# CSS Layout

There are about five to six different types of schooling for CSS Layout:

1. Normal Flow- simple bullet point or numeric layer of the information, pass ADA compliant, etc...

These is no graphic layout to this flow.

1. The display property
2. Flexbox
3. Grid
4. Floats
5. Positioning
6. Mobile Column Layout

The direction in which block element contents are laid out is described as the Block Direction. The Block Direction run vertically in a language such as English, which has a horizontal writing mode. It would run horizontal in any language with a Vertical Writing mode, such as Japanese. The corresponding Inline Direction is the direction which inline contents (such as sentence) would run.

## **The display property – standard values such as blocks, inline or inline block can change how element behave in a normal flow**

## **Floats- Applying a float value such as left can cause block level element to wrap alongside one side of an element, like the way images sometimes have text floating around in magazine layouts.**

The elements is moved to the left or right and removed from the normal flow, and the surrounding content floats around the floated item.

The float property has four possible values:

* left- Floats the element to the left.

**Code:**

<!DOCTYPE html>

<html>

<head>

<style>

.box{

float: left;

width: 150px;

height: 150px;

margin-right: 10px;

border: 4px solid green;

}

</style>

</head>

<body>

<h1> Simple Float Example </h1>

<div class = "box"> Float or Picture </div>

<p>This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.This sample text, more sample text.

</body>

</html>



* right- Floats the element to the right.
* none- Specifies no floating at all. This is the default value.
* inherit- Specifies that the value of the float property should be inherited from the element’s parent element

-------------------------------------------------------------------------------------------------------------------------------------

.wrapper{

display:flex;

}

/\*Add flex property to all our child items, with a value of 1 This will cause all the items to grow and fill the container, rather than leaving a space at the end. If there ar more space then the item will become wider, if there are less space they will become narrower. In addition, if you add another element to the markup the items will all become simaller to make space for it.They will adjust the size to take up the same amount of space.\*/

.wrapper> div{

flex: 1;

}

.box1{

border: 4px solid #666;

padding:0.5em;

}

.box2{

border: 4px solid #000001;

padding:0.5em;

}

.box3{

border: 4px solid blue;

padding:0.5em;

}

</style>

</head>

<body>

<div class= "wrapper">

<div class= "box1"> One </div>

<div class= "box2"> Two </div>

<div class= "box3"> Three </div>

</div>

## **Flex**

## **Grid**

While the flexbox is designed for one-dimensional layout. Grid designed two dimensional – lining things up in rows and columns.

Ex: MS Office Excel

You can switch on the Grid Layout with a specific value of display-



