LAB#04

# Object: CSS Layout - The position Property

The position property specifies the type of positioning method used for an element.

There are five different position values:

* static
* relative
* fixed
* absolute
* sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

**FLEXBOX LAYOUT**

One-dimensional flexible layout (hence the name ‘flex’).

Regular layouts (block which is vertically-based and inline which is horizontally-based) lack flexibility to support large or complex applications.

Flex layout is to give the container the ability to alter its items' width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes).

Provides a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic.

Some properties for flex layout are set on the container (parent element, known as "flex container") whereas the others are meant to be set on the children (said "flex items").

**GRID LAYOUT**

Grid Layout (aka "Grid"), is a two-dimensional layout system.

Usually we used tables, then floats, positioning and inline-block, but all of these methods were essentially hacks and left out a lot of important functionality (vertical centering, for instance).

Flexbox helped out, but it's intended for simpler one-dimensional layouts, not complex two-dimensional ones (linear layouts).

Grid is one of the most powerful CSS modules ever introduced.

**Task: Differentiate between flexbox and grid**

|  |  |
| --- | --- |
| **FLEX** | **GRID** |
| * They are one dimensional containers. * All of the images occupied width and height which was given to­­ them using CSS and didn’t moved out of their container. * Single item cannot occupy more than one row or column. * we cannot set the number of rows and columns in flex layout. | * 1.They are two dimensional containers. * We had to use object-fit property equal to cover on items so that image could occupy their whole container rather getting out of it and disturb alignment of other images in grid. * single image can span more than one row and column. * we can set the number of rows and columns we want in a grid. |



