

Lecture # 14

Course

Advance Web designing and Development

Advance CSS

Ms. Hafiza Alia
(alia@theProTec.com)

CSS Flexbox Layout Module

- ▶ Before the Flexbox Layout module, there were four layout modes:
 - ▶ Block, for sections in a webpage
 - ▶ Inline, for text
 - ▶ Table, for two-dimensional table data
 - ▶ Positioned, for explicit position of an element

CSS Flexbox Layout Module

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

How to make a Flexbox model?

Step 1:

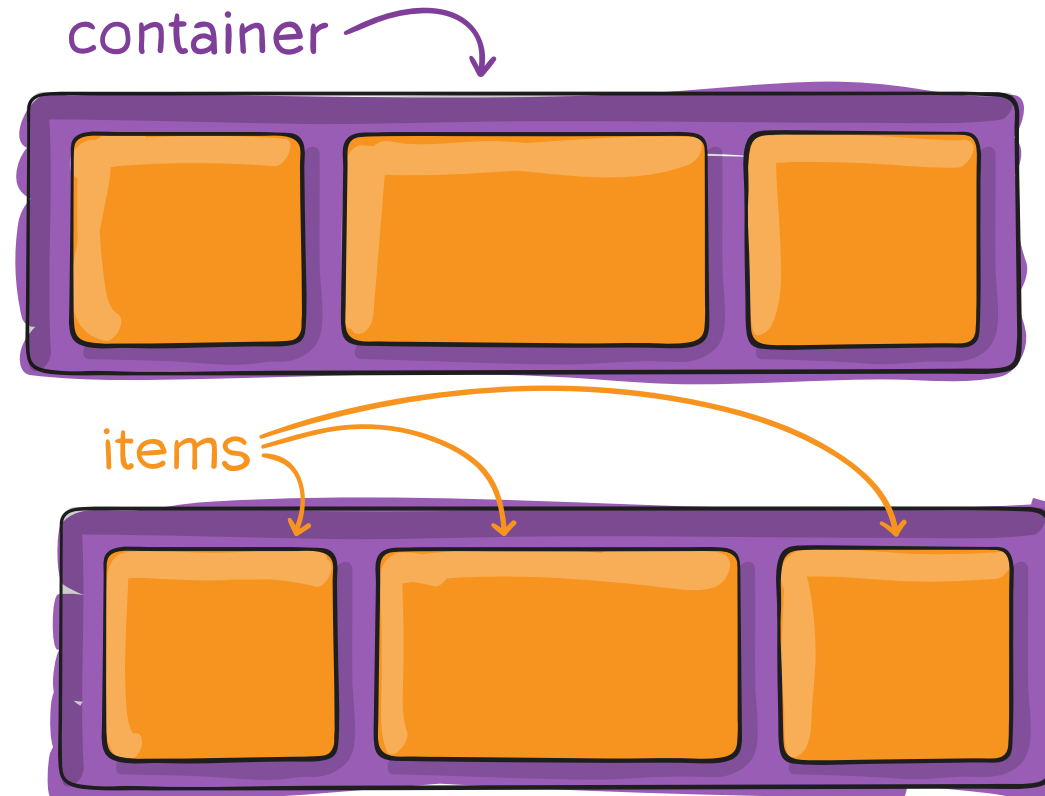
Define a flex container

```
<div class="flex">
```

```
</div>
```

Step 2:

Define display flex
for parent div
which is flex



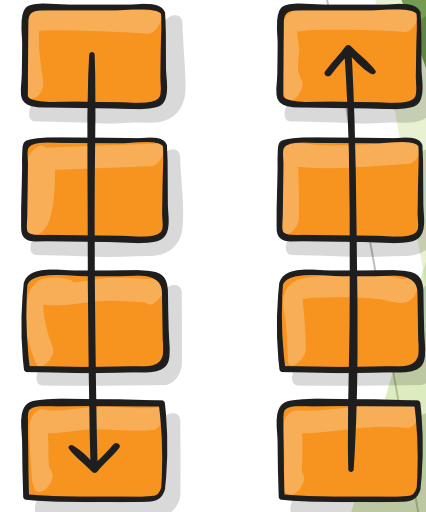
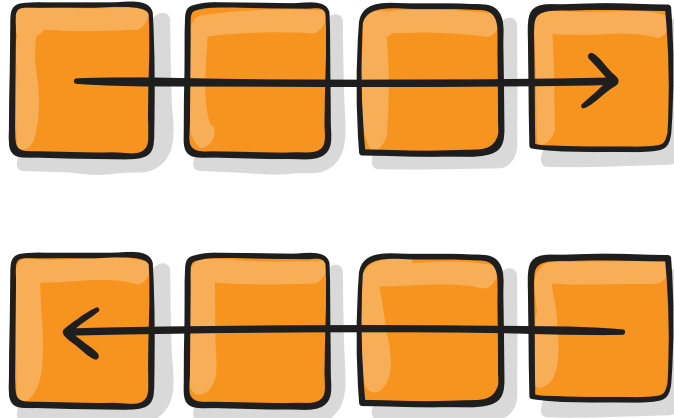
The flex container properties

- ▶ flex-direction
- ▶ flex-wrap
- ▶ justify-content
- ▶ align-items
- ▶ align-content

flex-direction

► Values

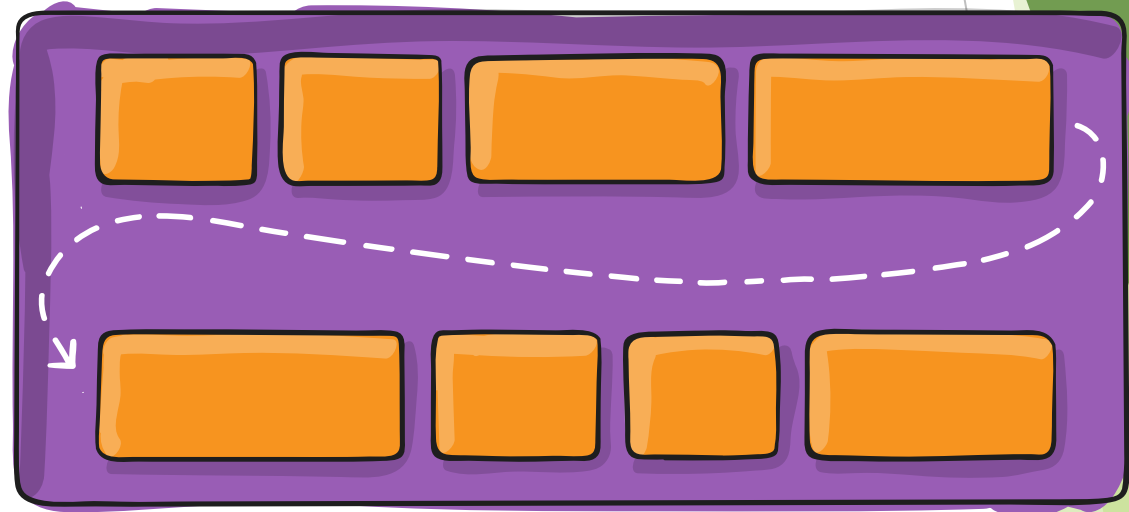
- Column
- column-reverse
- Row
- row-reverse



flex-wrap

► Values

- wrap
- nowrap
- wrap-reverse



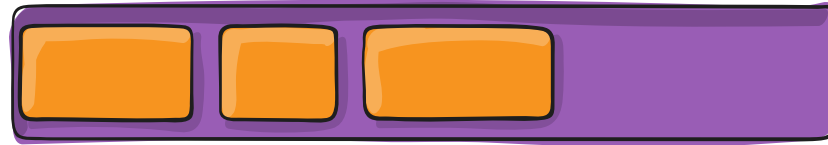
Flex Flow

- ▶ The **flex-flow** property is a shorthand property for:
 - ▶ flex-direction
 - ▶ flex-wrap

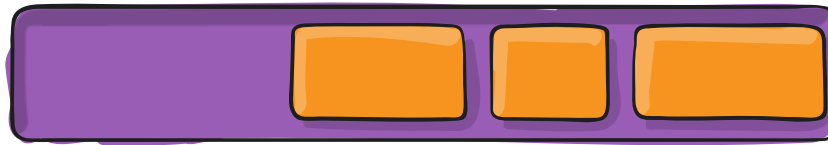
justify-content

- ▶ Align items horizontally
- ▶ Values
 - ▶ center
 - ▶ flex-start
 - ▶ flex-end
 - ▶ space-around
 - ▶ space-between

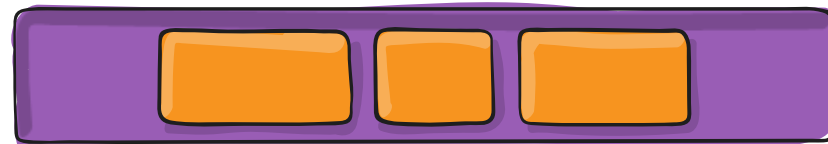
flex-start



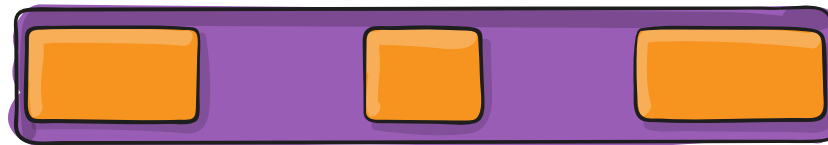
flex-end



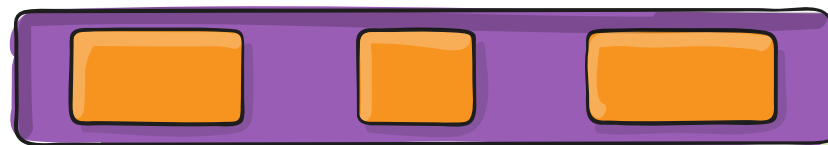
center



space-between



space-around

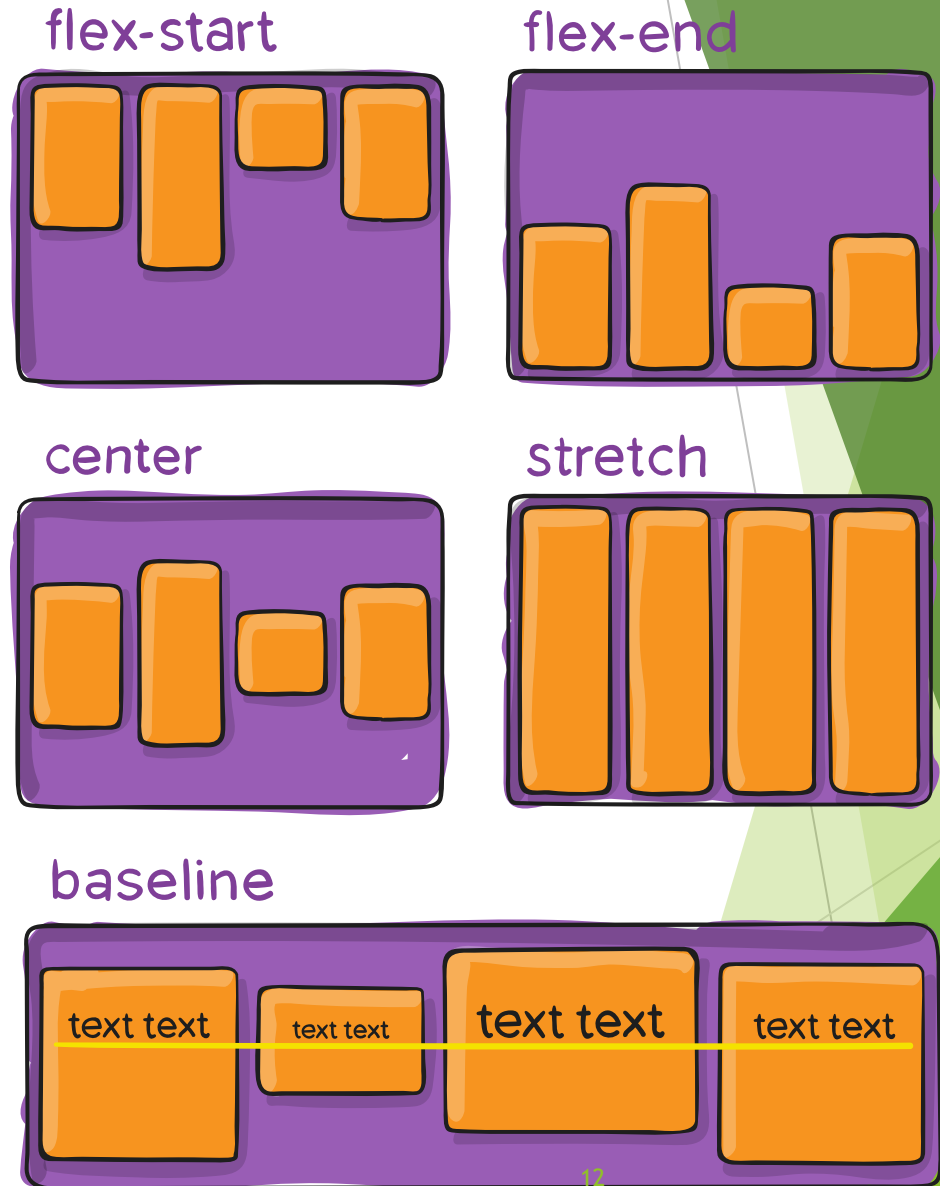


align-items

- ▶ The align-items property is used to align the flex items vertically.

- ▶ Values

- ▶ center
- ▶ flex-start
- ▶ flex-end
- ▶ Stretch (default)

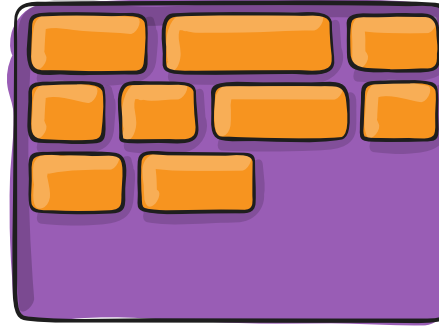


align-content

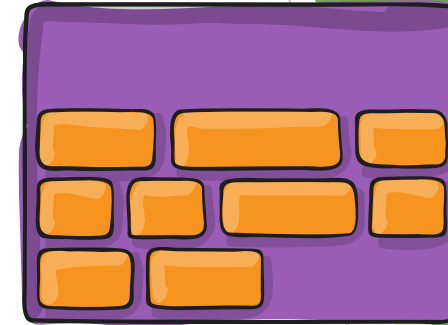
- ▶ Used to align flex lines
- ▶ Values
 - ▶ Space-between
 - ▶ Space-around
 - ▶ Stretch (default)
 - ▶ center

the flex-wrap property set to *wrap*, to better demonstrate the **align-content** property.

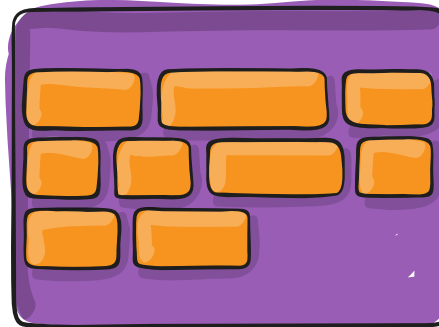
flex-start



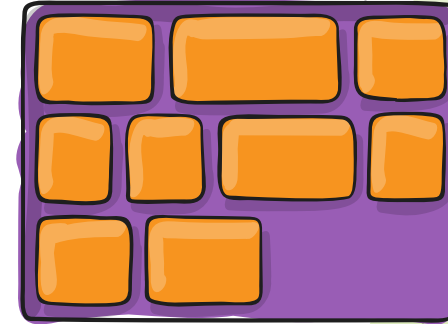
flex-end



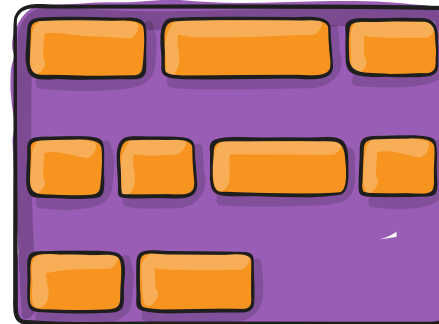
center



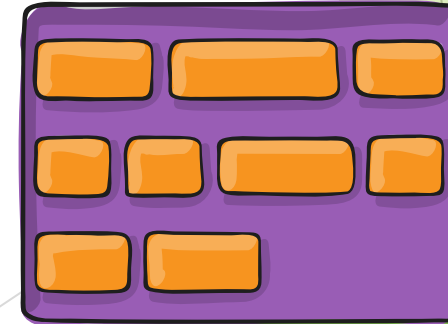
stretch



space-between



space-around

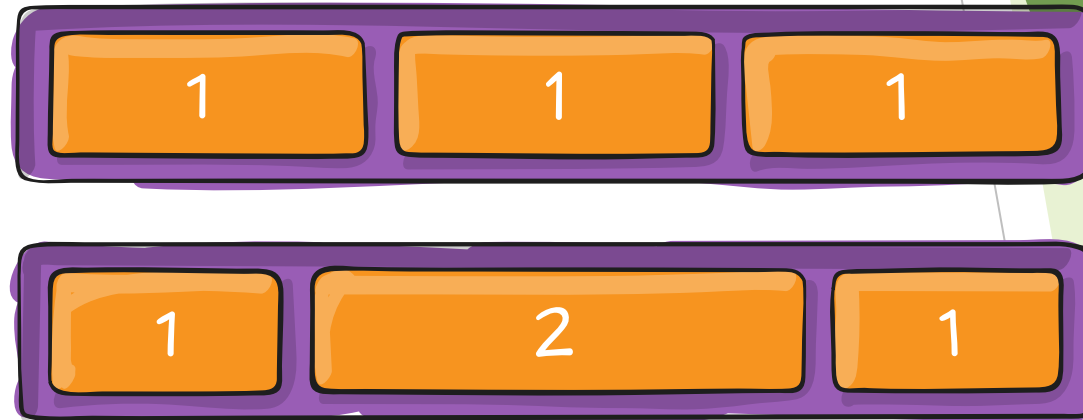


Flex order (Flex Item)

```
<div class="flex-container">  
  <div style="order: 3">1</div>  
  <div style="order: 2">2</div>  
  <div style="order: 4">3</div>  
  <div style="order: 1">4</div>  
</div>
```

Flex grow(Flex Item)

- ▶ how much it grow
- ▶ flex-grow:number default value is 0
- ▶ This defines the ability for a flex item to grow if necessary.
- ▶ If all items have flex-grow set to 1, the remaining space in the container will be distributed equally to all children.



```
<div class="flex-container">  
  <div style="flex-grow: 1">1</div>  
  <div style="flex-grow: 1">2</div>  
  <div style="flex-grow: 8">3</div>  
</div>
```

Flex shrink (Flex Item)

- This defines the ability for a flex item to shrink if necessary.

```
.item {  
  flex-shrink: <number>; /* default 1 */  
}
```

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div style="flex-shrink: 0">3</div>  
  <div>4</div>  
  <div>5</div>  
  <div>6</div>  
  <div>7</div>  
  <div>8</div>  
  <div>9</div>  
  <div>10</div>  
</div>
```


Flex basic

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div style="flex-basis: 200px">3</div>  
  <div>4</div>  
</div>
```

- specifies initial length
- OR defines the default size of an element before the remaining space is distributed.

```
.item {  
  flex-basis: <length> | auto; /* default auto */  
}
```

If set to 0, the extra space around content isn't factored in.

Flex:

- ▶ shorthand property for the flex-grow, flex-shrink, and flex-basis

Note

- ▶ All the div inside a flex container are flexible and are flex items
- ▶ The flex item properties are:
 - ▶ □ Order (set the order of flex order:number)
 - ▶ □ flex-grow (how much it grow flex-grow:number default value is 0)
 - ▶ □ flex-shrink ()
 - ▶ □ flex-basis (specifies initial length)
 - ▶ □ Flex (shorthand property for the flex-grow, flex-shrink, and flex-basis)
 - ▶ □ align-self (override the default alignment)

Flex Responsive

```
/* Responsive layout - makes a one column-layout instead of  
two-column layout */
```

```
@media (max-width: 800px) {  
  .flex-container {  
    flex-direction: column;  
  }  
}
```

CSS Media Queries

CSs Media Queries

- ▶ Used to make web page responsive
- ▶ What is Responsive Web designs ?
 - ▶ Responsive web design makes your web page look good on all devices.
 - ▶ Responsive web design uses only HTML and CSS.
 - ▶ Responsive web design is not a program or a JavaScript.

What are media queries

Media queries let you customize styles based on the **characteristics** of the **user's device** or **display**

What are media queries

This media query starts with the @media rule, and then specifies a media type followed by the characteristic we want to match against, called the media feature

@media screen and (max-width: 600px)

What are media queries

```
@media screen and (max-width: 600px)
{
  Body{}
  .class{}
}
```

This particular media feature, max-width:600px, tells the browser that the styles for this media query, which are contained within a set of curly brackets for the media query as a whole, should apply only up to a maximum width of 600 pixels.

What are media queries

@media screen and (max-width: 600px)

{

Body{ background-color:yellow}

.class{.....}

}

If the viewport width exceeds 600 pixels, the browser will ignore the styles inside this media query

What are media queries

@media screen and (max-width: 600px)

{

Body{ background-color:yellow}

.class{.....}

}

media query can be dropped right into your main style sheet.

Use Separate stylesheet

```
<link rel="stylesheet" media= “screen and  
(expressions)” href=“screen.css>
```

```
expression =“(max-width:780px)”
```

Now you know
media query syntax

but media queries don't actually
do anything

it's the CSS INSIDE
that changes the appearance

Let's see some examples

How to Change the background color when screen size is less than 600px;

Change the flex-direction from row to column when screen is less than and equals to 600px;

Hide an Item when screen is less than 780px

Hide an Item when screen is greater than or equals to 780px

Variable font size

Write media query for following:

If screen size is more than 600px wide or wider, set the font-size to 80px

If screen size is 600px wide, or less, set the font-size to 30px

Example

Use media queries to set the background-color to lavender if the viewport is 800 pixels wide or wider,

to lightgreen if the viewport is between 400 and 799 pixels wide.

If the viewport is smaller than 400 pixels, the background-color is lightblue:

Typical Device Breakpoints

There are tons of screens and devices with different heights and widths, so it is hard to create an exact breakpoint for each device. To keep things simple you could target five groups:

```
/* Extra small devices (phones, 600px and down) */  
@media only screen and (max-width: 600px) {  
    .example {background: red;}  
}
```

```
/* Small devices (portrait tablets and large phones,  
600px and up) */  
@media only screen and (min-width: 600px) {  
    .example {background: green;}  
}
```



```
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {  
    .example {background: blue;}  
}  
  
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {  
    .example {background: orange;}  
}  
  
/* Extra large devices (large laptops and desktops, 1200px  
and up) */  
@media only screen and (min-width: 1200px) {  
    .example {background: pink;}  
}
```

Responsive Web Design - Frameworks

- ▶ Bootstrap. Being one of the most popular front-end **development frameworks**, Bootstrap is now available with the latest Bootstrap 4 version.
- ▶ Foundation. Foundation is also counted among exceptional front-end **frameworks**. ...
- ▶ Pure. ...
- ▶ Skeleton. ...
- ▶ Montage. ...
- ▶ Siimple. ...
- ▶ Gumby. ...
- ▶ Semantic UI.

End Of Lecture 14

Next Topic

Java Script