[Lesson 11]

Roi Yehoshua 2018

[DAN.IT EDUCATION]

What we learnt last time?

- How to create a table with header, footer and content
- How to combine table cells horizontally and vertically
- New HTML5 semantic tags
- Background color with gradient
- Difference between radial and linear gradient;
- How to add shadow for a text
- How to add colorful shadow for block elements
- calc() function

[Our targets for today]

- Flexbox:
  - How to create multi-column design without necessity to clear the flow
  - How to create columns of same height
  - How to center elements in the parent block horizontally and vertically
  - How to make responsive and fixed width columns

[DAN.IT] EDUCATION

[CSS FlexBox]

- The Flexible Box Layout Module makes it easier to design flexible responsive layout structure without using float or positioning
- To start using the Flexbox model, you need to first define a **flex container**
- The flex container becomes flexible by setting its display property to *flex*:

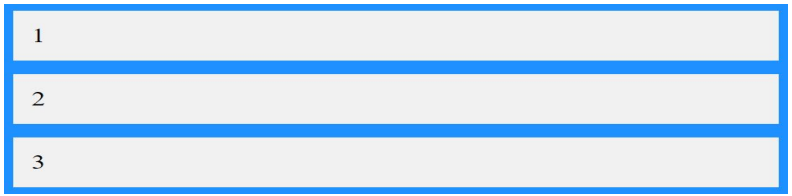
```
<style>
    .flex-container {
        display: flex;
        background-color: dodgerblue;
    }
</style>

<div class="flex-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
</div>
```

[DAN.IT] EDUCATION

The flex-direction Property

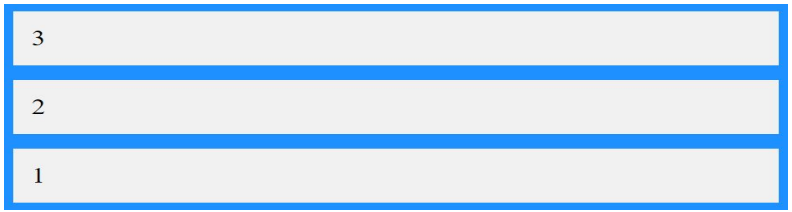
- flex-direction defines in which direction the container wants to stack the flex items
- For example, the *column* value stacks the flex items vertically (from top to bottom):

```
.flex-container {
   display: flex;
   flex-direction: column;
}
```



- The *column-reverse* value stacks the flex items vertically (but from bottom to top):

```
.flex-container {
   display: flex;
   flex-direction: column-reverse;
}
```



[DAN.IT] EDUCATION

The flex-direction Property

- The *row* value stacks the flex items horizontally (from left to right):

```
.flex-container {
   display: flex;
   flex-direction: row;
}
```

- The *row-reverse* value stacks the flex items horizontally (but from right to left):

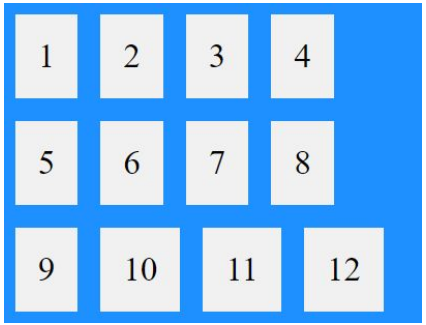
```
.flex-container {
   display: flex;
   flex-direction: row-reverse;
}
```

[DAN.IT] EDUCATION

The flex-wrap Property

- The flex-wrap property specifies whether the flex items should wrap or not

```
.flex-container {
    display: flex;
    flex-wrap: wrap;
}
```



- The **flex-flow** property is a shorthand property for setting both the flex-direction and flex-wrap properties:

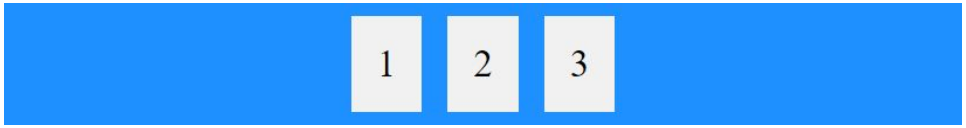
```
.flex-container {
    display: flex;
    flex-flow: row wrap;
}
```

[DAN.IT] EDUCATION

The justify-content Property

- The justify-content property is used to align the flex items:

```
.flex-container {
    display: flex;
    justify-content: center;
}
```



- The *flex-end* value aligns the flex items at the end of the container:

```
.flex-container {
    display: flex;
    justify-content: flex-end;
}
```

- The *space-around* value displays the flex items with space before, between, and after the lines:

```
.flex-container {
    display: flex;
    justify-content: space-around;
}
```

[DAN.IT] EDUCATION

The align-items Property

- The align-items property is used to align the flex items vertically:

```
.flex-container {
    display: flex;
    height: 200px;
    align-items: center;
}
```

- The *flex-end* value aligns the flex items at the bottom of the container:

```
.flex-container {
    display: flex;
    height: 200px;
    align-items: flex-end;
}
```

[Perfect Centering]

→ To achieve perfect centering set both the justify-content and align-items properties to *center,* and the flex item will be perfectly centered:

```
<style>
    .flex-container {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 300px;
        background-color: DodgerBlue;
    }
</style>

<div class="flex-container">
    <div></div>
</div>
```

[DAN.IT EDUCATION]

[Flex Items]

- The direct child elements of a flex container automatically becomes flexible (flex) items
- The flex item properties are:
  - order
  - flex-grow
  - flex-shrink
  - flex-basis
  - flex
  - align-self

[DAN.IT] EDUCATION

[Flex Items]

→ For example, the *order* property specifies the order of the flex items:

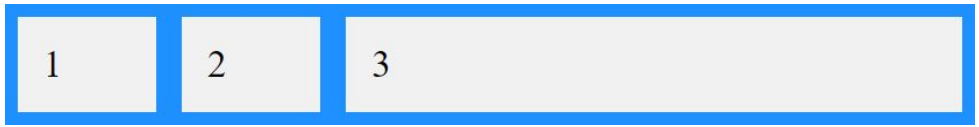
```
<style>
    .flex-container {
        display: flex;
    }
</style>

<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>
```

[DAN.IT] EDUCATION

[Flex Items]

- The *flex-grow* property specifies how much a flex item will grow relative to the rest of the flex items
- For example, we'll make the third flex item grow eight times faster than the other flex items:

```
<style>
    .flex-container {
        display: flex;
    }
</style>

<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>
```

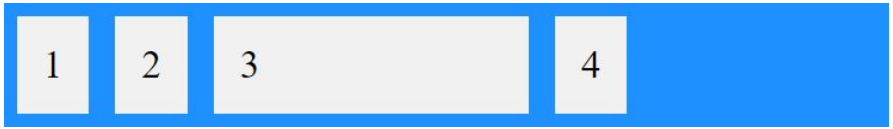


[DAN.IT] EDUCATION

Flex Items

→ The *flex-basis* property specifies the initial length of a flex item:

```
<style>
    .flex-container {
        display: flex;
    }
</style>

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

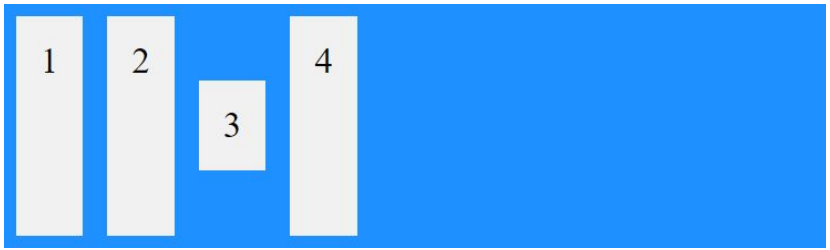


[DAN.IT] EDUCATION

[Flex Items]

- The *align-self* property specifies the alignment for the selected item inside the flexible container
- It overrides the default alignment set by the container's align-items property.

```
<style>
    .flex-container {
        display: flex;
        height: 200px;
    }
</style>

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



[DAN.IT EDUCATION]

[Flex]

- flex: - short format to write flex-grow, flex-shrink and flex-basis properties
- flex: 1; - is the same as 1 1 0
- flex-grow: defaults to 1 when omitted
- flex-shrink: defaults to 1 when omitted
- flex-basis: defaults to 0 when omitted

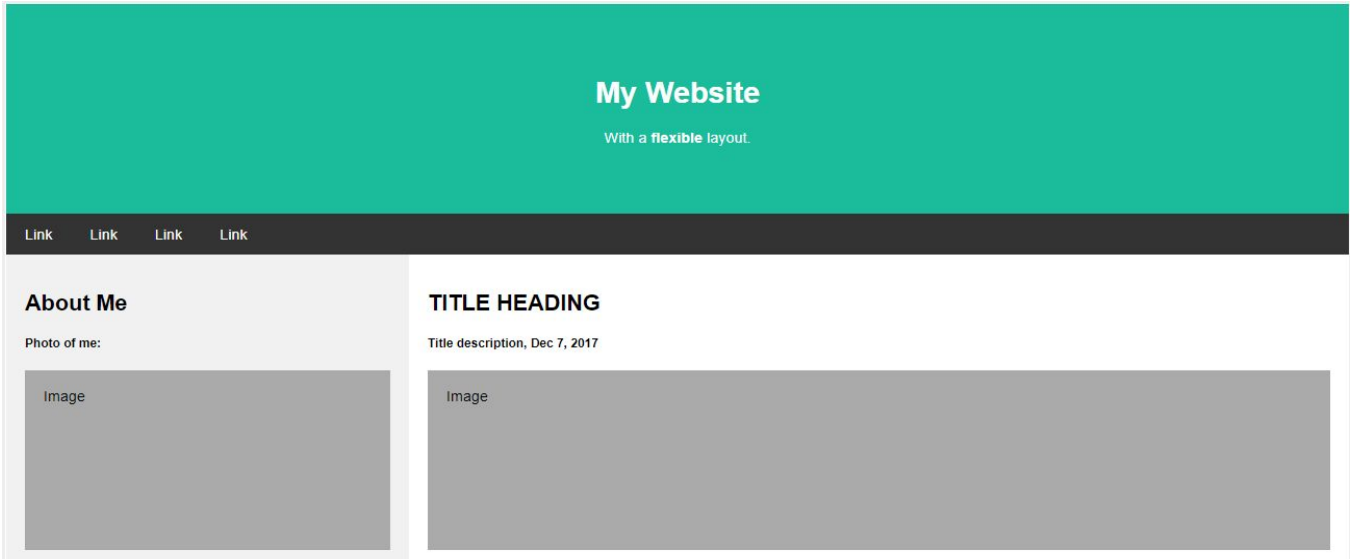
[DAN.IT] EDUCATION

CSS FlexBox Properties Summary

Prefix	Browsers
display	Specifies the type of box used for an HTML element
flex-direction	Specifies the direction of the flexible items inside a flex container
justify-content	Horizontally aligns the flex items when the items do not use all available space on the main-axis
align-items	Vertically aligns the flex items when the items do not use all available space on the cross-axis
flex-wrap	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
align-content	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
flex-flow	A shorthand property for flex-direction and flex-wrap
order	Specifies the order of a flexible item relative to the rest of the flex items inside the same container
align-self	Used on flex items. Overrides the container's align-items property
flex	A shorthand property for the flex-grow, flex-shrink, and the flex-basis properties

[Exercise]

→ Use flexbox to create a responsive website, containing a flexible navigation bar and flexible content:



[DAN.IT EDUCATION]

Control questions

1. What is flexbox layout?
2. What does flexbox layout makes easier?
3. Name flexbox styles for aligning items inside it
4. Name flexbox styles that can be used in responsive designs
5. How to place flex-elements in two rows if their total width is greater than their parent width?
6. How does “justify-content: space-around” work?
7. What is the difference between “align-items” and “align-content”?
8. How to place flex-element in the end of row?
9. How can you create a flex-element that could increase its width but not decrease?
10. How to vertically align only one flex-element out of many?

[DAN.IT] EDUCATION