:

```
<div class="post">
  <h2>Heading element</h2>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    Expedita
    unde ducimus minima laborum, veniam aliquam alias aliquid sequi
    ad
    impedit. Dolores corporis sapiente repudiandae! Rem autem nam
    tempora odio cupiditate.
  </p>
</div>
```

CSS:

```
.post{
    width:500px;
}

.post h2{
    text-transform:uppercase;/*Text Transform spacing css
property*/
}
```

result:

## HEADING ELEMENT

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Expedita unde ducimus minima laborum, veniam aliquam alias aliquid sequi ad impedit. Dolores corporis sapiente repudiandae! Rem autem nam tempora odio cupiditate.

### Text decoration

The text-decoration property allows you to specify the following values: **none**, **underline**, **overline**, **line-through**, **blink** (this animate the text to make it flash on and off).

html:



```

<div class="post">
  <h2>Heading element</h2>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Expedita
    unde ducimus minima laborum, veniam aliquam alias aliquid sequi ad
    impedit. Dolores corporis sapiente repudiandae! Rem autem nam
    tempora odio cupiditate.
  </p>
</div>

```

#### CSS:

```

.post{
  width:500px;
}

.post h2{
  text-decoration:underline;
}

```

#### result:

### Heading element

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Expedita unde ducimus minima laborum, veniam aliquam alias aliquid sequi ad impedit. Dolores corporis sapiente repudiandae! Rem autem nam tempora odio cupiditate.

## Line height

The **line-height** property is used to specify the space between lines: The value can be in **em**, **px**, **%**.

#### html:

```

<div class="post">
  <h2>Heading element</h2>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Expedita
    unde ducimus minima laborum, veniam aliquam alias aliquid sequi ad
    impedit. Dolores corporis sapiente repudiandae! Rem autem nam
    tempora odio cupiditate.
  </p>
</div>

```

#### CSS:

```

.post{
  width:500px;
}

.post p{
  line-height: 30px;
}

```

}  
**result:**

---

## Heading element

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Expedita unde ducimus minima laborum, veniam aliquam alias aliquid sequi ad impedit. Dolores corporis sapiente repudiandae! Rem autem nam tempora odio cupiditate.

## Letter spacing

The **letter-spacing** property is used to specify the space between the characters in a text. **em, px, %.** **letter-spacing: 5px;**

**html:**

```
<body>
  <div class="post">
    <h2>Heading element</h2>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Expedit
      unde ducimus minima laborum, veniam aliquam alias aliquid sequi
      ad impedit. Dolores corporis sapiente repudiandae! Rem autem nam
      tempora odio cupiditate.
    </p>
  </div>
</body>
```

**CSS:**

```
.post h2{
  text-transform:uppercase;
  letter-spacing:15px; /*Letter spacing css property*/
}
```

result:



## Word spacing

This is use to provide space between words, larger gap between words increases readability. the value for the word space property can be in *em*, *px*, *%*.

*word-spacing: 5px;*

html:

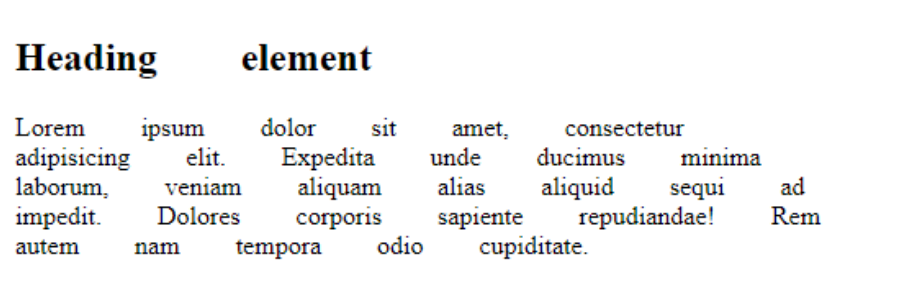
Using the above html code

css:

```
.post h2{
    word-spacing:45px; /*Word spacing css property*/
}

.post p{
    word-spacing:30px; /*Word spacing css property*/
}
```

result:



## Text indent

The **text-indent** property allows you to indent the first line of text within an element. The value can be in **em**, **px**, % **text-indent: 5px;**

### html:

Using the html above:

### css:

```
.post p{
    text-indent:30px; /*Text indent CSS property*/
}
```

### result:

## Heading element

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Expedita unde ducimus minima laborum, veniam aliquam alias aliquid sequi ad impedit. Dolores corporis sapiente repudiandae! Rem autem nam tempora odio cupiditate.

## Text shadow

The text-shadow property is used to create a drop shadow effect. The first length indicates how far to the left or right the shadow should fall. The second value indicates the distance to the top or bottom that the shadow should fall. The third value is optional and specifies the amount of blur that should be applied to the drop shadow. The fourth value is the color of the drop shadow.

**text-shadow: 1px 1px 0px #000000;**

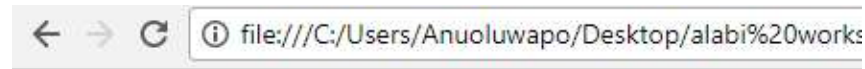
### html:

Using the html above:

### css:

```
h1{
text-shadow:1px 1px 15px black;
}
```

result:



## Text Shadow

### Vertical align

The vertical align text is used to align text and other inline elements such as `<img>`, `<em>`, `<strong>` elements to top, middle, bottom of table cells (`<td>`, `<th>` elements). Vertical align can have the following values *baseline*, *sub*, *super*, *top*, *text-top*, *middle*, *bottom*, *text-bottom*

***vertical-align:top;***

### Text align

The ***text-align*** property allows you to control the alignment of text. The property can take one of four values: **left** , **center** , **right** , **justify**.

**html:**

```
<body>
  <div class="post">
    <h2>Heading Element</h2>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit.
      Velit, eum. Facilis repellendus debitis eum ipsa provident
      totam sit voluptas aliquam blanditiis. Aliquid officiis
      magnam natus excepturi illo atque quibusdam nisi
      !Lorem ipsum dolor sit amet, consectetur adipisicing elit.
      Velit, eum. Facilis repellendus debitis eum ipsa provident
      totam sit voluptas aliquam blanditiis. Aliquid officiis
      magnam natus excepturi illo atque quibusdam nisi!
    </p>
  </div>
</body>
```

**CSS:**

```
.post{
  width:500px;
  text-align:justify;
}
```



result:

## Heading Element

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Velit, eum. Facilis repellendus debitis eum ipsa provident totam sit voluptas aliquam blanditiis. Aliquid officiis magnam natus excepturi illo atque quibusdam nisi !Lorem ipsum dolor sit amet, consectetur adipisicing elit. Velit, eum. Facilis repellendus debitis eum ipsa provident totam sit voluptas aliquam blanditiis. Aliquid officiis magnam natus excepturi illo atque quibusdam nisi!

## List style type

The **list-style-type** property allows you to control the shape or style of a bullet point . It can be used on rules that apply to the **<ol>**, **<ul>**, and **<li>** elements.

### Ordered Lists

For an ordered (numbered) list you can use the following values **lower-alpha**, **upper-alpha**, **lower-roman**, **upper-roman**, **decimal**, **decimal-leading-zero**.

html:

```
<body>

    <ol class="menu">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
    </ol>

</body>
```

css:

```
.menu{
    list-style-type:upper-alpha;
}
```

result:

---

- A. [Home](#)
- B. [About](#)
- C. [Blog](#)
- D. [Services](#)
- E. [Contact](#)

## Unordered Lists

For an unordered list you can use the following values **none**, **disc**, **circle**, **square**.

html:

`<body>`

```
<ol class="menu">
  <li><a href="#">Home</a></li>
  <li><a href="#">About</a></li>
  <li><a href="#">Blog</a></li>
  <li><a href="#">Services</a></li>
  <li><a href="#">Contact</a></li>
</ol>
```

`</body>`

CSS:

```
.menu{
  list-style-type: circle;
}
```

result:

---

- [Home](#)
- [About](#)
- [Blog](#)
- [Services](#)
- [Contact](#)

# Chapter 4

## Fonts

### Font family

The font-family property allows you to specify the typeface that should be used for any text inside the element(s) to which a CSS rule applies.

Since typeface of font are external file to the html file, font-family declared in css must be installed on user's computer for them to display. With the advent of CSS3 there are fix to this, which will be explained later.

The value for font-family is a list of typeface separated by comma. The browser picks the font to use from left to right depending on the one installed on the user's computer.

```
font-family: "Times New Roman", Georgia, serif;  
font-family: Verdana, Helvetica, sans-serif;  
font-family: Courier, "Courier New", monospace;
```

### Font Weight

The **font-weight** property allows you to create bold text. There are two values that this property commonly takes: **normal** and **bold**. You can also specify "**bolder**", "**lighter**", or a number from **100** to **900**.

### Font Style

If you want to create italic text, you can use the **font-style** property. There are three values this property can take: **normal**, **italic**, **oblique**.

### Font size

The font-size property enables you to specify a size for the font. There are several ways to specify the size of a font. The most common are: **px**, **%**, **em**. The default size of text by most browsers is 16px.

#### Font size (px)

When you specify your font in px the size of the text remains the same when you increase or decrease the font size on your Web browser. This type of unit (px) is referred to as an **absolute unit**.

## Font size (em)

When you specify your font in **em** the size of the text changes as you increase or decrease the font size on your Web browser. This type of unit (*em*) is referred to as *relative unit*. It is recommended that you specify your font size in *em* as this allows users to increase and decrease the size of their browser, **1em** is equivalent to **16px** and **2em** will be **32px**.

## Font size (%)

Percent (%) is a relative unit just as *em* when you specify your font size in % the size of the text changes as you increase or decrease the font size on your Web browser. **100%** is equivalent to **16px** and **75%** is equivalent to **12px**. If a parent has the font-size of **20px** and the child has a font-size of **50%**, it will come out to **10px**.

# Excerise

## ALABIAN DAILIES

[Download Document](#)

[Download Document](#)

### Buhari Commissions Seme Border Post

*L*orem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

[Read More](#)

#### States in Nigeria

- a. Lagos
- b. Edo
- c. Ondo
- d. Delta

### Kaduna Boils Again

*L*orem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

[Read More](#)

#### States in Nigeria

# Module 2

# Chapter 1

## Color

The color property allows you to specify the color of text inside an element . You can specify any color in CSS in one of three ways: **RGB, Color Names, Hex Codes**

### Color names

There are 147 predefined color names that are recognized by browsers. For example:

#### DarkCyan

```
.article{
    color:blue;
}
```

### HEX CODES

These are six-digit codes that represent the amount of red, green and blue in a color, preceded by a pound or hash # sign. For example: **#ee3e80**

```
h1{
    color:#0BA8A3;
}
```

### RGB VALUES

These express colors in terms of how much red, green and blue are used to make it up. For example: **rgb(100,100,90)**

#### html:

```
<body>
  <h1>Heading 1</h1>

  <div class="article">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Et
    aliquid quisquam magnam, eveniet, reiciendis quas fugit eos
    repellat vitae libero recusandae iure iste ab, quia illo modi quo
    quasi molestiae?
  </div>
</body>
```

#### css:

```
body{
    color:rgb(152, 11, 170);
}
```

result:

---

## Heading 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Et aliquid quisquam magnam, eveniet, reiciendis quas fugit eos repellat vitae libero recusandae iure iste ab, quia illo modi quo quasi molestiae?

### Opacity (rgba)

The CSS3 rgba property allows you to specify a color, just like you would with an RGB value, but adds a fourth value to indicate opacity. This value is known as an alpha value and is a number between 0.0 and 1.0 (so a value of **0.5 is 50% opacity** and **0.15 is 15% opacity**). The rgba value will only affect the element on which it is applied (not child elements).

**html:**

```
<body>

  <div class="post">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    Fugiat numquam atque, corrupti animi velit facilis consequatur,
    eaque ad natus, modi corporis quasi repudiandae sapiente
    pariatur, amet dolorem asperiores itaque adipisci.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    Fugiat numquam atque, corrupti animi velit facilis consequatur,
    eaque ad natus, modi corporis quasi repudiandae sapiente
    pariatur, amet dolorem asperiores itaque adipisci.</p>
  </div>
  <div class="article">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    Fugiat numquam atque, corrupti animi velit facilis consequatur,
    eaque ad natus, modi corporis quasi repudiandae sapiente
    pariatur, amet dolorem asperiores itaque adipisci.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
    Fugiat numquam atque, corrupti animi velit facilis consequatur,
    eaque ad natus, modi corporis quasi repudiandae sapiente
    pariatur, amet dolorem asperiores itaque adipisci.</p>
  </div>
</body>
```

**CSS:**

```
<style type="text/css">

  body{
    background-image:url("img-rgba.jpg");
    background-repeat: no-repeat;
    background-size:cover;
  }

  .post{
    width:600px;
    margin:auto;
```



```

padding:10px;
background-color:rgb(4, 69, 107);
color:#fff;
margin-bottom: 20px;
}

.article{
width:600px;
margin:auto;
padding:10px;
background-color:rgba(4, 69, 107, 0.6);/*this is opacity
(rgba) */
color:#fff;
}

```

**result:**



# Chapter 2

## Background

The CSS background properties are used to define the background effects for elements.

CSS background properties:

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

## Background Color

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

### html

```
<div id="container">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Fuga
    soluta, eligendi illo neque animi natus porro delectus ut tempore, maiores
    distinctio voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.Lorem ipsum dolor sit amet, consectetur adipisicing elit. Fuga
    soluta, eligendi illo neque animi natus porro delectus ut tempore, maiores
    distinctio voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.Lorem ipsum dolor sit amet, consectetur adipisicing elit. Fuga
    soluta, eligendi illo neque animi natus porro delectus ut tempore, maiores
    distinctio voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.Lorem ipsum dolor sit amet, consectetur adipisicing elit. Fuga
    soluta, eligendi illo neque animi natus porro delectus
</div>
```

### CSS

```
#container{
width:650px;
height:250px;
margin:auto;
color:#fff;
background-color:red;
}
```

## Background Image

The **background-image** property allows you to place an image behind any HTML element. This could be the entire page or just part of the page. By default, a background image will repeat to fill the entire box.

**html:**

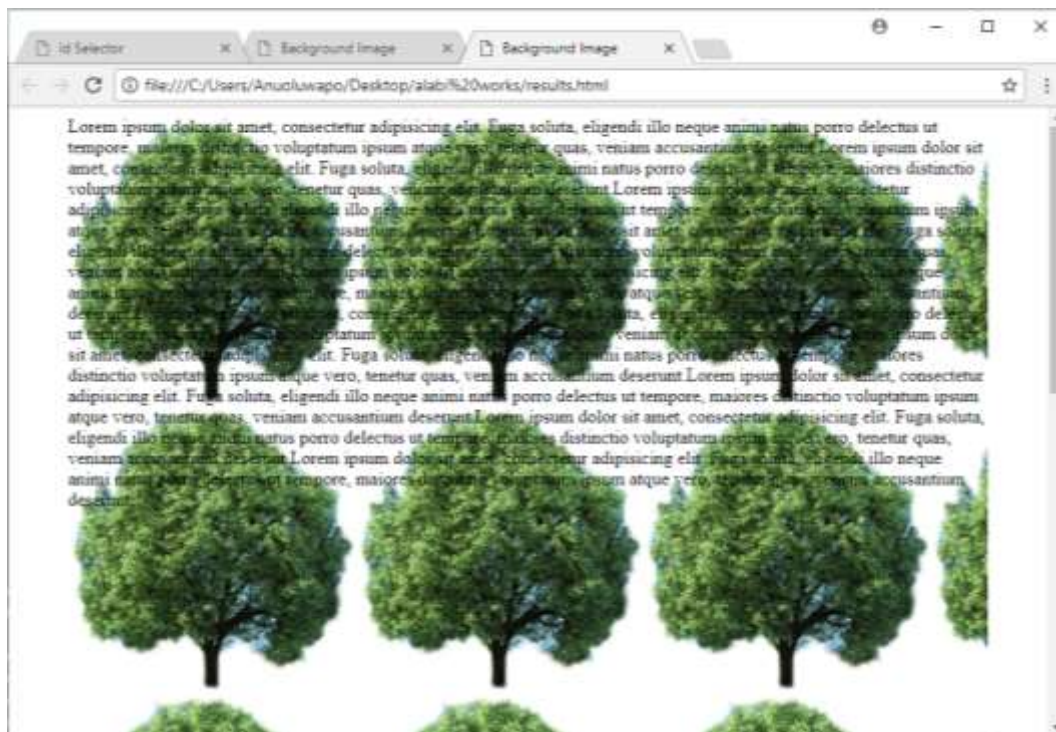
```
<body>
  <div id="container">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Fuga
    soluta, eligendi illo neque animi natus porro delectus ut
    tempore, maiores distinctio voluptatum ipsum atque vero, tenetur
    quas, veniam accusantium deserunt.Lorem ipsum dolor sit amet,
    consectetur adipisicing elit. Fuga soluta, eligendi illo neque
    animi natus porro delectus ut tempore, maiores distinctio
    voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.Lorem ipsum dolor sit amet, consectetur adipisicing
    elit. Fuga soluta, eligendi illo neque animi natus porro delectus
    ut tempore, maiores distinctio voluptatum ipsum atque vero,
    tenetur quas, veniam accusantium deserunt.Lorem ipsum dolor sit
    amet, consectetur adipisicing elit. Fuga soluta, eligendi illo
    neque animi natus porro delectus ut tempore, maiores distinctio
    voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.Lorem ipsum dolor sit amet, consectetur adipisicing
    elit. Fuga soluta, eligendi illo neque animi natus porro delectus
    ut tempore, maiores distinctio voluptatum ipsum atque vero,
    tenetur quas, veniam accusantium deserunt.Lorem ipsum dolor sit
    amet, consectetur adipisicing elit. Fuga soluta, eligendi illo
    neque animi natus porro delectus ut tempore, maiores distinctio
    voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.Lorem ipsum dolor sit amet, consectetur adipisicing
    elit. Fuga soluta, eligendi illo neque animi natus porro delectus
    ut tempore, maiores distinctio voluptatum ipsum atque vero,
    tenetur quas, veniam accusantium deserunt.Lorem ipsum dolor sit
    amet, consectetur adipisicing elit. Fuga soluta, eligendi illo
    neque animi natus porro delectus ut tempore, maiores distinctio
    voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.Lorem ipsum dolor sit amet, consectetur adipisicing
    elit. Fuga soluta, eligendi illo neque animi natus porro delectus
    ut tempore, maiores distinctio voluptatum ipsum atque vero,
    tenetur quas, veniam accusantium deserunt.Lorem ipsum dolor sit
    amet, consectetur adipisicing elit. Fuga soluta, eligendi illo
    neque animi natus porro delectus ut tempore, maiores distinctio
    voluptatum ipsum atque vero, tenetur quas, veniam accusantium
    deserunt.
  </div>
</body>
```

**CSS:**

```
#container{
    width:800px;
    height:1200px;
    margin:auto;
    background-image: url("tree.png");
}
```

**result:**





## Background image property

### Background repeat

The **background-repeat** property can have four values:

#### repeat

The **background image** is repeated both horizontally and vertically (the default way it is shown if the background-repeat property isn't used).

#### repeat-x

The image is repeated horizontally only (as shown in the first example on the left).

#### html:

Using the HTML code above

#### css:

```
#container{
    width:800px;
    height:1200px;
    margin:auto;
    background-image: url("tree.png");
    background-repeat: repeat-x;
}
```

result:



## repeat-y

The image is repeated vertically only.

html:

Using the HTML code above

css:

```
#container{
    width:800px;
    height:1200px;
    margin:auto;
    background-image: url("tree.png");
    background-repeat: repeat-y;
}
```

result:



## no-repeat

The image is only shown once.

Using the HTML code above

**CSS:**

```
#container{
    width:800px;
    height:1200px;
    margin:auto;
    background-image: url("tree.png");
    background-repeat:no-repeat;
}
```

**result:**



## Background position

When an image is not being repeated, you can use the background-position property to specify where in the browser window the background image should be placed. This property usually has a pair of values. The first represents the horizontal position and the second represents the vertical.

left top, left center, left bottom, center top, center center , center bot tom, right top, right center, right bot tom, 50% 50%, 20px 30px.

html:

Using the HTML code above

## CSS:

```
#container{
  width:800px;
  height:1200px;
  margin:auto;
  background-image: url("tree.png");
```



}

**result:**



## Background attachment

Background attachment is use to specify that background image is fixed or scroll to the rest of the page. It have two values.

**fixed**

The background image stays in the same position on the page.

**html:**

Using the HTML code above

**CSS:**

```
#container{
    width:800px;
    height:1200px;
    margin:auto;
    background-image: url("tree.png");
    background-repeat:no-repeat;
    background-position:center top;
    background-attachment:fixed;
}
```

**result:**

Check your browser

## scroll

The background image moves up and down as the user scrolls up and down the page.

**html:**

Using the HTML code above

**css:**

```
#container{
    width:800px;
    height:1200px;
    margin:auto;
    background-image: url("tree.png");
    background-repeat:no-repeat;
    background-position:center top;
    background-attachment:scroll;
}
```

**result:**

Check your browser

## Background shorthand

**html:**

Using the HTML code above

**css:**

```
#container{
    width:800px;
    height:1200px;
    margin:auto;

    background:url("tree.png") no-repeat center top #fff fixed;
}
```



result:



## Background size

The CSS3 **background-size** property allows you to specify the size of background images inside its container. Before CSS3, the size of a background image was the actual size of the image. The size can be specified in pixels, percentages, or by using one of the two keywords: **contain** or **cover**.

### cover

The cover keyword scales the background image so that the content area is completely covered by the background image (both its width and height are equal to or exceed the content area) . As such, some parts of the background image may not be visible in the background positioning area.

The following example illustrates the use of **cover**:

html:

```
<body>

</body>
```

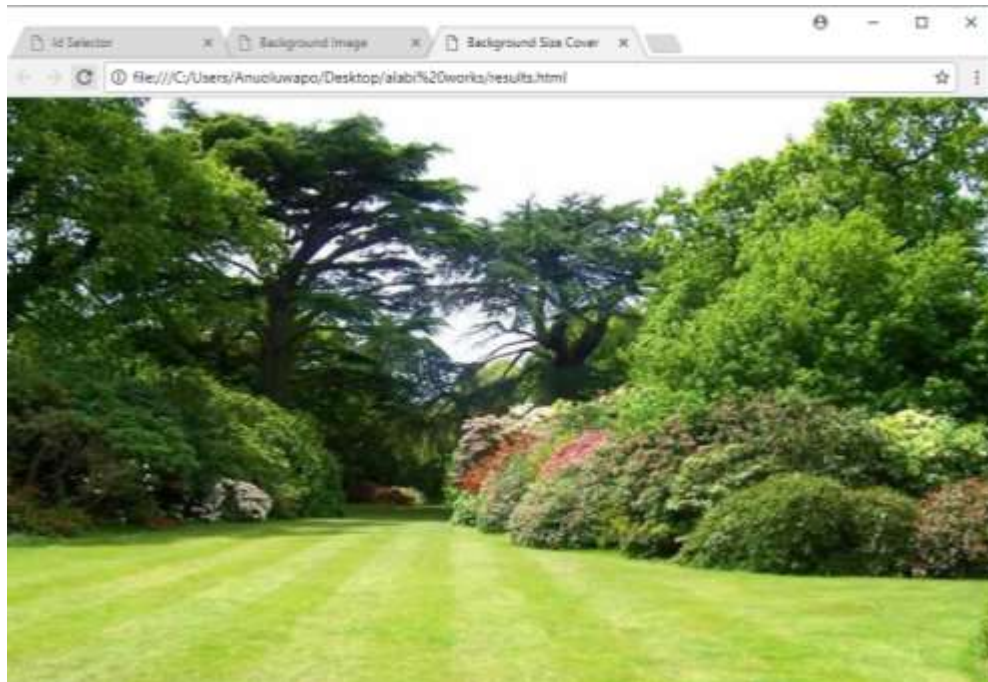
css:

```
body{

    background-image:url("back.jpg");
    background-repeat:no-repeat;
    -webkit-background-size:cover;
    -moz-background-size:cover;
    -ms-background-size:cover;
    -o-background-size:cover;
    background-size:cover;

}
```

## Result:



### **contain**

The contain keyword scales the background image to be as large as possible (but both its width and its height must fit inside the content area). As such, depending on the proportions of the background image and the background positioning area, there may be some areas of the background which are not covered by the background image.

# Chapter 2

## Web Developer tools

We are going to be looking at tools that will make your life easy when developing websites or web applications.

### Google Chrome inspect element

Google Chrome is used to make quick CSS changes, it works by showing you the stylesheet relevant to the page you are looking at , in addition to that Google chrome inspect element can help you diagnose issues with the construction of your web page.

### Firebug

Firebug is an extension that is added to mozilla firefox, it allows editing your HTML and CSS, this is also similar to Google Chrome.

<http://www.getfirebug.com>

### ColorZilla

ColorZilla is an Eyedropper that allows you to sample colors with in your web page. ColorZilla extension is available for Chrome and Firefox.

<http://www.colorzilla.com>

### WhatFont for Chrome

WhatFont is used to find out fonts used in a web page. With this extension, you could inspect web fonts by just hovering on them.

<http://www.chengyinliu.com/whatfont.html>

### Scrapbook for Firefox

This is a firefox extension that helps you to save web pages and easily manage

<https://addons.mozilla.org/en-US/firefox/addon/scrapbook-plus-2/>

### Awesome Screen for Chrome

Capture all or part of any web page. Add annotations, comments, blur sensitive info, and share with one-click uploads. Now with free desktop capture!

<http://www.awesomescreenshot.com/>

# Chapter 3

## More on fonts

To fix the issue of typeface declared for the font-family in css not been installed on user's computer, Google Font is used to link external typeface files or CSS3 font is used to host these files on our server.

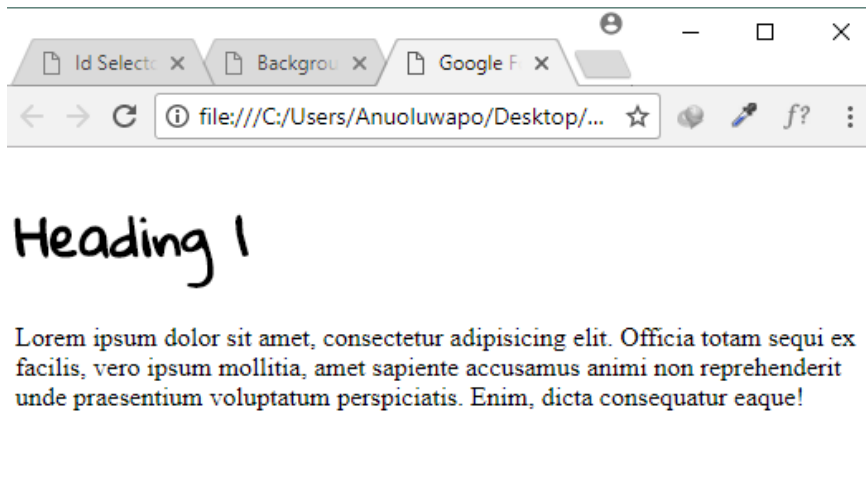
## Using Google Fonts

- Navigate to [fonts.google.com](https://fonts.google.com) and locate the desired font.
- Select the font-family you want (you can select more than 1 font-family)
- Copy the code snippet for linking the font's typeface file and paste in the head section of your html code.
- You can now safely use the font-family for your styling purpose

### html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Google Fonts</title>
  <link href="https://fonts.googleapis.com/css?family=Gloria+Hallelujah"
rel="stylesheet">
  <style type="text/css">
    h1{
      font-family: 'Gloria Hallelujah', cursive;
    }
  </style>
</head>
<body>
  <h1>Heading 1</h1>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Officia totam
    sequi ex facilis, vero ipsum mollitia, amet sapiente accusamus animi
    non reprehenderit unde praesentium voluptatum perspiciatis. Enim, dicta
    consequatur eaque!
  </p>
</body>
</html>
```

## Result:



## CSS 3 Font

@font-face is a CSS rule which allows you to show a font on a Web page even if that font is not installed on the users' computer. This technique allows you to specify a path to a copy of the font that will be downloaded if it is not on the user's machine. It is important to specify different font format so the visitor's browser can pick any format that it can support.

Here are the list of format

- .ttf
- .eot
- .svg
- .woff
- .woff2

## font-family

This specifies the name of the font. This name can then be used as a value of the font-family property in the rest of the style sheet (as shown in the rule for the <h1> element).

## src

This specifies the path to the font. In order for this technique to work in all browsers, you will probably need to specify paths to a few different versions of the font, as shown on the code above.

## format

This specifies the format that the font is supplied in.

## html:

```
<body>
```

```
    <h1>Main Heading</h1>
```

```
</body>
```

#### CSS:

```
@font-face {
    font-family: 'amaranthBold';
    src: url('fonts/amaranth-bold-webfont.eot');
    src: url('fonts/amaranth-bold-webfont.eot?#iefix')
    format('embedded-opentype'),
        url('fonts/amaranth-bold-webfont.woff2')
    format('woff2'),
        url('fonts/amaranth-bold-webfont.woff') format('woff'),
        url('fonts/amaranth-bold-webfont.ttf')
    format('truetype'),
        url('fonts/amaranth-bold-webfont.svg#amaranthregular')
    format('svg');
    font-weight: normal;
    font-style: normal;
}

h1{
    font-family: 'amaranthBold', sans-serif;
}
```

result:

# Main Heading

# MODULE 3

# Chapter 1

## Width and Height

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** (the default and this implies that the browser calculates the height and width), or be specified.

## Block level element

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

### html:

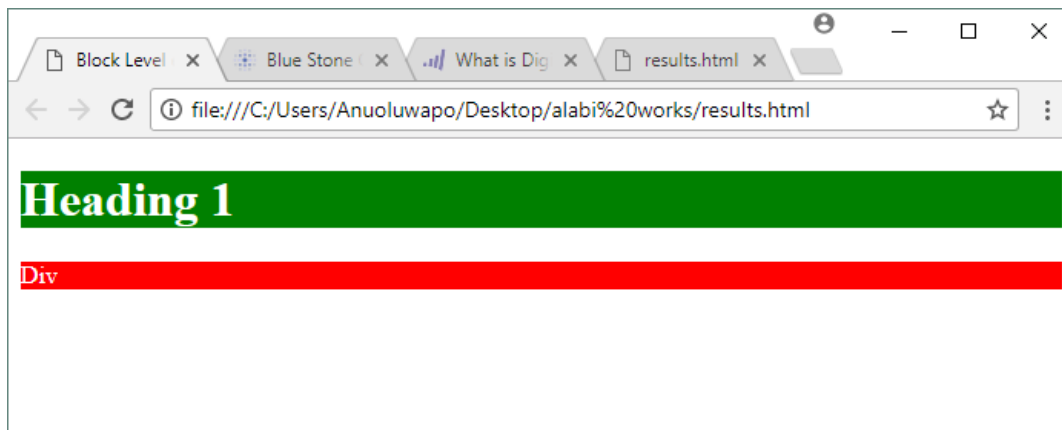
```
<h1>Heading 1</h1>
<div>Div</div>
```

### CSS:

```
h1{
    background-color:green;
    color:white;
}

div{
    background-color:red;
    color:white;
}
```

### Result:





## Inline element

An inline element does not start on a new line and only takes up as much width as necessary.

### html:

```
<a href="#">Anchor</a>  
<span>Span</span>
```

### css:

```
a{  
    background-color:blue;  
    color:white;  
}  
span{  
    background-color:green;  
    color:white;  
}
```

### result:



## Display

The **display** property allows you to turn an inline element into a block-level element or vice versa, and can also be used to hide an element from the page.

The values this property can take are:

- block
- inline
- inline-block
- none

### inline

This causes a block-level element to act like an inline element.

### block

This causes an inline element to act like a block-level element.

### inline-block

This causes a block-level element to flow like an inline element, while retaining other features of a block-level element.

## **none**

This hides an element from the page. In this case, the element acts as though it is not on the page at all (although a user could still see the content of the box if they used the view source option in their browser).

## **visible**

This shows the element. If the visibility of an element is set to hidden, a blank space will appear in its place.

If you do not want a blank space to appear, then you should use the display property with a value of none instead.

# Chapter 2

## Border, Margin, Padding, Box Model

### Border

The **border** property styles the edge around the box and is specified as "thickness style color".

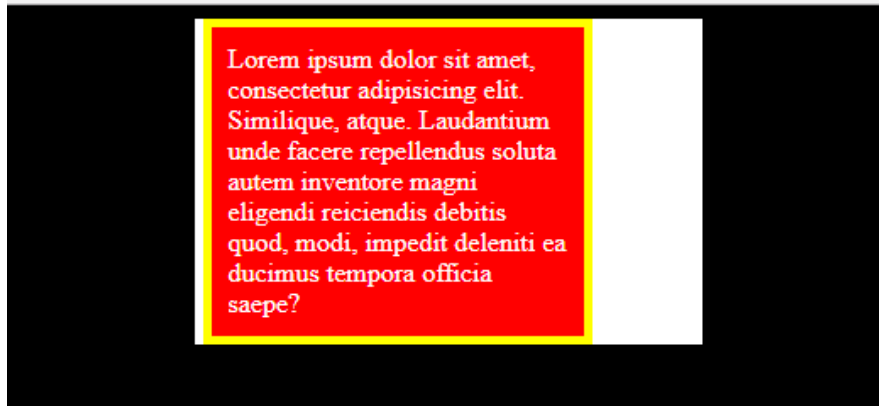
#### html:

```
<body>
  <div id="container">
    <div class="box1">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit.
      Similique, atque. Laudantium unde facere repellendus soluta
      autem inventore magni eligendi reiciendis debitis quod,
      modi, impedit deleniti ea ducimus tempora officia saepe?
    </div>
  </div>
</body>
```

#### CSS:

```
body{
  background-color:#000;
  color:#fff;
}
#container{
  background-color: #fff;
  width:300px;
  margin:auto;
}
.box1{
  padding:10px;
  background-color:red;
  width:200px;
  margin-left:5px;
  border-color:yellow;
  border-width:5px;
  border-style:solid;
}
```

#### result:



## Border Shorthand for the four sides

The shorthand for the border on all the sides are shown below

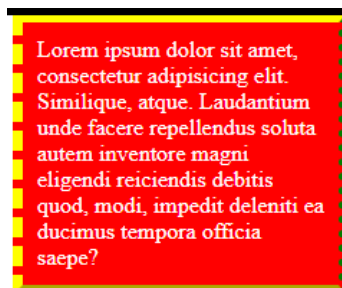
```
.box1{  
    border:yellow 5px solid;  
}
```

## Shorthand for one side

You can apply a border shorthand to each side of an element

```
.box1{  
    border-top:5px yellow solid;  
    border-right:5px green dotted;  
    border-left:7px yellow dashed;  
    border-bottom:7px yellow groove;  
}
```

**result:**



## Margin

Margins sit outside the edge of the border. You can set the width of a margin to create a gap between the borders of two adjacent boxes.

Please note: If a width is specified for a box, margin is added onto the width of the box.

You can specify different values for each side of a box using: **margin-top**, **margin-right**, **margin-bottom**, **margin-left**

## Before Margin

html:

```
<body>
<div id="container">
    <div class="box1">
        Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
    </div>
    <div class="box2">
        Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
    </div>
    <div class="box3">
        Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
    </div>
</div>
</body>
```

css:

```
body{
    background-color:#fff;
    color:#fff;
}

.box1, .box2, .box3{
    background-color:red;
    border:1px solid black;
    width:200px;
}
```

result:



Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

## After Margin

## CSS

```
body{  
    background-color:#fff;  
    color:#fff;  
}  
  
.box1, .box2, .box3{  
    background-color:red;  
    border:1px solid black;  
    width:200px;  
}  
  
.box1{  
    margin-bottom:15px;  
}  
  
.box2{  
    margin-bottom:20px;  
}
```

## result:

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?





## Margin shorthand

### margin: 8px 20px

**8px** will affect the top and bottom margin and **20px** will affect the left and right margin

### margin: 15px

You use this shorthand when you want the sides whole of the box (margin-top, margin-right, margin-bottom, margin-left) to have the same value for margin.

### margin: 8px 4px 10px 12px

Use this shorthand when you want different sides of a box to have different values for margin. 8px 4px 10px 12px will affect the Top, Right, Bottom and Left margin in that order.

## Padding

The padding property allows you to specify how much space should appear between the content of an element and its border.

Please note: If a width is specified for a box, padding is added onto the width of the box. You can specify different values for each side of a box using:

padding-top, padding-right, padding-bottom, padding-left or you can use a shorthand (where the values are in clockwise order: **top, right, bottom, left**) **padding:10px 5px 3px 1px**; another way to remember this order is by using the mnemonics **TRBL** which is pronounce “trouble” T for top, R for right, B for bottom, L for.

## Before Padding

html:

```
<body>
```

```
    <div id="container">
      <div class="box1">
        Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
      </div>
      <div class="box2">
        Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
      </div>
      <div class="box3">
```

Lorem ipsum dolor sit amet, consectetur adipisicing  
elit. Similique, atque. Laudantium unde facere repellendus soluta autem  
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea  
ducimus tempora officia saepe?

```
                  </div>  
</div>
```

```
</body>
```

### CSS:

```
body{  
  
                  background-color:#fff;  
                  color:#fff;  
  
}  
  
.box1, .box2, .box3{  
                  background-color:red;  
                  border:1px solid black;  
                  width:200px;  
  
}  
  
.box1{  
                  margin-bottom:15px;  
  
}  
  
.box2{  
                  margin-bottom:20px;  
  
}
```

### result:

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

## After padding

### html:

```
<body>

    <div id="container">
        <div class="box1">
            Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
        </div>
        <div class="box2">
            Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
        </div>
        <div class="box3">
            Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Similique, atque. Laudantium unde facere repellendus soluta autem
inventore magni eligendi reiciendis debitis quod, modi, impedit deleniti ea
ducimus tempora officia saepe?
        </div>
    </div>
```

### </body>

### CSS:

```
body{

    background-color:#fff;
    color:#fff;

}

.box1, .box2, .box3{
    background-color:red;
    border:1px solid black;
    width:200px;
}

.box1{
    margin-bottom:15px;
    padding:8px 20px;
}

.box2{
    padding:10px;
    margin-bottom:20px;
    padding: 8px 4px 10px 20px;
}
```

result:

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

Lorem ipsum dolor sit amet,  
consectetur adipisicing elit.  
Similique, atque. Laudantium  
unde facere repellendus soluta  
autem inventore magni  
eligendi reiciendis debitis  
quod, modi, impedit deleniti ea  
ducimus tempora officia  
saepe?

## Padding shorthand

### padding: 8px 20px

8px will affect the top and bottom padding and 20px will affect the left and right padding

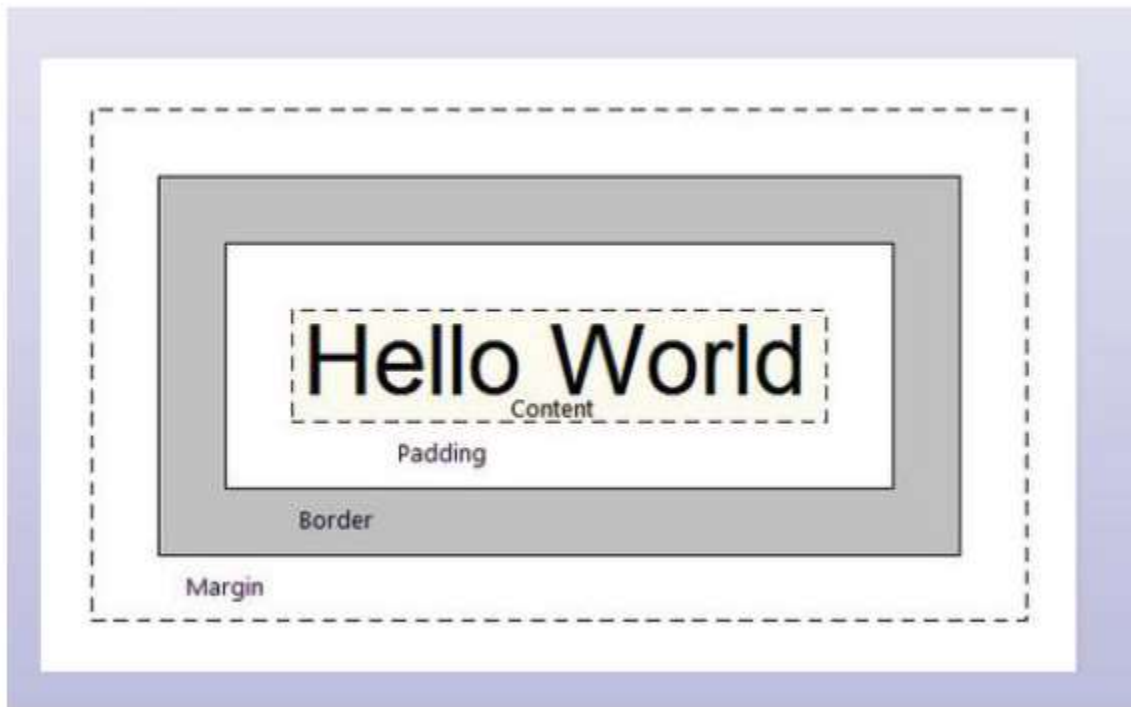
### padding: 15px

You use this shorthand when you want the sides whole of the box (padding-top, padding-right, padding-bottom, padding-left) to have the same value for padding.

## Box Model

All elements are treated as boxes by the browser's rendering engine (either "inline" or "block" boxes).

A box consists of content, padding, border, and margin:



# Chapter 3

## Box Sizing, Border Radius, Box Shadow

### CSS3 Box Sizing

The CSS3 box-sizing property allows us to include the padding and border in an element's total width and height.

#### Browser Support

The numbers in the table specify the first browser version that fully supports the property. Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property	Chrome	Edge	Explorer	Firefox	Safari	Opera
Border-radius	10.0 4.0 - webkit-	12.0	8.0	29.0 2.0 - moz-	5.1 3.1 - webkit-	9.5

#### Without the CSS3 box-sizing Property

By default, the width and height of an element is calculated like this:

width + padding + border = actual width of an element  
height + padding + border = actual height of an element

This means: When you set the width/height of an element, the element often appear bigger than you have set (because the element's border and padding are added to the element's specified width/height).

The following illustration shows two boxes elements with the same specified width and height:

## html:

```
<body>

    <div class="box1">
        box 1
    </div>
    <div class="box2">
        box 2
    </div>

</body>
```

## CSS:

```
.box1, .box2{
    background-color:blue;
    width:300px;
    height:100px;
    text-align: center;
    line-height:100px;
    text-transform: uppercase;
    font-weight: bold;
    color:#fff;
    margin-bottom: 15px;
}

.box2{
    padding:15px;
}
```

## result:



**box1 is smaller (width is 300px and height is 100px).**

**box2 is bigger (width is also 300px and height is 100px).**

**The two boxes elements above end up with different sizes in the result (because box2 has a padding specified):**

So, for a long time web developers have specified a smaller width value than they wanted, because they had to subtract out the padding and borders.

With CSS3, the box-sizing property solves this problem.

## With the CSS3 box-sizing Property

The CSS3 **box-sizing** property allows us to include the padding and border in an element's total width and height.

If you set **box-sizing: border-box;** on an element padding and border are included in the width and height:

**Both boxes are the same size now!**

**Here is the same example as above, with box-sizing: border-box; added to both boxes:**

Since the result of using the box-sizing: border-box; is so much better, many developers want all elements on their pages to work this way.

The code below ensures that all elements are sized in this more intuitive way. Many browsers already use **box-sizing: border-box;** for many form elements (but not all which is why inputs and textareas look different at width: 100%;).

Applying this to all elements is safe and wise:

**html:**

```
<body>
```

```
  <div class="box1">
    box 1
  </div>
  <div class="box2">
    box 2
  </div>
```

```
</body>
```

**CSS:**

```
.box1, .box2{
    background-color:blue;
    width:300px;
    height:100px;
```



```

        text-align: center;
        line-height: 70px;
        text-transform: uppercase;
        font-weight: bold;
        color: #fff;
        margin-bottom: 15px;
        padding: 15px;
        box-sizing: border-box;
    }

```

result:



## Border Radius

The border-radius property is used to add rounded corners to an element.

### Browser Support

The numbers in the table specify the first browser version that fully supports the property. Numbers followed by -webkit- or -moz- specify the first version that worked with a prefix.

Property	Chrome	Edge	Explorer	Firefox	Safari	Opera
Border-radius	5.0 4.0 - webkit-	12.0	9.0	4.0 3.0 -moz-	5.0 3.1 - webkit-	10.5

The border-radius property is a shorthand property for setting the four border-\*-radius properties.

If you specify only one value for the border-radius property, this radius will be applied to all 4 corners.

#### html

```
<div class="box">

</div>
```

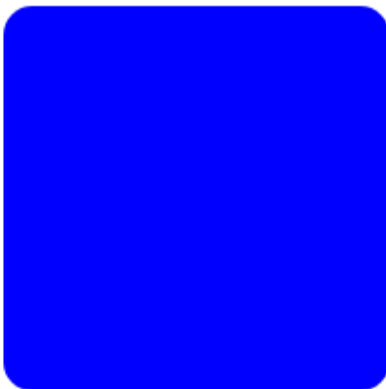
#### CSS

```
.box{

    width:200px;
    height:200px;
    background-color:blue;
    -webkit-border-radius:15px;
    -moz-border-radius:15px;
    -ms-border-radius:15px;
    border-radius:15px;

}
```

#### result



### Creating a circle with border radius

You create a circle by giving the width and height an equal value and then give the border radius a value of 50%.

#### html:

```
<body>

    <a href="#" class="readmore"></a>

</body>
```

#### CSS:

```
.readmore{

    height: 200px;
    width: 200px;
    background-color:#F7C23D;
    color:#fff;
```

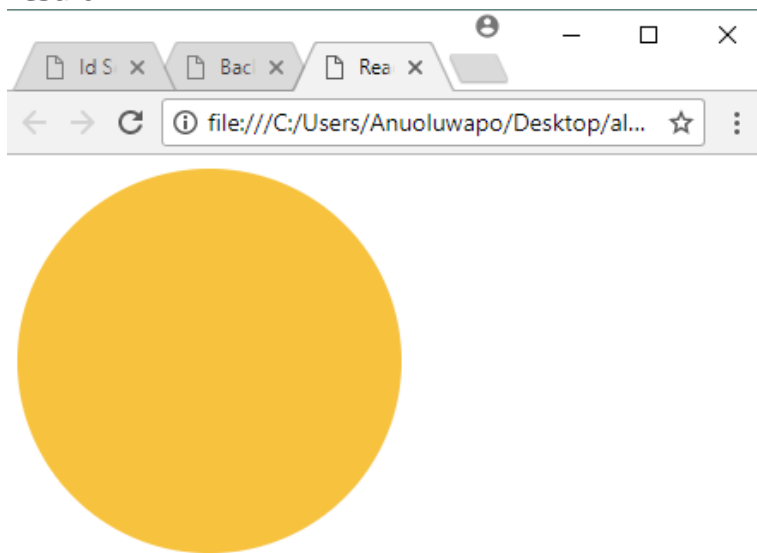
```

display:inline-block;
text-decoration: none;
padding:12px 12px;
-webkit-border-radius: 100%;
-moz-border-radius: 100%;
-ms-border-radius:100%;
border-radius:100%;
}

.readmore:hover{
background-color:#BC8505;
}

```

result:



## Box Shadow

The CSS3 *box-shadow* property applies shadow to elements. In its simplest use, you only specify the horizontal shadow and the vertical shadow:

**html:**

```
<body>
```

```

    <div class="box">
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
        Illo, qui. Laboriosam consequatur sapiente, odit debitis itaque odio facere
        nobis iste quaerat possimus quisquam minima, excepturi dolorum ratione
        praesentium quod sint.</p>
    </div>

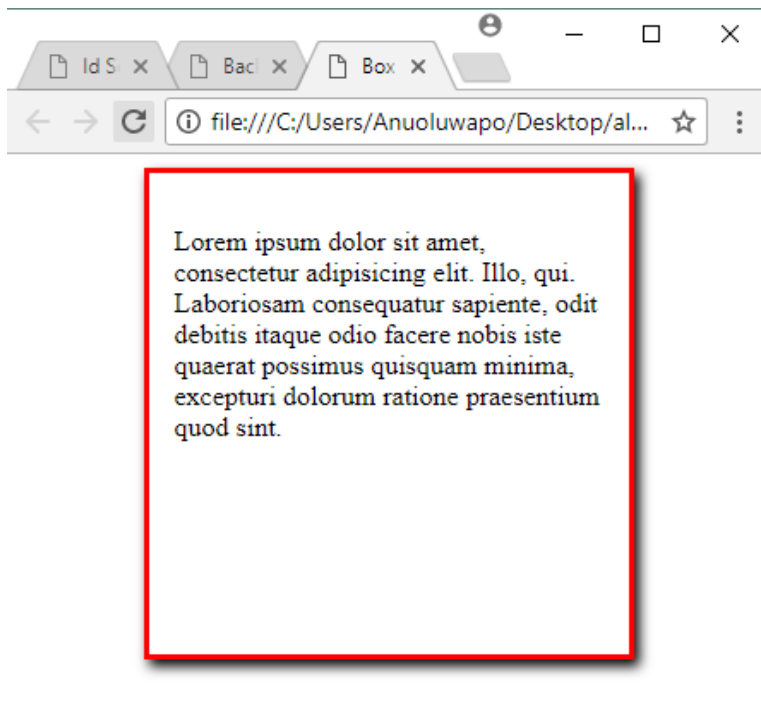
```

```
</body>
```

## CSS:

```
.box{  
    width:250px;  
    height:250px;  
    padding:15px;  
    background-color:#fff;  
    border:solid 3px red;  
    margin:auto;  
    -webkit-box-shadow:3px 4px 10px #000;  
    -moz-box-shadow:3px 4px 10px #000;  
    -ms-box-shadow:3px 4px 10px #000;  
    box-shadow:3px 4px 10px #000;  
}
```

## result:



## Browser Support

The numbers in the table specify the first browser version that fully supports the property. Numbers followed by -webkit- or -moz- specifies the first version that worked with a prefix.

Property	Chrome	Edge	Explorer	Firefox	Safari	Opera
box-shadow	10.0 4.0 - webkit	12.0	9.0	4.0 3.5 -moz-	5.1 5.1 - webkit-	10.5

The box shadow css property can take four values the first value is the horizontal shadow, second value is vertical shadow, third value is how blur you can the shadow to be and the fifth one is the color of the blur. The horizontal and vertical shadow can take negative values as well.

# Chapter 4

---

## Position

The position property is used to specify a positioning scheme for an element . The default is "static" which puts the element in the "normal" flow.

In normal flow, inline boxes flow from left to right, block level element will begin in a new line as usual.

In normal flow, block boxes flow from top to bottom, making a new line af ter every box.

## 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.

### html:

```
<body>

    <div id="container">
        <div class="box1">
            box1
        </div>
        <div class="box2">
            box2
        </div>
        <div class="box3">
            box3
        </div>
    </div>
```

</body>

### CSS:

```
body{

    background-color: #000;

}

#container{
    background-color:#fff;
    width:450px;
    margin:auto;
}

.box1, .box2, .box3{
```

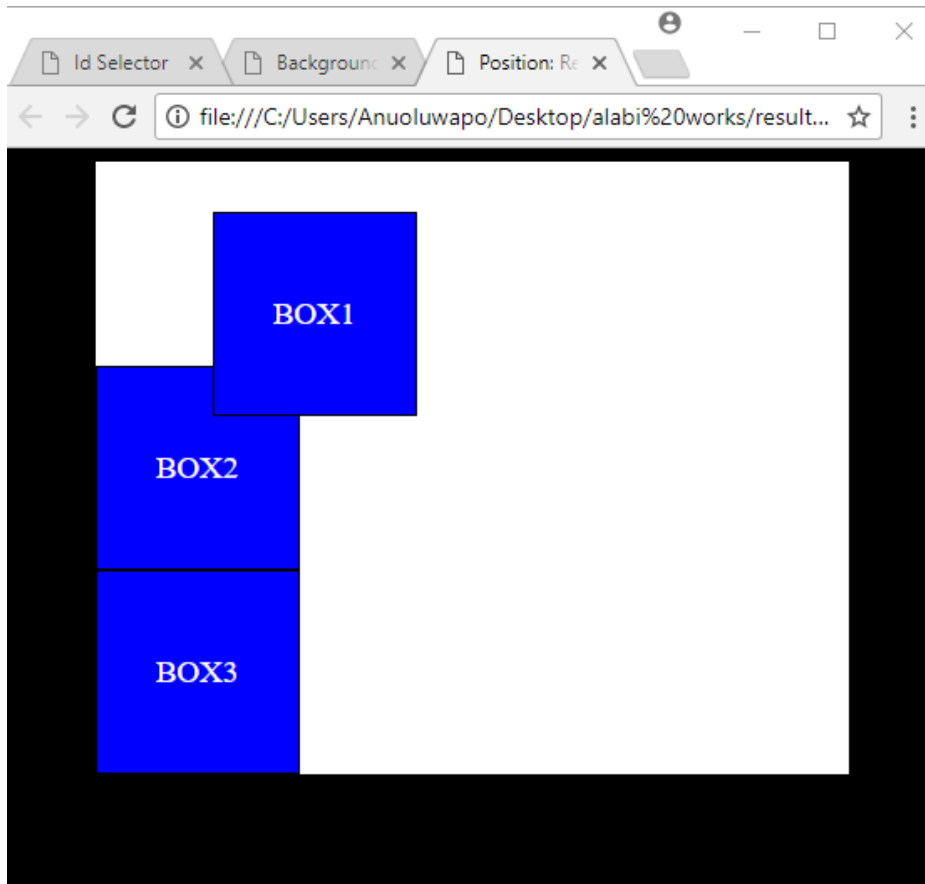
```

        background-color:blue;
        width:120px;
        height:120px;
        color:#fff;
        border:1px solid #000;
        text-transform: uppercase;
        font-size:19px;
        text-align: center;
        line-height:120px
    }

    .box1{
        position:relative;
        top:30px;
        left:70px;
    }

```

**result:**



## Position: Absolute

If is positioned with absolute, that element is positioned in relation to the browser window and not the parent element.

**html:**

<body>

```

<div id="container">
  <div class="box1">
    box1
  </div>
  <div class="box2">
    box2
  </div>
  <div class="box3">
    box3
  </div>
</div>

```

</body>

**CSS:**

```

body{
    background-color: #000;
}

#container{
    background-color:#fff;
    width:450px;
    margin:auto;
}

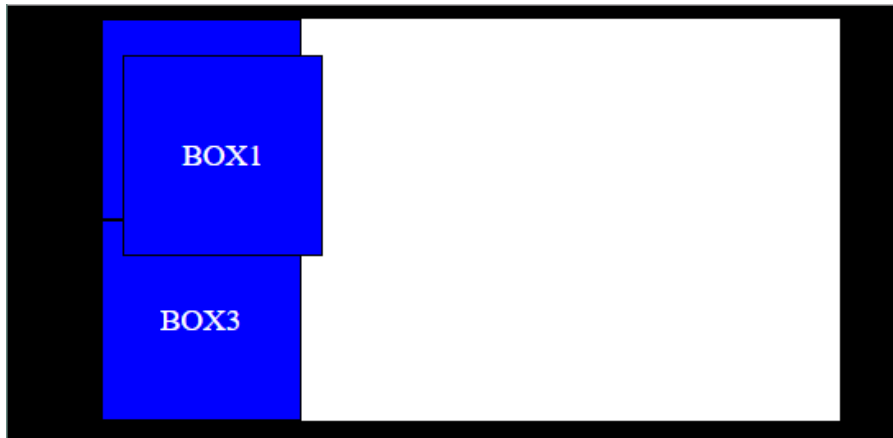
.box1, .box2, .box3{
    background-color:blue;
    width:120px;
    height:120px;
    color:#fff;
    border:1px solid #000;
    text-transform: uppercase;
    font-size:19px;
    text-align: center;
    line-height:120px;
}

.box1{
    position:absolute;
    top:30px;
    left:70px;
}

```

**result:**





## Relative Absolute

If you want an element to be positioned absolutely in relation to its parent element or you want an element to move within its parent, give the parent a position of relative and give the child element a position of absolute.

### html:

```
<body>
```

```
    <div id="container">
      <div class="box1">
        box1
      </div>
      <div class="box2">
        box2
      </div>
      <div class="box3">
        box3
      </div>
    </div>
```

```
</body>
```

### CSS:

```
body{
    background-color: #000;
}

#container{
    background-color:#fff;
    width:450px;
    position:relative;
    margin:auto;
}

.box1, .box2, .box3{
    background-color:blue;
    width:120px;
    height:120px;
    color:#fff;
    border:1px solid #000;
```

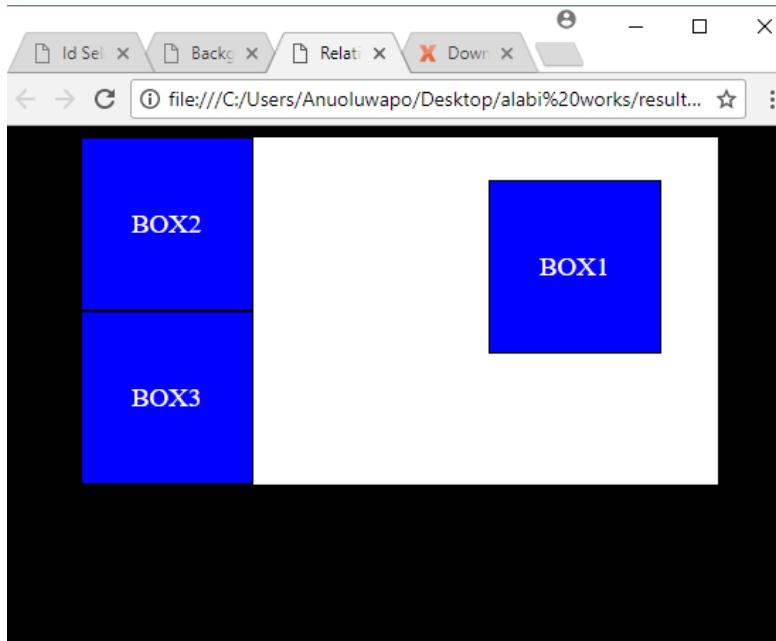
```

        text-transform: uppercase;
        font-size: 19px;
        text-align: center;
        line-height: 120px;
    }

    .box1{
        position: absolute;
        top: 30px;
        right: 40px;
    }

```

**result:**



## Position: Fixed

If you want an element to remain fixed while a user scrolls down a long page, simply give the element a position of fixed.

**html:**

<body>

```

    <div id="container">
        <div class="box1">
            box1
        </div>
        <div class="box2">
            box2
        </div>
        <div class="box3">
            box3
        </div>
    </div>

```

</body>

**CSS:**

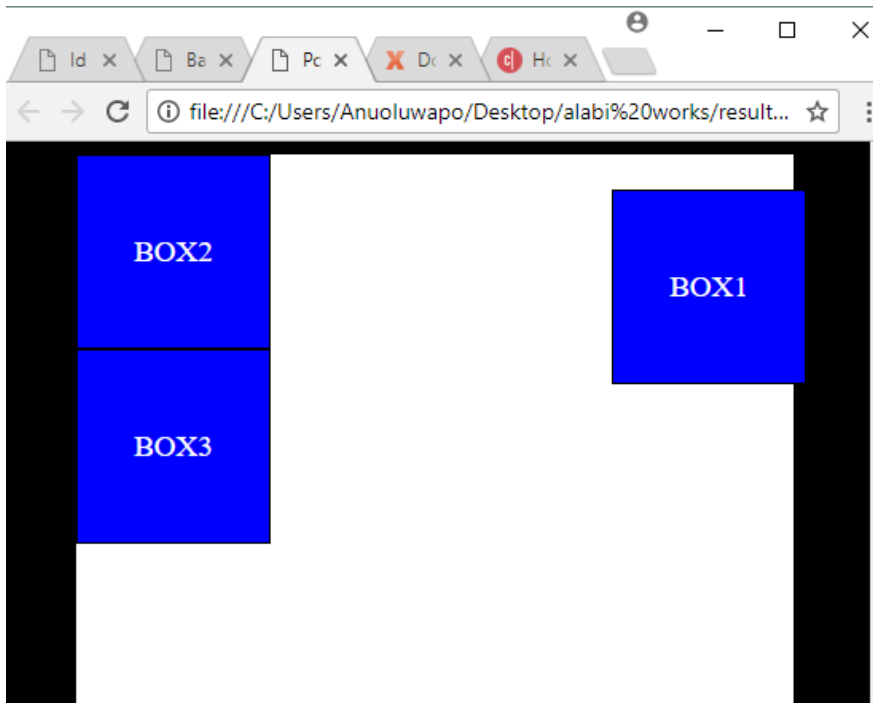
```
body{
    background-color: #000;
}

#container{
    background-color:#fff;
    width:450px;
    height:850px;
    position:relative;
    margin:auto;
}

.box1, .box2, .box3{
    background-color:blue;
    width:120px;
    height:120px;
    color:#fff;
    border:1px solid #000;
    text-transform: uppercase;
    font-size:19px;
    text-align: center;
    line-height:120px;
}

.box1{
    position:fixed;
    top:30px;
    right:40px;
}
```

**result:**



## Z-index

The z-index property specifies the stack order of an element. An element with greater stack order is always in front of an element with a lower stack order.

Note: z-index only works on positioned elements (position:absolute, position:relative, or position:fixed).

### html:

```
<body>

    <div id="container">
        <div class="box1">
            box1
        </div>
        <div class="box2">
            box2
        </div>
        <div class="box3">
            box3
        </div>
    </div>
```

</body>

### CSS:

```
body{
    background-color: #000;
}

#container{
    background-color:#fff;
    width:450px;
```

```

        height:450px;
        position:relative;
        margin:auto;
    }

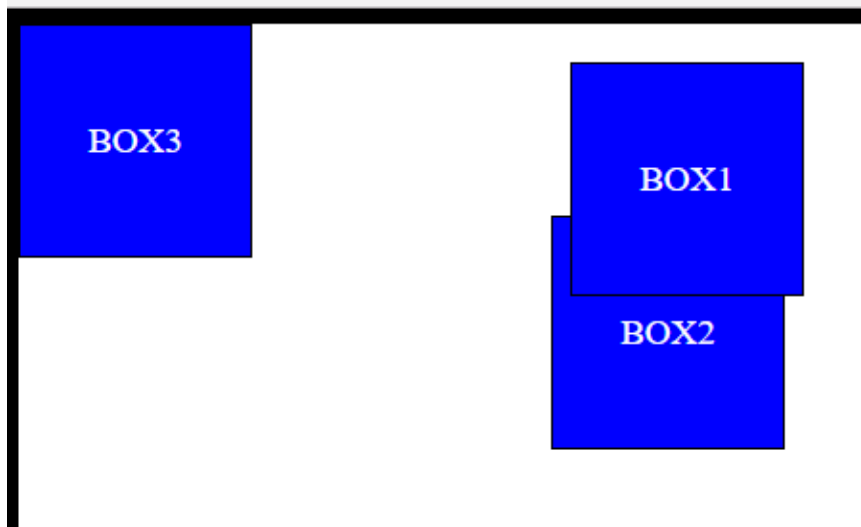
    .box1, .box2, .box3{
        background-color:blue;
        width:120px;
        height:120px;
        color:#fff;
        border:1px solid #000;
        text-transform: uppercase;
        font-size:19px;
        text-align: center;
        line-height:120px;
    }

    .box1{
        position:absolute;
        z-index:2;
        top:20px;
        right:40px;
    }

    .box2{
        position:absolute;
        z-index:1;
        top:100px;
        right:50px;
    }

```

result:



## Float

The float property allows you to take an element in normal flow and place it as far to the left or right of the containing element as possible. It is important to always clear floated elements by giving their parent a property of **overflow:hidden;**

result:



The float property can take the following values **none**, **left**, **right**, **initial**, **inherit**.

**none:** The element is not floated, and will be displayed just where it occurs in the text. This is default.

**left:** The element floats to the left.

**right:** The element floats to the right.

**initial:** Sets this property to its default value.

**inherit:** Inherits this property from its parent element.

## Before Float

html:

```
<div id="container">
  <div class="box1">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Cupiditate dignissimos porro minus ut nesciunt voluptates laudantium
perferendis. Possimus cupiditate, ab eligendi ipsa eveniet id iusto magnam.
Incidunt eligendi, molestias error.</p>
    <p>Accusantium, omnis, repellat. Odio nemo molestiae cumque
sit. Consectetur rem id tempore ducimus aliquid delectus soluta, blanditiis,
dignissimos at quam, eveniet nulla rerum maxime inventore corporis quasi sit
totam iusto.</p>
  </div>
  <div class="box2">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Voluptatem excepturi illum quisquam, sint iusto beatae quo in delentis
aut aliquid quas, aliquam rem veniam aspernatur dolorum atque qui quae
ea.</p>
    <p>Quia voluptatibus adipisci ipsa assumenda accusantium
blanditiis architecto quas magnam consequuntur ex dolorem fugit impedit eaque
minus quo delectus saepe doloremque atque nulla numquam laudantium voluptatum
facere, reprehenderit unde. Quaerat!</p>
  </div>
</div>
```

css:

```
body{
  background-color: #000;
```

```

        color:#fff;
    }

#container{
    background-color:#fff;
    width:600px;
    margin:auto;
}

.box1{
    width:200px;
    background-color:red;
}

.box2{
    width:150px;
    background-color:green;
}

```

result:



## After Float

It is obvious that the white containing element is missing in the below picture, this can be solved by giving the containing element an `overflow:hidden` or `overflow:auto` declaration.

**html:**

Using the above Html code

**CSS:**

```

body{
    background-color: #000;
    color:#fff;
}

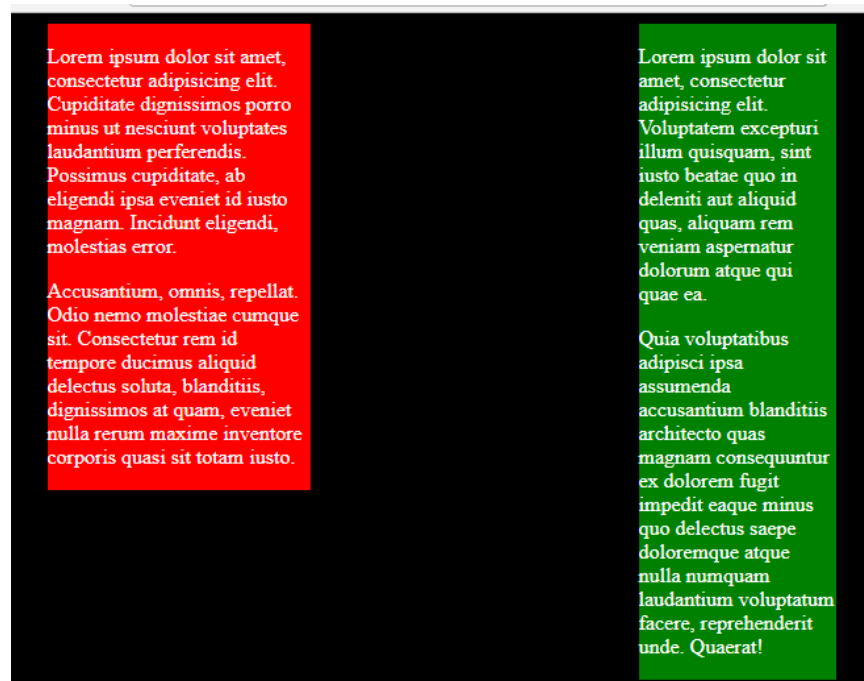
```

```
#container{
    background-color:#fff;
    width:600px;
    margin:auto;
}

.box1{
    width:200px;
    background-color:red;
}

.box2{
    width:150px;
    background-color:green;
}
```

result:



## Clearing Float

It is obvious that the white containing element is missing in the above picture, this can be solved by giving the containing element an **overflow:hidden** or **overflow:auto** declaration.

### CSS:

```
body{
    background-color: #000;
    color:#fff;
}

#container{
    background-color:#fff;
    width:600px;
    margin:auto;
}
```



```

        overflow: hidden;
    }

    .box1 {
        width: 200px;
        background-color: red;
        float: left;
    }

    .box2 {
        width: 150px;
        background-color: green;
        float: right;
    }

```

result:



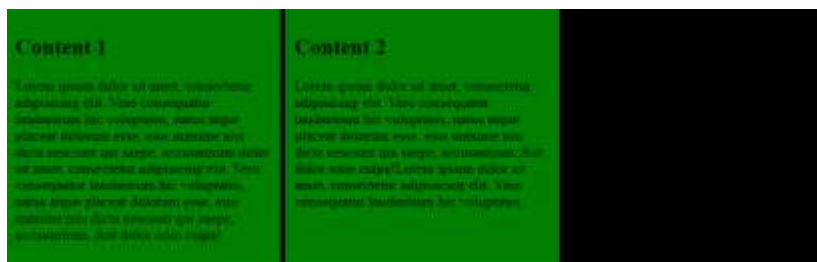
# Chapter 5

## Flex Box

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

## Flexbox Elements

To start using the Flexbox model, you need to first define a flex container.



The element above represents a flex container (the black area)

## Parent Element (Container)

The flex container becomes flexible by setting the **display** property to flex:

### CSS

```
#wrapper {  
  display: flex;  
}
```

### html

```
<div id="wrapper">  
  <div class="column">  
    <h2>Content 1</h2>  
    <p>  
      Lorem ipsum dolor sit amet, consectetur adipisicing  
      elit. Vero consequatur laudantium hic voluptates,  
      natus atque placeat dolorum esse, eius maxime nisi  
      dicta nesciunt qui saepe, accusantium.dolor sit amet,  
      consectetur adipisicing elit. Vero consequatur  
      laudantium hic voluptates, natus atque placeat  
      dolorum esse, eius maxime nisi dicta nesciunt qui  
      saepe, accusantium. Aut dolor odio culpa!  
    </p>  
  </div>  
  <div class="column">
```

```

        <h2>Content 2</h2>
        <p>
            Lorem ipsum dolor sit amet, consectetur adipisicing
            elit. Vero consequatur laudantium hic voluptates,
            natus atque placeat dolorum esse, eius maxime nisi
            dicta nesciunt qui saepe, accusantium. Aut dolor odio
            culpa!Lorem ipsum dolor sit amet, consectetur
            adipisicing elit. Vero consequatur laudantium hic
            voluptates,
        </p>
    </div>
</div>

```

## CSS

```

*{
    box-sizing: border-box;
}

body{
    background-color: #fff;
}

#wrapper{
    display: flex;
    width: 900px;
    background-color: #000;
}

.column{
    background-color: green;
    padding: 10px;
    width: 300px;
}

#wrapper > .column:nth-child(2){
    margin-left: 5px;
}

```

The flex container properties are as follows

- flex-direction
- flex-wrap
- justify-item
- align-item

## The flex-direction Property

The **flex-direction** property defines in which direction the container wants to stack the flex items. The flex-direction can take the following values: row, column, row-reverse, column-reverse.

### CSS

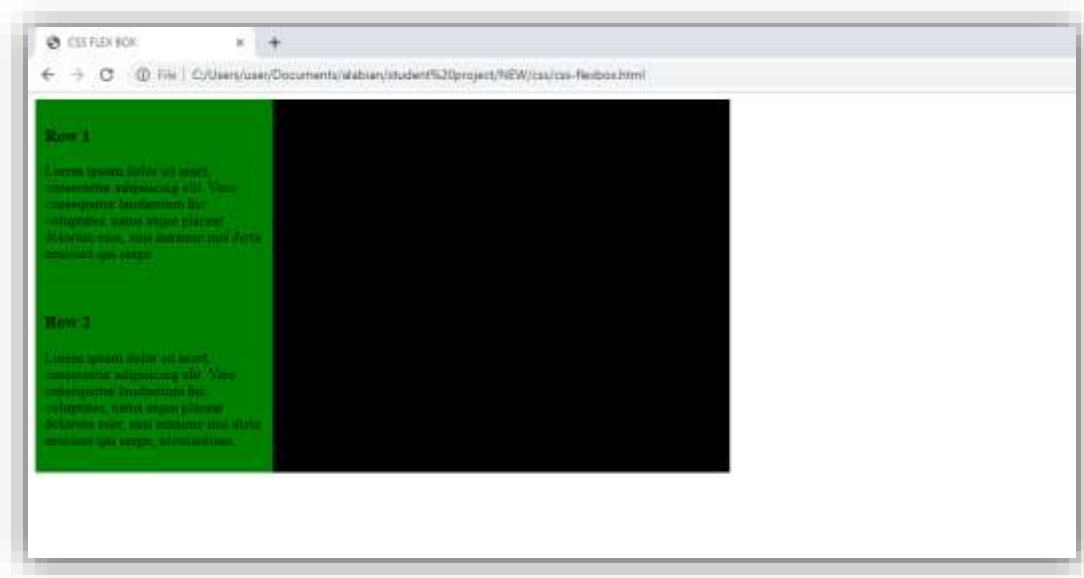
```
#wrapper{
    display: flex;
    width: 760px;
    background-color: #000;
    flex-direction: column;
}

.column{
    background-color: green;
    padding: 10px;
    width: 240px;
}
```

### html

```
<div id="wrapper">
  <div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
      elit. Vero consequatur laudantium hic voluptates,
      natus atque placeat dolorum esse, eius maxime nisi
      dicta nesciunt qui saepe
    </p>
  </div>
  <div class="column">
    <h3>Row 2</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
      elit. Vero consequatur laudantium hic voluptates,
      natus atque placeat dolorum esse, eius maxime nisi
      dicta nesciunt qui saepe, accusantium.
    </p>
  </div>
</div>
```

### Result



## The flex-wrap Property

The **flex-wrap** property specifies whether the flex items should wrap or not. The flex-wrap property can take the following values **wrap**, **nowrap**, **wrap-reverse**

### Wrap

The wrap value specifies that the flex items will wrap if necessary

### Nowrap

The nowrap value specifies that the flex items will not wrap (this is default)

### Wrap-reverse

The wrap-reverse value specifies that the flexible items will wrap if necessary, in reverse order

### html

```
<div id="wrapper">
  <div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
      elit. Vero consequat laudantium hic voluptates,
      natus
    </p>
  </div><div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
      elit. Vero consequat laudantium hic voluptates,
      natus
    </p>
  </div><div class="column">
    <h3>Row 2</h3>
    <p>
```

```

        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit. Vero consequatur laudantium hic voluptates,
        natus
    </p>

</div><div class="column">
    <h3>Row 3</h3>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit. Vero consequatur laudantium hic voluptates,
        natus
    </p>

</div>
<div class="column">
    <h3>Row 4</h3>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit. Vero consequatur laudantium hic voluptates,
        natus
    </p>

</div>
<div class="column">
    <h3>Row 5</h3>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit. Vero consequatur laudantium hic voluptates,
        natus
    </p>

</div>
<div class="column">
    <h3>Row 6</h3>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit. Vero consequatur laudantium hic voluptates,
        natus
    </p>

</div>

```

</div>

## CSS

```

#wrapper{
    display: flex;
    width: 800px;
    background-color: #fff;
    flex-wrap: wrap;
}

.column{
    background-color: green;
    padding: 10px;
    width: 190px;
    height: 190px;
}

```

## Result



## Justify Content

The **justify-content** property is used to align the flex items horizontally. The justify-content can take the following values **flex-start**, **flex-end**, **space-between**, **space-around**, **center**

### Flex start

The flex-start value aligns the flex items at the beginning of the container (this is default):

#### html

```
<div id="wrapper">
  <div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus
    </p>
  </div>
  <div class="column">
    <h3>Row 2</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus
    </p>
  </div>
  <div class="column">
    <h3>Row 3</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus
    </p>
  </div>
  <div class="column">
    <h3>Row 4</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus
    </p>
  </div>
  <div class="column">
    <h3>Row 5</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus
    </p>
  </div>
  <div class="column">
    <h3>Row 6</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus
    </p>
  </div>
</div>
```

```

    </div>
    <div class="column">
      <h3>Row 3</h3>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing
        elit. Vero consequatur laudantium hic voluptates,
        natus
      </p>
    </div>
  </div>

```

## CSS

```

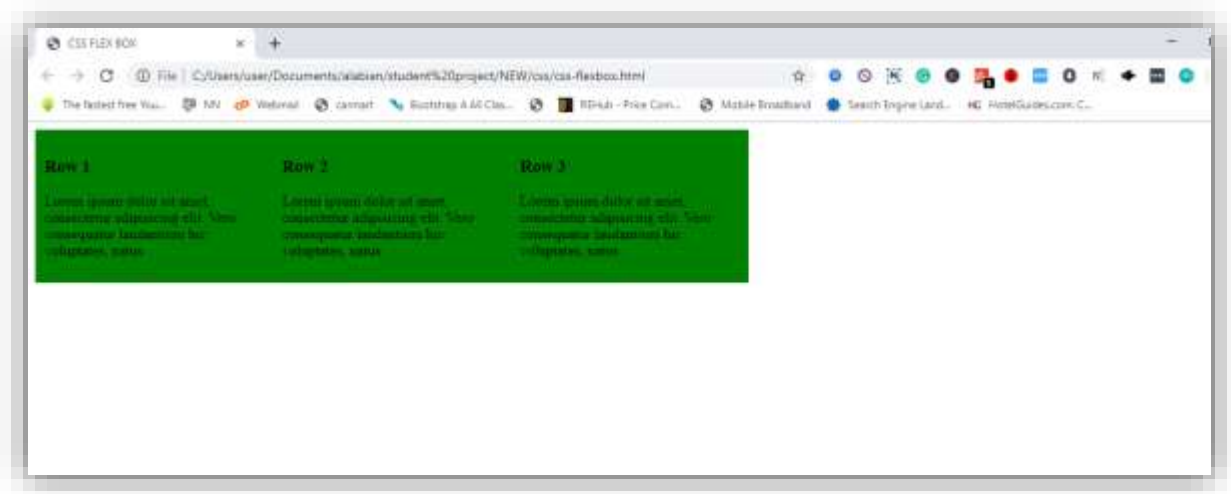
#wrapper{
  display: flex;
  width: 960px;
  background-color: #fff;
  justify-content: flex-start;
}

.column{
  background-color: green;
  padding: 10px;
  width: 240px;
}

```



## Result



## Center

The center value aligns the flex items at the center of the container:

## html

```
<div id="wrapper">
  <div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing
      elit. Vero consequatur laudantium hic voluptates,
      natus
    </p>
  </div>
  <div class="column">
    <h3>Row 2</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing
      elit. Vero consequatur laudantium hic voluptates,
      natus
    </p>
  </div>
  <div class="column">
    <h3>Row 3</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing
      elit. Vero consequatur laudantium hic voluptates,
      natus
    </p>
  </div>
</div>
```

## CSS

```
#wrapper{
  display: flex;
  width: 960px;
```

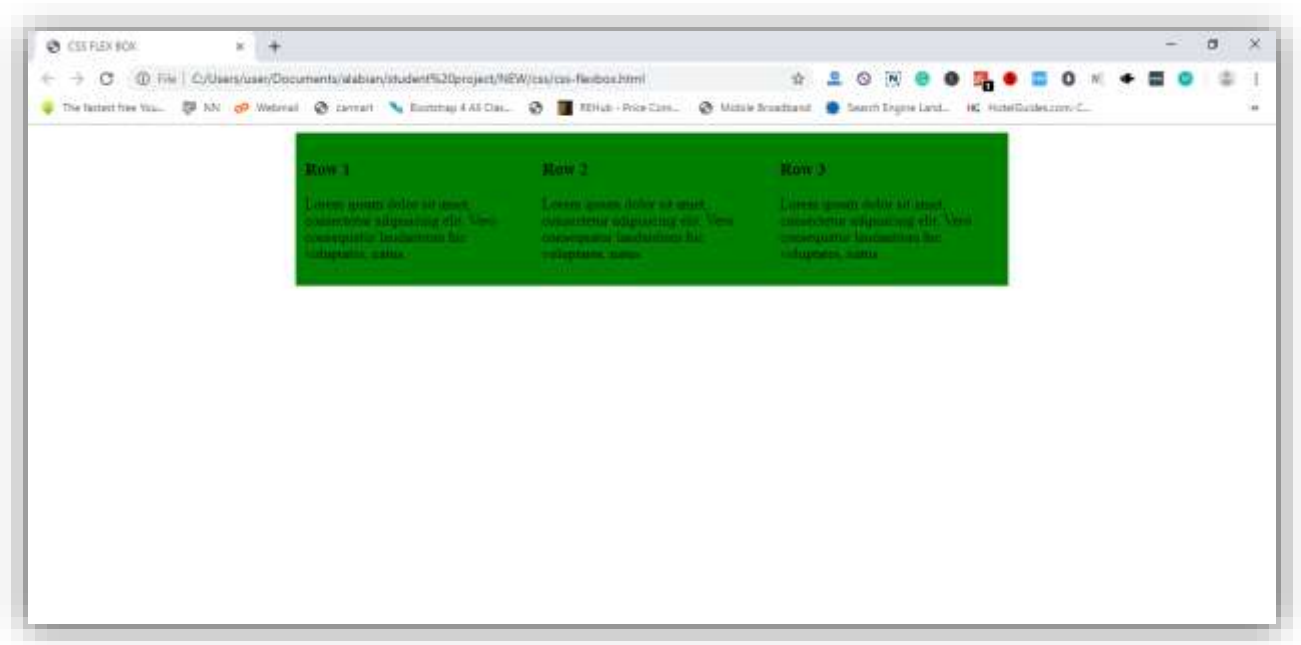
```

        background-color: #fff;
        justify-content: center;
    }

    .column{
        background-color: green;
        padding: 10px;
        width: 240px;
    }

```

## Result



## Flex-end

The flex-end value aligns the flex items at the end of the container:

## Space-around

The space-around value displays the flex items with space before, between, and after the lines:

## Space-between

The space-between value displays the flex items with space between the lines:

## The align-items Property

The **align-items** property is used to align the flex items vertically.

## Center

The center value aligns the flex items in the middle of the container

### html

```
<div id="wrapper">
  <div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
      elit. Vero consequatur laudantium hic voluptates,
      natus
    </p>
  </div>
</div>
```

### CSS

```
#wrapper{
  display: flex;
  width: 650px;
  height: 400px;
  background-color: #000;
  align-items: center;
}

.column{
  background-color: green;
  padding: 10px;
  width: 240px;
  height: 200px;
}
```

### result



### Flex-start

The flex-start value aligns the flex items at the top of the container

### Flex-end

The flex-end value aligns the flex items at the bottom of the container

### Stretch

The stretch value stretches the flex items to fill the container (this is default):

### Baseline

The baseline value aligns the flex items such as their baselines aligns

## Child Elements (Items)

The direct child elements of a flex container automatically becomes flexible (flex) items.

The flex item properties are:

- order
- flex-grow
- flex-basis
- flex

## Order

The **order** property specifies the order of the flex items. The first flex item in the code does not have to appear as the first item in the layout.

The order value must be a number, default value is 0.

### html

```
<div id="wrapper">
  <div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Vero consequatur laudantium hic voluptates, natus atque placeat dolorum
esse, eius maxime nisi dicta nesciunt qui saepe, accusantium. Aut dolor odio
culpa!Lorem
    </p>
  </div>
  <div class="column">
    <h3>Row 2</h3>
    <p>
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus atque placeat dolorum esse, eius maxime nisi dicta nesciunt qui saepe, accusantium. Aut dolor odio culpa!Lorem

```
        </p>
    </div>
    <div class="column">
        <h3>Row 3</h3>
        <p>
```

                    Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero consequatur laudantium hic voluptates, natus atque placeat dolorum esse, eius maxime nisi dicta nesciunt qui saepe, accusantium. Aut dolor odio culpa!Lorem

```
        </p>
    </div>
```

```
</div>
```

## CSS

```
body{
    background-color: #000;
    display: flex;
    justify-content: center;
}

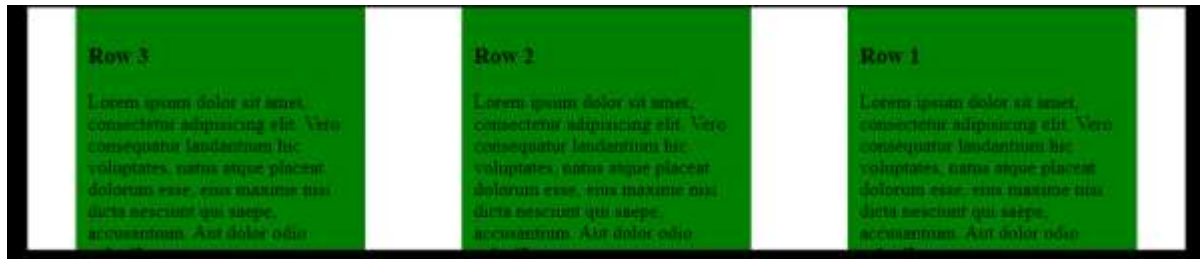
*{
    box-sizing: border-box;
}

#wrapper{
    display: flex;
    width: 960px;
    background-color: #fff;
    justify-content: space-around;
}

.column{
    background-color: green;
    padding: 10px;
    width: 240px;
    height: 200px;
}

#wrapper > .column:nth-child(1){
    order:3;
}
#wrapper > .column:nth-child(2){
    order:2;
}
#wrapper > .column:nth-child(3){
    order:1;
}
```

## result



## Flex-grow

The **flex-grow** property specifies how much a flex item will grow relative to the rest of the flex items.

## html

```
<div id="wrapper">
  <div class="column">
    <h3>Row 1</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Vero consequatur laudantium hic voluptates, natus atque placeat dolorum
esse, eius maxime nisi dicta nesciunt qui saepe, accusantium. Aut dolor odio
culpa!Lorem
    </p>
  </div>
  <div class="column">
    <h3>Row 2</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Vero consequatur laudantium hic voluptates, natus atque placeat dolorum
esse, eius maxime nisi dicta nesciunt qui saepe, accusantium. Aut dolor odio
culpa!Lorem
    </p>
  </div>
  <div class="column">
    <h3>Row 3</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Vero consequatur laudantium hic voluptates, natus atque placeat dolorum
esse, eius maxime nisi dicta nesciunt qui saepe, accusantium. Aut dolor odio
culpa!Lorem
    </p>
  </div>
</div>
```

## CSS

```
body{
  background-color: #000;
  display: flex;
  justify-content: center;
}
```

```

*{
    box-sizing: border-box;
}

#wrapper{
    display: flex;
    width: 1200px;
    background-color: #fff;
    justify-content: space-around;
}

.column{
    background-color: green;
    padding: 10px;
    width: 240px;
    height: 200px;
    overflow: hidden;
    margin-right: 10px;
}

#wrapper > .column:nth-child(1){
    flex-grow: 3;
}

```

result



## Flex-shrink

The **flex-shrink** property specifies how much a flex item will shrink relative to the rest of the flex items.

html

```

<div id="wrapper">
    <div class="column">
        1
    </div>
    <div class="column">
        2
    </div>
    <div class="column">
        3
    </div>
    <div class="column">
        4
    </div>
    <div class="column">
        5
    </div>
</div>

```

```

        </div>
        <div class="column">
            6
        </div>
        <div class="column">
            7
        </div>
        <div class="column">
            8
        </div>
    </div>

```

## CSS

```

body{
    background-color: #000;
    display: flex;
    justify-content: center;
}

*{
    box-sizing: border-box;
}

#wrapper{
    display: flex;
    width: 700px;
    background-color: #fff;
}

.column{
    background-color: green;
    color: white;
    width: 100px;
    margin: 10px;
    text-align: center;
    line-height: 75px;
    font-size: 30px;
}

.column:nth-child(5){
    flex-shrink: 0;
}

```

## result



## Flex-basis

The **flex-basis** property specifies the initial length of a flex item.





## html

```
<div id="wrapper">
  <div class="column">
    1
  </div>
  <div class="column">
    2

  </div>
  <div class="column">
    3

  </div>
  <div class="column">
    4
  </div>

</div>
```

## CSS

```
body{
    background-color: #000;
    display: flex;
    justify-content: center;
}

*{
    box-sizing: border-box;
}

#wrapper{
    display: flex;
    width: 500px;
    background-color: #fff;
}

.column{
    background-color: green;
    color: white;
    width: 100px;
    margin: 10px;
    text-align: center;
    line-height: 75px;
    font-size: 30px;
}

.column:nth-child(3){
    flex-basis: 200px;
}
```

result



## Flex

The **flex** property is a shorthand property for the **flex-grow**, **flex-shrink**, and **flex-basis** properties.

## html

```
<div id="wrapper">
  <div class="column">
    1
  </div>
  <div class="column">
    2
  </div>
  <div class="column">
    3
  </div>
</div>
```

## CSS

```
body{
    background-color: #000;
    display: flex;
    justify-content: center;
}

*{
    box-sizing: border-box;
}

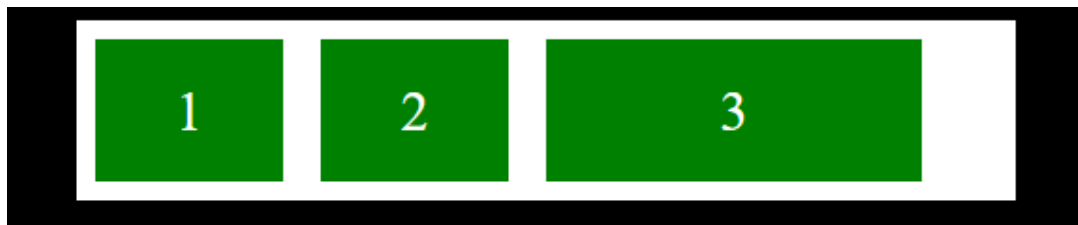
#wrapper{
    display: flex;
    width: 500px;
}
```

```
        background-color: #fff;
    }

    .column{
        background-color: green;
        color: white;
        width: 100px;
        margin: 10px;
        text-align: center;
        line-height: 75px;
        font-size: 30px;
    }

    .column:nth-child(3){
        flex:0 0 200px;
    }
}
```

result



# Chapter 6

## Page layout

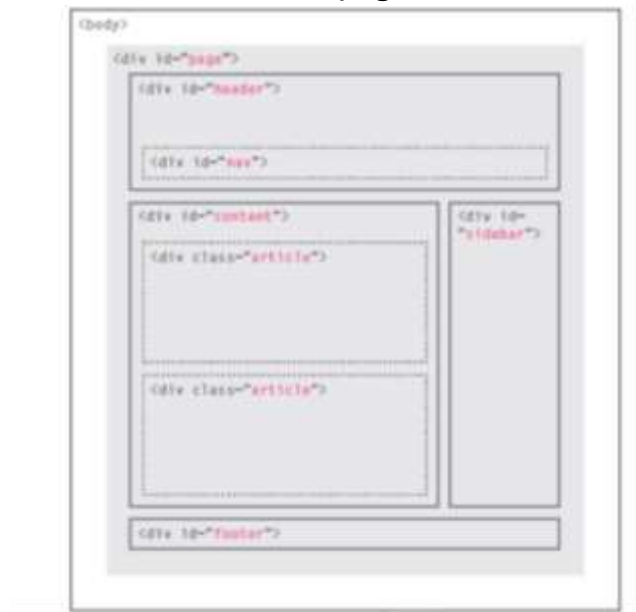
With all these CSS properties combined, you can create nice looking non-linear page layouts. Before designing a website it is important to do the following

**Determine the width of the website:** Before you start designing or planning out your website first all determine the width of the website.

**Sketch or create mockup of your design:** Sketch your ideas using pencil and piece of paper or if you are comfortable with Photoshop or any other graphic tool you can use it to create a web mockup or prototype of the website.

## Traditional HTML Layouts

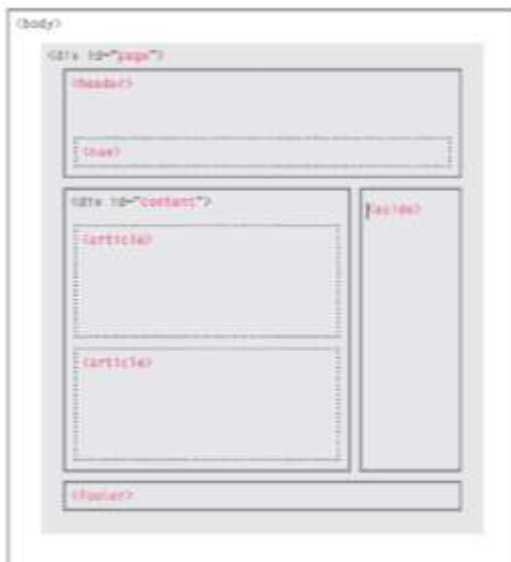
For a long time, web page authors used `<div>` elements to group together related elements on the page (such as the elements that form a header, an article, footer or sidebar). Authors used class or id attributes to indicate the role of the `<div>` element in the structure of the page.



## New Html 5 Layout Elements

HTML5 introduces a new set of elements that allow you to divide up the parts of a page. The names of these elements indicate the kind of content you will find in them. They are still subject to change, but that has not stopped many web page authors using them already.

- `<header>` `<footer>`
- `<nav>`
- `<article>`
- `<aside>`
- `<section>`
- `<hgroup>`
- `<figure>` `<figcaption>`



## Fixed width Layout

This is when you define the width of layout element in **pixels**. Fixed-width layouts are very common, as they give the developer more control over layout and positioning. If you set the width of your design to be 960 pixels wide, it will always be 960 pixels. Knowing the exact width of each element allows you to lay them out precisely and know where everything will be. This makes fixed-width layout the easiest and therefore most common approach.

However, fixed-width designs have their downsides. First, because they are fixed, they are always the same size no matter what your window size. As such, they don't make good use of the available space.

On large screen resolutions, designs created for 1024×760 can appear tiny and lost in the middle of the screen. Conversely, a design created for a 1024×760 screen will cause horizontal scrolling (or crawling) on smaller screen resolutions. With an increasingly diverse range of screen sizes to contend with, fixed-width designs don't adapt well to the flexible nature of the Web. As such, they often feel like a poor compromise.

### Ideal width for fixed width website

- **960** is divisible by 2, 3, 4, 5, 8, 8, 10, 12, 15 and 16.
- **1020** is divisible by 2, 3, 4, 5, 6, 10, 12, 15 but not 8 and 16. It's not much wider than 960.
- **1040** is divisible by 2, 4, 5, 8, 10, 16 but not 3, 12, or 15. Yet it has a reasonable width that sits somewhere between the lower end of 960 and higher end of users browsing full screen.
- **1080** is divisible 2, 3, 4, 5, 6, 8, 10, 12, 15 but oddly enough, not 16.

### Liquid layout

With liquid layouts, dimensions are set using percentages instead of pixels. This allows liquid layouts to scale in relation to the browser window. As the browser window gets bigger, the columns get wider. Conversely, as the window gets smaller, the columns will reduce in width. Liquid layouts make for very efficient use of space, and the best liquid layouts aren't even noticeable.

However, liquid layouts are not without their own problems. At small window widths, line lengths can get incredibly narrow and difficult to read. This is especially true in multicolumn layouts. As such, it may be worth adding a **min-width** in pixels or ems to prevent the layout from becoming too narrow. However, set the **min-width** too large and your liquid designs inherit the same constraints as their fixed-width cousins.

## Web Page

