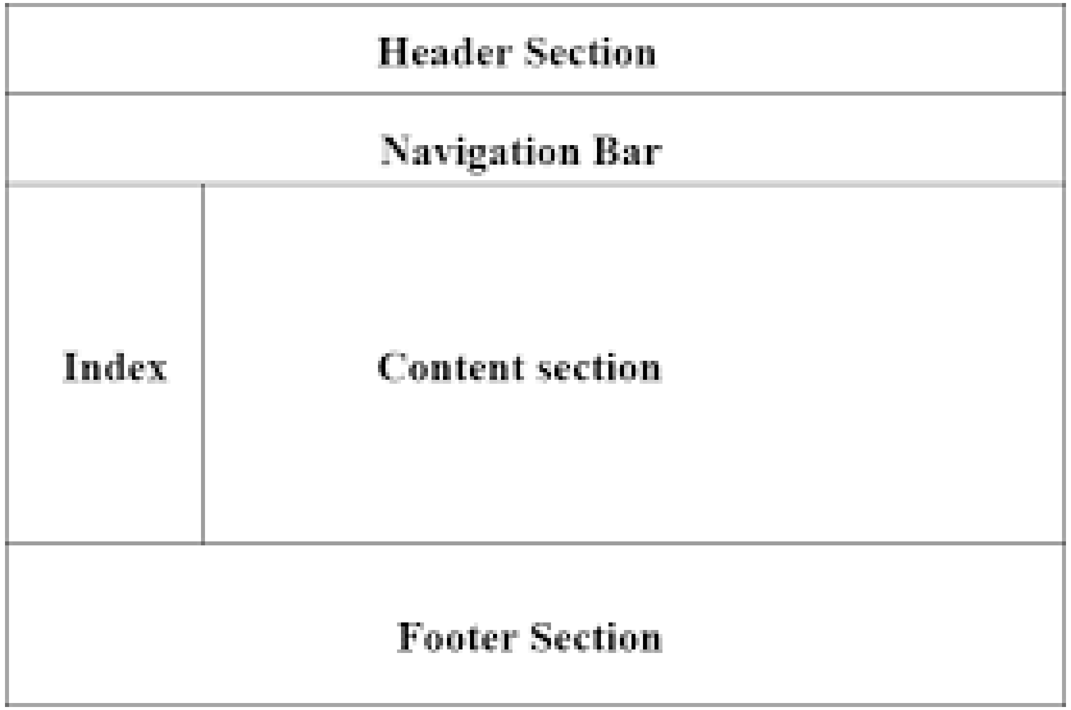
Basic Layout

- The HTML Layouts specifies the **arrangement of components** on an HTML web page



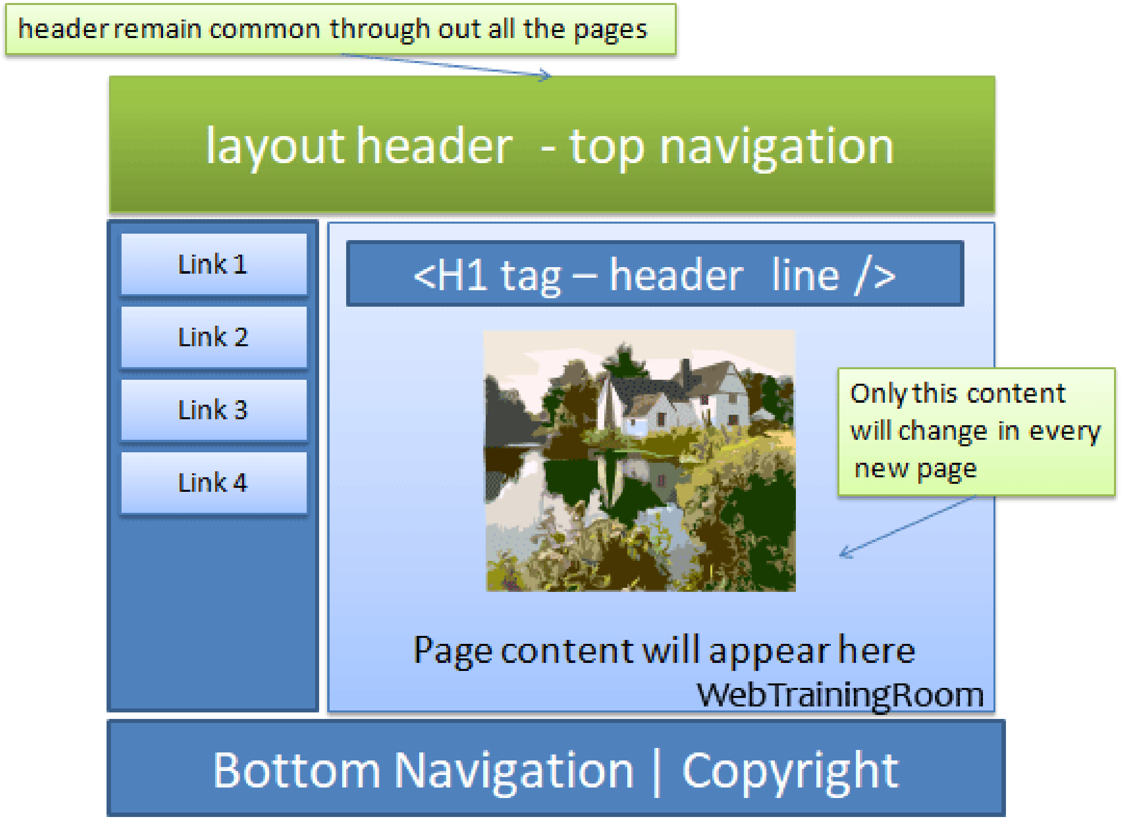
Header Section

Navigation Bar

Index

Content section

Footer Section



header remain common through out all the pages

layout header - top navigation

Link 1

Link 2

Link 3

Link 4

<H1 tag – header line />

Only this content will change in every new page

Page content will appear here

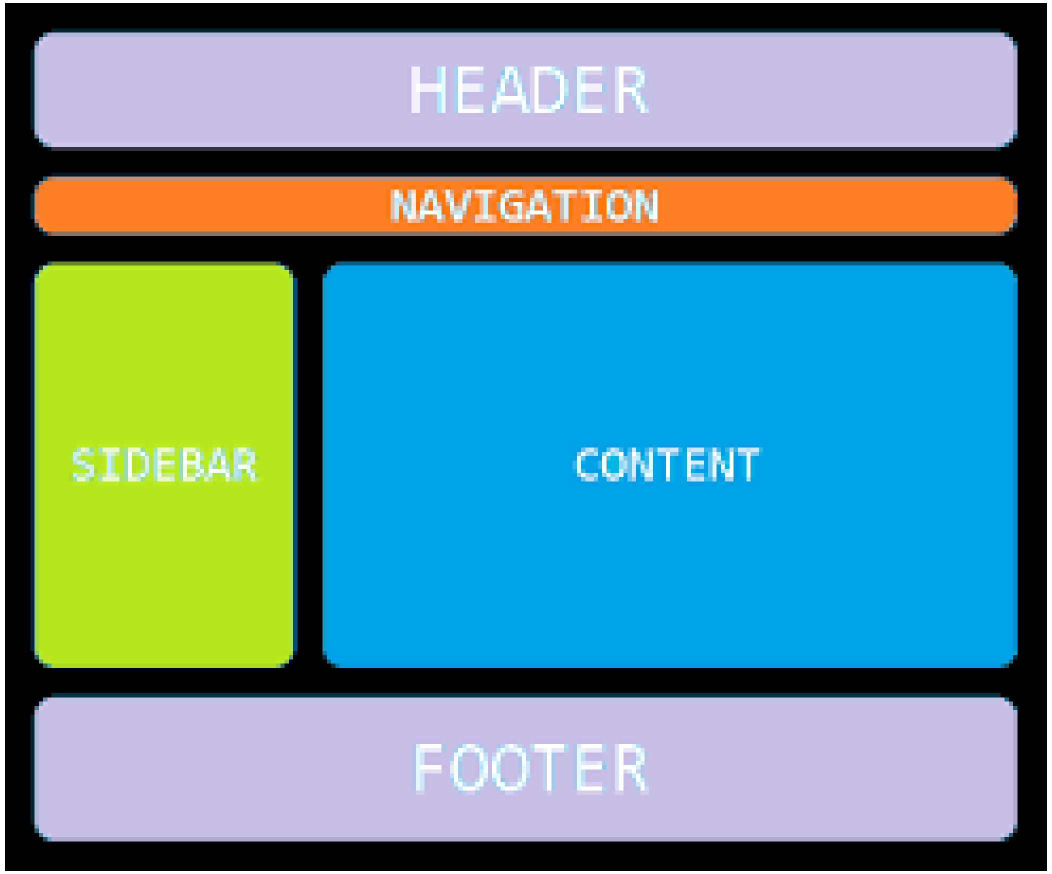
WebTrainingRoom

Bottom Navigation | Copyright

Basic Layout

HTML Layout Elements:

- **<header>** - Defines **a header for a document**
- **<nav>** - Defines **a set of navigation links**
- **<section>** - Defines **a section in a document**
- **<footer>** - Defines **a footer for a document**
- **<article>** - Defines an independent, self-contained content
- **<aside>** - Defines content aside from the content (like a sidebar)
- **<details>** - Defines additional details that the user can open and close on demand
- **<summary>** - Defines a heading for the <details> element

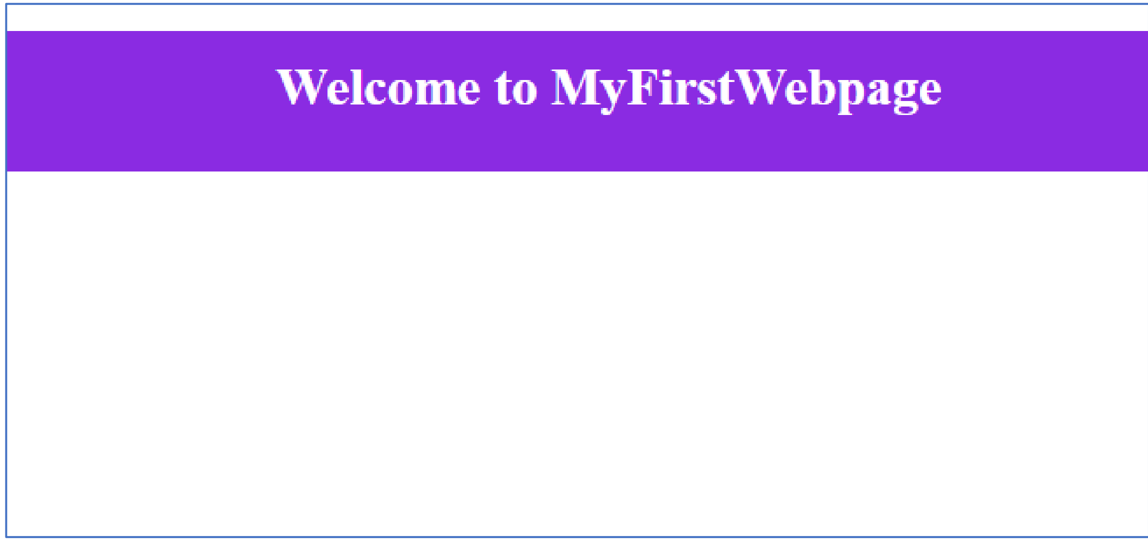


Basic Layout

<header> - Defines a header for a document or a section

- **The <header> element is used to create header section of web pages. The header contains the introductory content, heading element, logo or icon for the webpage, and authorship information.**

```
<!DOCTYPE html>
<html>
  <head>
                <title>First Webpage</title>
  </head>
 <body>
      <header style="background-color: #303030; height: 80px;width:
        100%">
                <h1 style="font-size: 30px; color: white;text-align: center;
padding-top: 15px;">
                Welcome to MyFirstWebpage</h1>
   </header>
 </body>
</html>
```



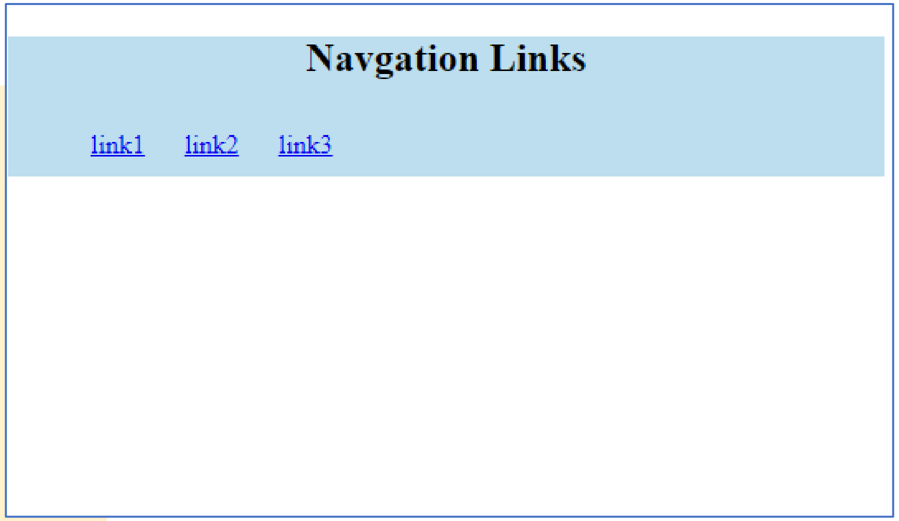
Welcome to MyFirstWebpage

Basic Layout

HTML <nav>

- The <nav> elements is a container for the **main block of navigation links**.
- It can contain links for the same page or for other pages.

```
<html>
<body>
<nav style="background-color:#bcdeef;">
  <h1 style="text-align: center;">Navgation Links</h1>
    <ul>
        <li style="display: inline-block; padding: 10px;">
               <a href="#">link1</a>
        </li>
        <li style="display: inline-block; padding: 10px;">
                      <a href="#">link2</a>
        </li>
        <li style="display: inline-block; padding: 10px;">
               <a href="#">link3</a>
        </li>
      </ul>
    </nav>
 </body>
</html
```



Navgation Links

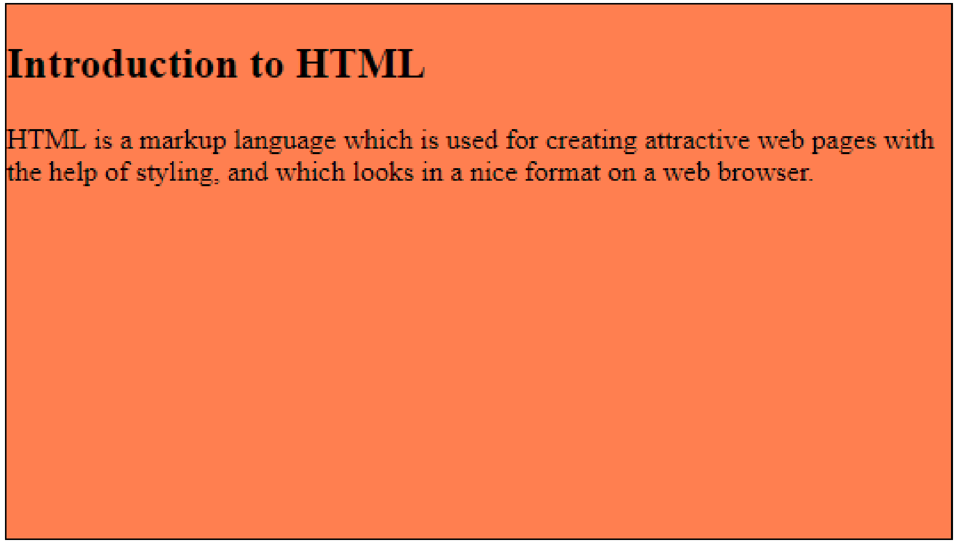
link1 link2 link3

Basic Layout

HTML <section>

- HTML <section> elements represent a **separate section of a web page** which contains related element grouped together.
- It can contain: text, images, tables, videos, etc

```
<!DOCTYPE html>
<html>
<head>
        <title>Page Section</title>
 </head>
  <body>

   <section style="background-color:#ff7f50; height:300px;width: 100%;
border: 1px      solid black;">
        <h2>Introduction to HTML</h2>
        <p>HTML is a markup language which is used for creating
        attractive web pages with the help of styling, and which looks
        in a nice format on a web browser.</p>
   </section>
  </body>
</html>
```



Introduction to HTML

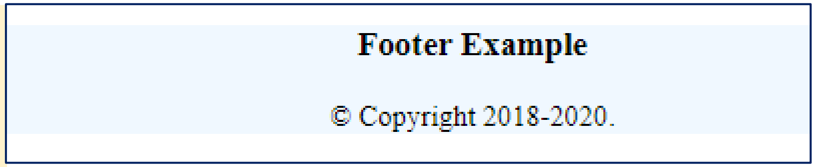
HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser.

Basic Layout

HTML <footer>

- HTML <footer> element defines the **footer for that document or web page**.
- It mostly contains information about author, copyright, other links, etc.

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
 <head>
    <title>Footer Section</title>
 </head>
  <body>
  <footer style="background-color:#f0f8ff; width: 100%; text-align: center;">
    <h3>Footer Example</h3>
    <p>© Copyright 2022-2023. </p>
   </footer>
  </body>
</html>
```



Footer Example

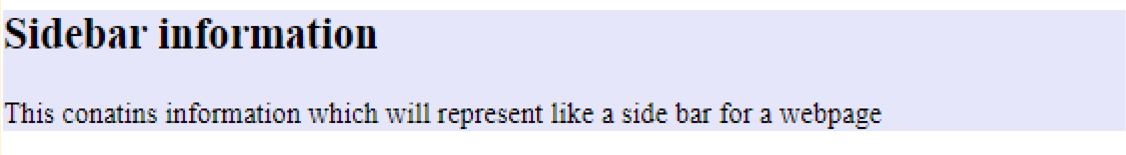
© Copyright 2018-2020.

Basic Layout

HTML <aside>

- HTML <aside> define aside content related to primary content. The <aside> content must be related to the primary content. It can function as side bar for the main content of web page.

```
<!DOCTYPE html>
<html>
<head>
        <title>Aside Example</title>
</head>
<body>
  <aside style="background-color:#e6e6fa">
        <h2>Sidebar information</h2>
        <p>This conatins information which will represent like a side
bar for a webpage</p>
  </aside>
</body>
</html>
```



Sidebar information

This conatins information which will represent like a side bar for a webpage

Flow of HTML

In HTML, the flow refers to the way elements are positioned and arranged on a web page.

- The flow of element → **the order in which they appear** in the HTML document and the **type of positioning** applied to them.

The **default flow** → **normal flow**, and it follows a **top-to-bottom, left-to-right** order.

- In this flow, **ELEMENTS ARE POSITIONED ONE AFTER ANOTHER**, starting from the **top-left corner to bottom- right as necessary.**

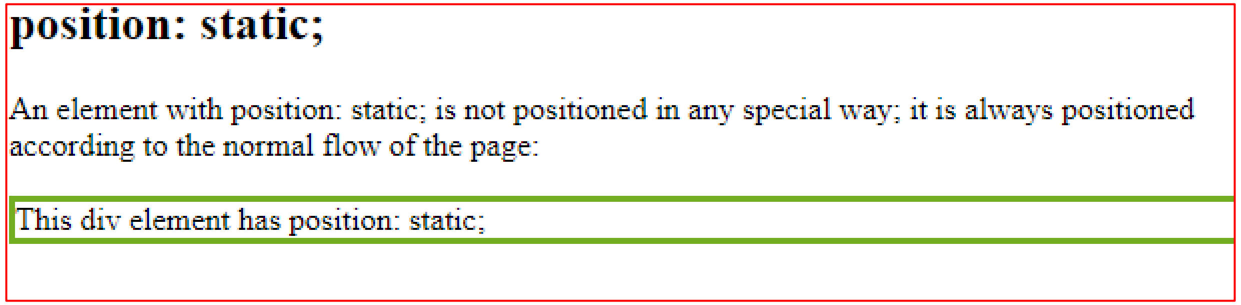
HTML elements can be **positioned in following ways**:

1. **Static:** Elements are positioned in the **normal flow** of the page.
2. **Relative:** Elements are **positioned relative to their normal position in the flow**. They can be moved up, down, left, or right using the top, bottom, left, and right properties.
3. **Absolute:** Elements are **positioned relative to their nearest positioned ancestor**.
4. Z-index: The z-index property **specifies the stack order of an element.**

Flow of HTML- static flow

i). position: static;

- HTML elements are positioned static by default.
- Static positioned elements are **not affected by the top, bottom, left, and right properties**.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:
- This <div> element has position: static



position: static;

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

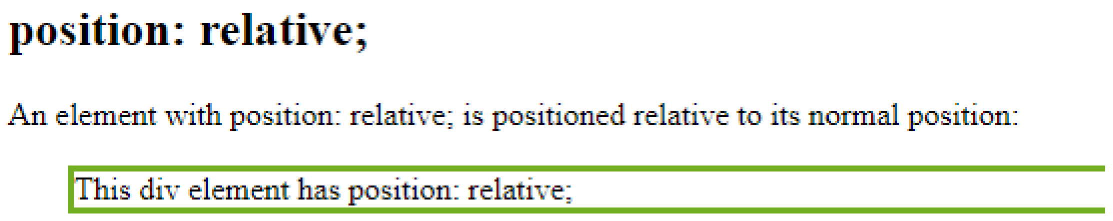
This div element has position: static;

```
<html>
<head>
<style>
div.static {
  position: static;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>position: static;</h2>
<p>An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:</p>
<div class="static">
        This div element has position: static;
</div>

</body>
</html>
```

Flow of HTML- static flow

ii). position: relative;

- positioned **relative to its normal position.**
- We can set relative positions **top, right, bottom, and left**
- **These properties** of a relatively-positioned element will cause it to be adjusted away from its normal position.



position: relative;

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

This div element has position: relative;

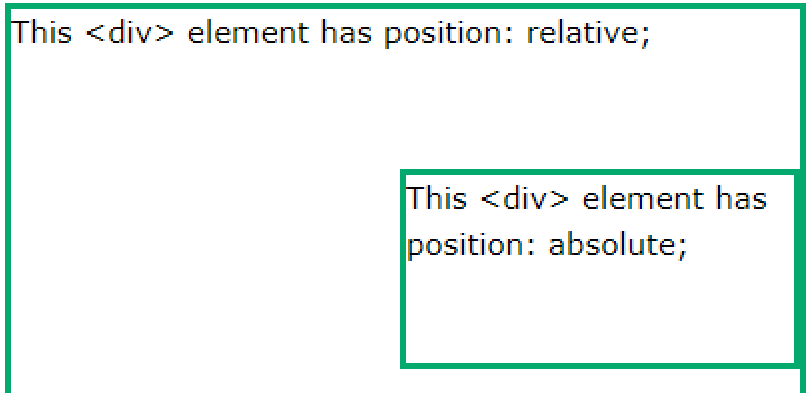
```
<html>
<head>
<style>
 div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>position: relative;</h2>
<p>An element with position: relative; is positioned relative
to its normal position:</p>

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

</body>
</html>
```

Flow of HTML

iii). position: absolute;

- An element with position: absolute; is positioned relative to the nearest positioned ancestor.
- However; if an **absolute positioned element has no positioned ancestors**, it uses the document body, and moves along with page scrolling.
- Absolute positioned elements are removed from the normal flow, and can overlap elements.

```
<html>
<head>
<style>
div.relative {
  position: relative;   width: 400px;
  height: 200px;   border: 3px solid #73AD21;
}
div.absolute {
  position: absolute;   top: 80px;   right: 0;   width: 200px;
  height: 100px;   border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>position: absolute;</h2>
<div class="relative">This div element has position: relative;
<div class="absolute">This div element has position:
absolute;</div>
</div>
</body>
</html>
```



This <div> element has position: relative;

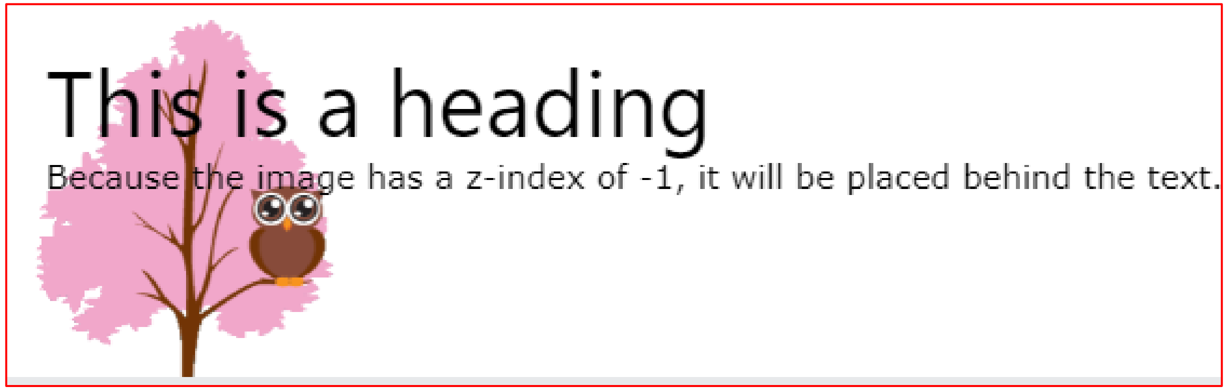
This <div> element has position: absolute;

Flow of HTML

iv). The z-index Property

- When elements are positioned, they can overlap other elements.
- The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).
- An element can have a positive or negative stack order.

```
<html>
<head>
<style>
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<img src="img_tree.png">
<p>Because the image has a z-index of -1, it will be placed behind the text.</p>

</body>
</html>
```



This is a heading

Because the image has a z-index of -1, it will be placed behind the text.

INLINE DISPLAY -- display: inline

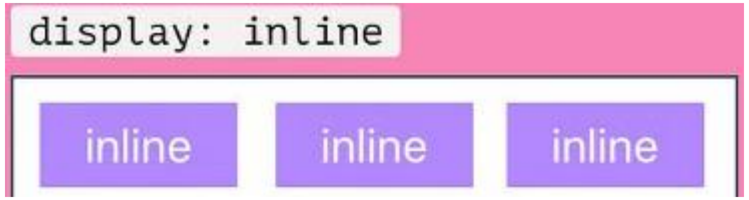
- Inline display in HTML refers to a **way of displaying an element on a web page**, where the element is **displayed on the same line** as its adjacent content.

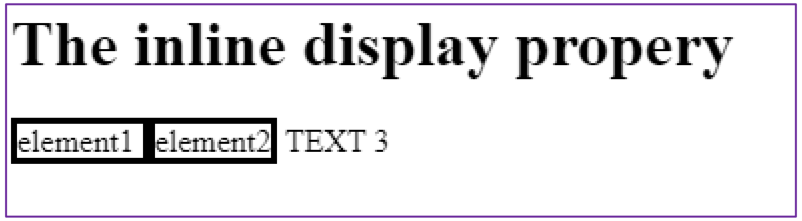
- **Displays an element as an inline element** (like <span>). Any **height and width properties will have no effect**

- To set an element to display inline in HTML, you can use the CSS "display" property and set its value to "inline"

  display:inline

```
<html>
<head>
<style>
  .in {
        display: inline;
        border-style: solid;
        }
</style>
</head>
<body>
<h1>The  inline display propery</h1>
<p class="in">element1 </h2>
<p class="in">element2</p> TEXT 3
</div>
</body>
</html>
```



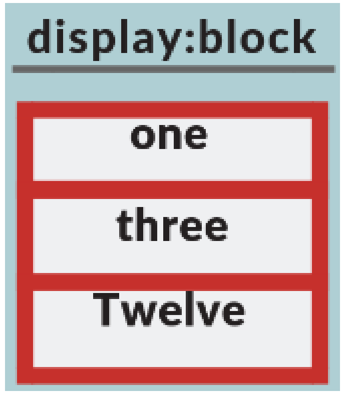


The inline display propery

element1 element2 TEXT 3

BLOCK DISPLAY -- display: block

- A block-level **display** **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).

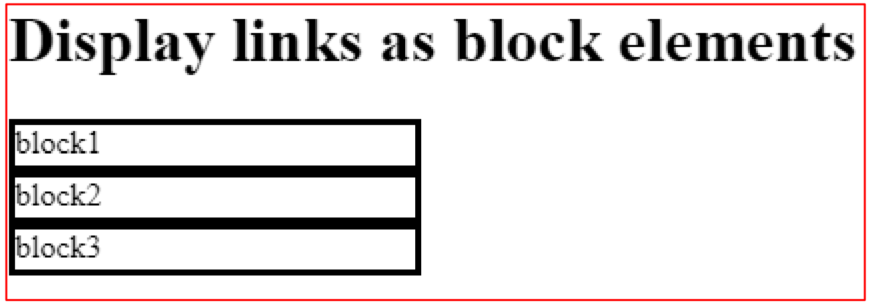

display:block

one

three

Twelve

Examples of block-level elements:
<div>
<h1> - <h6>
<p>
<form>
<header>
<footer>
<section>


Display links as block elements

block1

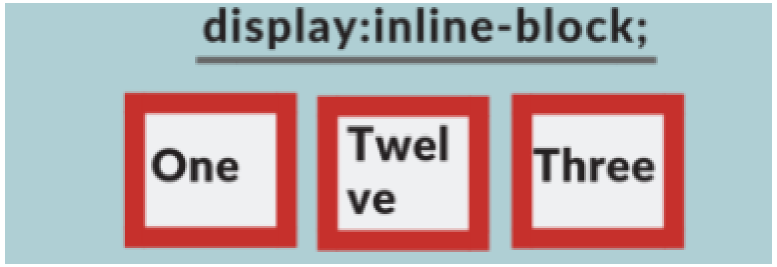
block2

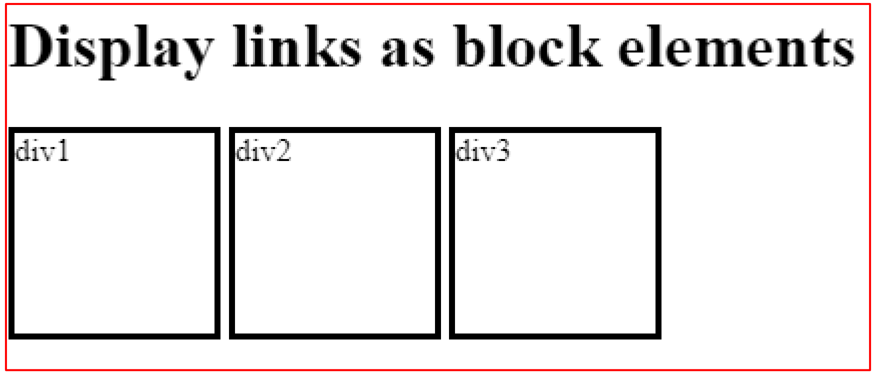
block3

```
<html>
<head>
<style>
div{
 display:block;
 height: 20px;
 width: 200px;
 border-style: solid;
}
</style>
</head>
<body>
<h1>Display links as block elements</h1>
<div>block1</div>
<div>block2</div>
<div>block3</div>
</body>
</html>
```

INLIE-BLOCK DISPLAY- display: inline-block

- Element is **displayed as an inline-level element**, **with a width, height, margin, and padding** like a block-level element





Display links as block elements

div1 div2 div3

```
<html>
<head>
<style>
div {
  display: inline-block;
  height:100px;
  width:100px;
  border-style:solid;
}
</style>
</head>
<body>
<h1>Display links as block elements</h1>
<div> div1
</div>
<div>div2</div>
<div>div3</div>
</body>
</html>
```

CSS Layout - float and clear

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

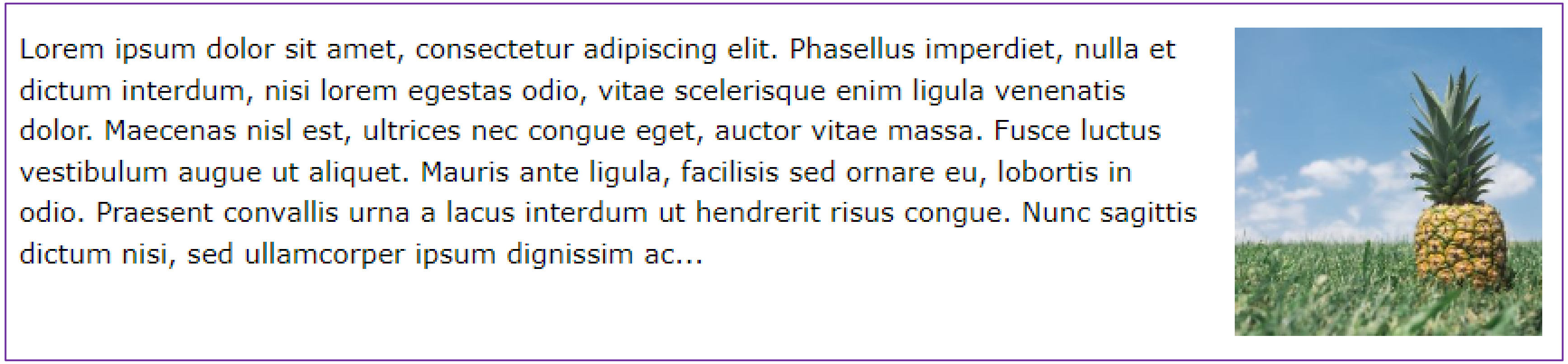
The float Property

- The **float property** is used **for positioning and formatting content**
- e.g. let an image **float left** to the text in a container.

FLAOT PROPERY VAUES :

  - **left** - The element **floats to the left of its container**
  - **right -** The element **floats to the right of its container**
  - **none** - The element **does not float .** This is default
  - **inherit** - The element **inherits the float value of its parent**

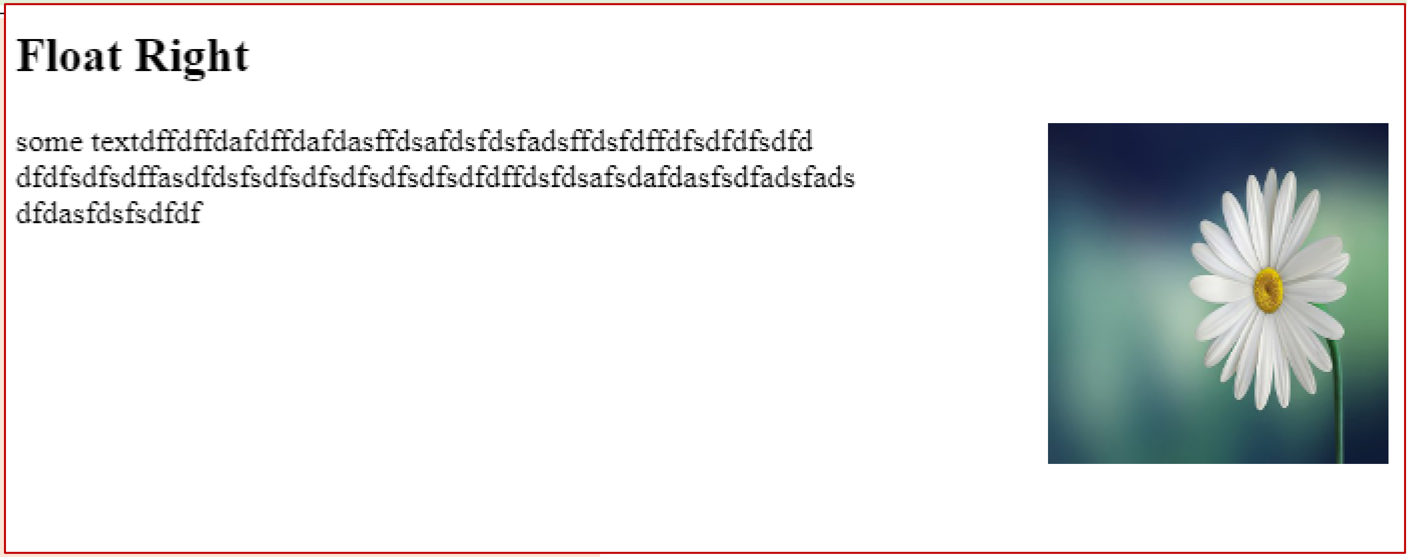
float: right;



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Float right

```
<html>
<head>
<style>
img {
  float: right;
}
</style>
</head>
<body>
<h2>Float Right</h2>
<p><img src="flower1.jpg" alt="Flower"
    style="width:170px;height:170px;margin-left:15px;">
some textdffdffdafdffdafdasffdsafdsfdsfadsffdsfdffdfsdfdfsdfd
Dfdfsdfsdffasdfdsfsdfsdfsdfsdfsdfsdfdffdsfdsafsdafdasfsdfads
Fads dfdasfdsfsdfdf</p>
</body>
</html>
```



Float Right

some textdffdffdafdffdafdasffdsafdsfdsfadsffdsfdffdfsdfdfsdfd
dfdfsdfsdffasdfdsfsdfsdfsdfsdfsdfsdfdffdsfdsafsdafdasfsdfadsfads
dfdasfdsfsdfdf

CSS Layout - clear

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

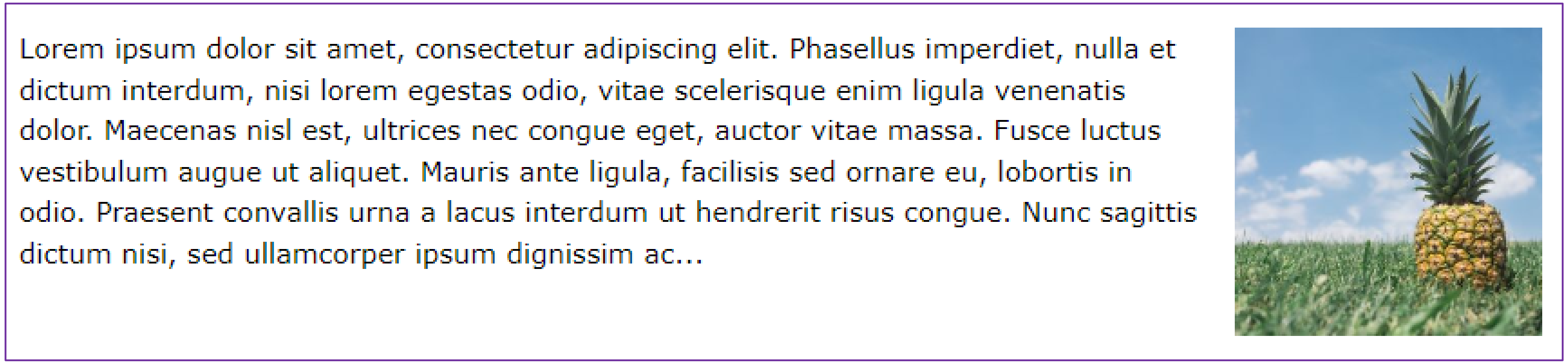
The float Property

- The **float property** is used **for positioning and formatting content**
- e.g. let an image **float left** to the text in a container.

FLAOT PROPERY VAUES :

  - **left** - The element **floats to the left of its container**
  - **right -** The element **floats to the right of its container**
  - **none** - The element **does not float .** This is default
  - **inherit** - The element **inherits the float value of its parent**

float: right;

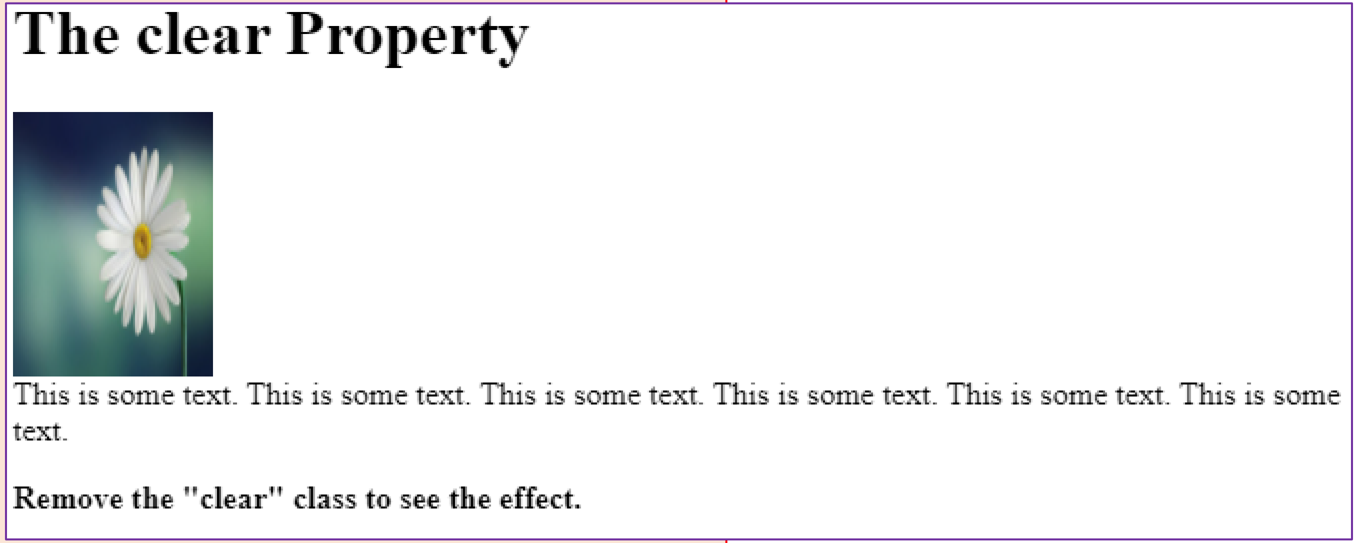


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

CSS Layout - float and clear

- The **clear** property controls the flow next to floated elements.
- The **clear property** specifies **what should happen with the element that is next to a floating element.**

```
<html>
<head>
<style>
img {
  float: left;
}
p.clear {
  clear: left;
}
</style>
</head>
<body>
<h1>The clear Property</h1>
    <img src=“flower1.jpeg" width="100" height="132">
<p class="clear">
This is some text. This is some text. This is some text. This is
some text. This is some text. This is some text.</p>
<p><strong>Remove the "clear" class to see the effect.</strong></p>
</body>
</html>
```



The clear Property

This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

Remove the "clear" class to see the effect.