**CSS Grid and CSS Flexbox**

**CSS Grid:**

CSS Grid Layout, is a two-dimensional grid-based layout system with rows and columns, making it easier to design web pages without having to use floats and positioning. Like tables, grid layout allow us to align elements into columns and rows.

To get started you have to define a container element as a grid with **display: grid,** set the column and row sizes with grid-template-columns and grid-template-rows, and then place its child elements into the grid with grid-column and grid-row.

**Example:** In this example, we are using the above-explained method.

|  |
| --- |
| <!DOCTYPE html>  <**html** lang="en">  <**head**>      <**style**>          .main{              display: grid;              grid: auto auto / auto auto auto auto;              grid-gap: 10px;              background-color: green;              padding: 10px;          }          .gfg {              background-color: rgb(255, 255, 255);              text-align: center;              padding: 25px 0;              font-size: 30px;          }      </**style**>  </**head**>  <**body**>      <**h2** style="text-align: center;">            Welcome To GeeksForGeeks        </**h2**>      <**div** class="main">          <**div** class="gfg">Home</**div**>          <**div** class="gfg">Read</**div**>          <**div** class="gfg">Write</**div**>          <**div** class="gfg">About Us</**div**>          <**div** class="gfg">Contact Us</**div**>          <**div** class="gfg">Privacy Policy</**div**>      </**div**>  </**body**>  </**html**> |

A green and white rectangular box with black text

Description automatically generated

**CSS Flexbox:**

The CSS Flexbox offers a one-dimensional layout. It is helpful in allocating and aligning the space among items in a container (made of grids). It works with all kinds of display devices and screen sizes.

To get started you have to define a container element as a grid with **display: flex;**

**Example:** In this example, we are using the above-explained method.

|  |
| --- |
| <!DOCTYPE html>  <**html** lang="en">  <**head**>      <**style**>          .main{              display: flex;              grid: auto auto / auto auto auto auto;              grid-gap: 10px;              background-color: green;              padding: 10px;          }          .gfg {              background-color: rgb(255, 255, 255);              text-align: center;              padding: 25px 0;              font-size: 30px;          }      </**style**>  </**head**>  <**body**>      <**h2** style="text-align: center;">            Welcome To GeeksForGeeks        </**h2**>      <**div** class="main">          <**div** class="gfg">Home</**div**>          <**div** class="gfg">Read</**div**>          <**div** class="gfg">Write</**div**>          <**div** class="gfg">About Us</**div**>          <**div** class="gfg">Contact Us</**div**>          <**div** class="gfg">Privacy Policy</**div**>      </**div**>  </**body**>  </**html**> |

A green rectangle with black text

Description automatically generated

**Uniqueness In Grid And Flexbox:**

**One Vs Two Dimensions:**

* Grid is made for a two-dimensional layout while Flexbox is for one. This means Flexbox can work on either row or columns at a time, but Grids can work on both.
* Flexbox gives you more flexibility while working on either element (row or column). HTML markup and CSS will be easy to manage in this type of scenario.
* GRID gives you more flexibility to move around the blocks irrespective of your HTML markup.

**Content-First vs Layout-First:**

* The major Uniqueness between Flexbox and Grids is that the former works on content while the latter is based on the layout.
* The Flexbox layout is best suited to application components and small-scale layouts, while the Grid layout is designed for larger-scale layouts that are not linear in design.

| **Property** | **Grid** | **Flexbox** |
| --- | --- | --- |
| Dimension | Two – Dimensional | One – Dimensional |
| Features | Can flex combination of items through space-occupying Features | Can push content element to extreme alignment |
| Support Type | Layout First | Content First |

**Conclusion**

* CSS Grids helps you create the outer layout of the webpage. You can build complex as well responsive design with this. This is why it is called ‘layout first’.
* Flexbox mostly helps align content & move blocks.
* CSS grids are for 2D layouts. It works with both rows and columns.
* Flexbox works better in one dimension only (either rows OR columns).
* It will be more time-saving and helpful if you use both at the same time.