



Google Developer Student Clubs
PIEAS

Web Development Bootcamp: Intro to CSS Part - 2

CSS



Abu Bakar

GDSC PIEAS

Layouts in CSS?

“CSS layouts define the structure and positioning of elements on a web page.”

It's like arranging the furniture in a room. Layouts help organize content, making it visually appealing and easy to navigate.

Flexbox & Grid?

Flexbox

“Flexbox is a one-dimensional layout method for laying out items in rows or columns.”

It's like a parent managing children in a line. Flexbox makes it easy to create flexible and responsive layouts

Grid

“CSS Grid Layout is a two-dimensional layout system for the web.”

Picture it as a grid of intersecting rows and columns. You can precisely position items anywhere on the grid.

Flex vs Grid

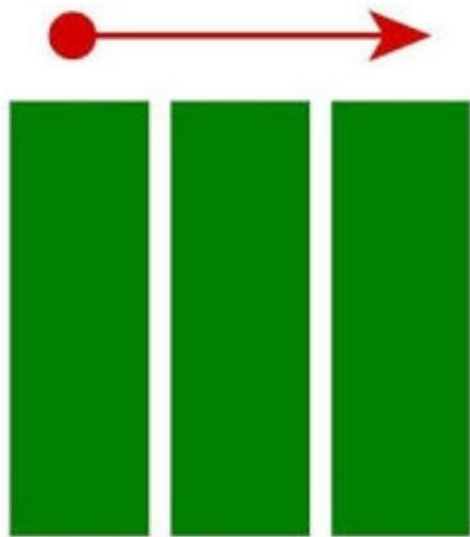
CSS Flexbox	CSS Grid
CSS Flexbox is a one-dimensional layout model.	CSS Grid is a two-dimensional model.
A Flexbox container can either facilitate laying out things in a row, or lay them out in a column.	Grid can facilitate laying out items across and down at once.
Flexbox cannot intentionally overlap elements or items in a layout.	CSS Grid helps you create layouts with overlapping elements.
Flexbox is basically content based and it listens to the content and adjusts to it.	Grid operates more on the layout level and it is container based.

Flex vs Grid

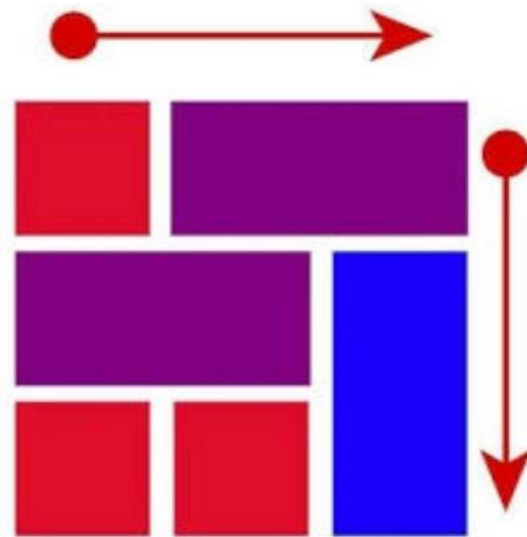


Flexbox on top, Grid on bottom

Flex vs Grid



Flexbox
One Dimensions



CSS Grids
Two Dimensions

Flex

```
.box {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```



Google Developer Student Clubs

PEAS

```
/*1*/  
child: Column(  
  crossAxisAlignment: CrossAxisAlignment.  
  children: [  
    /*2*/  
    Container(  
      padding: const EdgeInsets.  
      child: const Text(  
        'Oeschinen Lake Camp  
        style: TextStyle(  
          fontWeight: FontWe  
      ),  
    ),  
  ],  
)
```


Grid

```
.box {  
  display: grid;  
  grid: auto auto;  
  grid-gap: 10px;  
}
```



Google Developer Student Clubs

PEAS

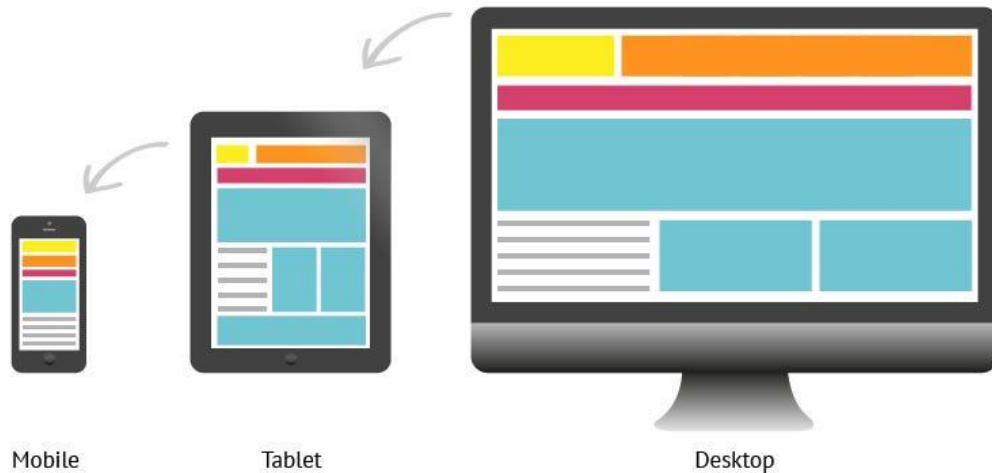
```
/*1*/  
child: Column(  
  crossAxisAlignment: CrossAxisAlignment.  
  children: [  
    /*2*/  
    Container(  
      padding: const EdgeInsets.  
      child: const Text(  
        'Oeschinen Lake Campo  
        style: TextStyle(  
          fontWeight: FontWeig  
      ),  
    ),  
  ],  
)
```

Responsive Designs / Responsiveness?

“Adapting websites for different screen sizes and devices.”

Your website should look good on a smartphone, tablet, or computer. It's like having an outfit that fits perfectly no matter where you go.

Responsive Designs / Responsiveness?



Techniques for Responsiveness

1. Media Query
2. Relative units
3. Viewport
4. Layouts (Flexbox and Grid 😊)
5. Responsive Images

Self Learning

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design

https://www.w3schools.com/css/css_rwd_intro.asp

CSS Frameworks?

“Frameworks provide pre-designed components and grids for rapid development.”

It's like having a toolbox full of pre-built items. Bootstrap is like IKEA furniture, Tailwind is like LEGO bricks.

Self Learning

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>



Bonus – Adding Animations

```
.box {  
  transition: transform 0.5s ease-in-out;  
}
```

```
.box:hover {  
  transform: scale(1.2);  
}
```

https://www.w3schools.com/css/css3_animations.asp