[Lesson 12]

Roi Yehoshua 2018

[DAN.IT EDUCATION]

What we learnt last time?

- 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

[Our targets for today]

- CSS Grid Layout

[DAN.IT EDUCATION]

[CSS Grids]

→ CSS Grid offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning

```
<style>
    .grid-container {
        display: grid;
        grid-template-columns: auto auto auto;
        background-color: DodgerBlue;
        padding: 10px;
    }

    .grid-item {
        background-color: #f1f1f1;
        border: 1px solid rgba(0, 0, 0, 0.8);
        padding: 20px 0;
        font-size: 30px;
        text-align: center;
    }
 </style>
```

```
<div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
    <div class="grid-item">7</div>
    <div class="grid-item">8</div>
    <div class="grid-item">9</div>
</div>
```

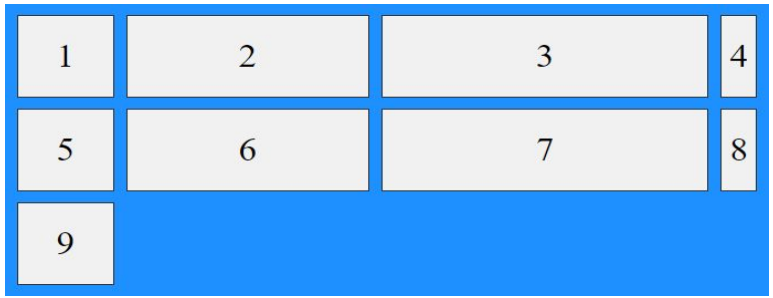
1	2	3
4	5	6
7	8	9

[DAN.IT] EDUCATION

[Grid Container]

- To make an HTML element behave as a grid container, you have to set the display property to *grid* or *inline-grid*
- Grid containers consist of grid items, placed inside columns and rows
- The **grid-template-columns** property defines the number of columns in your grid layout, and it can define the width of each column
  - If you want your grid layout to contain 4 columns, specify the width of the 4 columns, or "auto" if all columns should have the same width

```
.grid-container {
    display: grid;
    grid-template-columns: 80px 200px auto 30px;
    grid-gap: 10px;
}
```

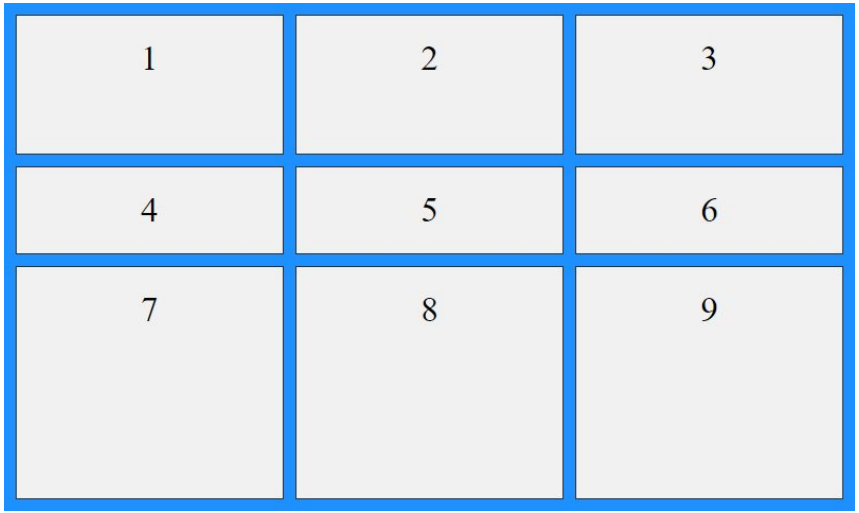


[DAN.IT EDUCATION]

The grid-template-rows Property

→ The grid-template-rows property defines the height of each row:

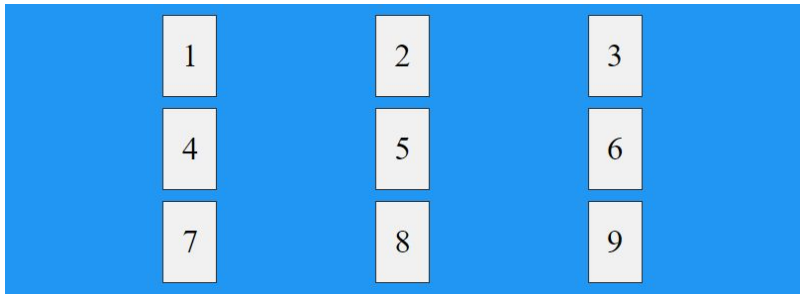
```
.grid-container {
    display: grid;
    grid-template-columns: auto auto auto;
    grid-template-rows: 120px auto 200px;
    grid-gap: 10px;
}
```



The justify-content Property

→ The justify-content property is used to align the whole grid inside the container:

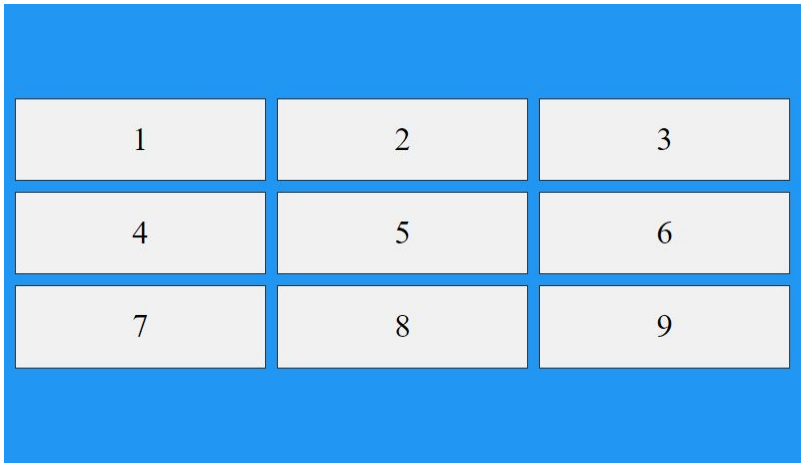
```
.grid-container {
    display: grid;
    justify-content: space-evenly;
    grid-template-columns: 50px 50px 50px;
    /*Make the grid smaller than the
    container*/  grid-gap: 10px;
}
```



The align-content Property

- The align-content property is used to *vertically* align the whole grid inside the container

```
.grid-container {
    display: grid;
    height: 400px;
    align-content: center;
    grid-template-columns: auto auto auto;
    grid-gap: 10px;
}
```

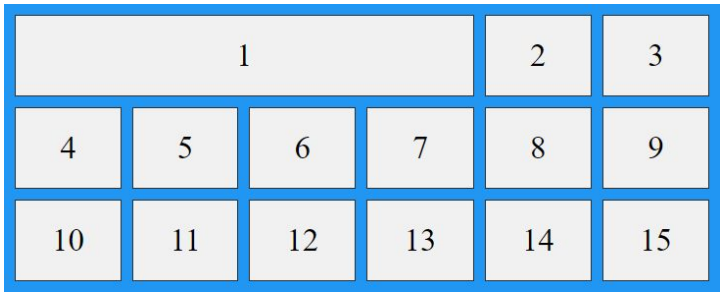


[Grid Items]

- A grid *container* contains grid *items*
- The **grid-column** property defines on which column(s) to place an item
- You define where the item will start, and where the item will end
- For example, to make "item1" start on column 1 and end on column 5:

```
.grid-container {
    display: grid;
    grid-template-columns: auto auto auto auto auto auto;
    grid-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
}

.item1 {
    grid-column: 1 / 5;
}
```



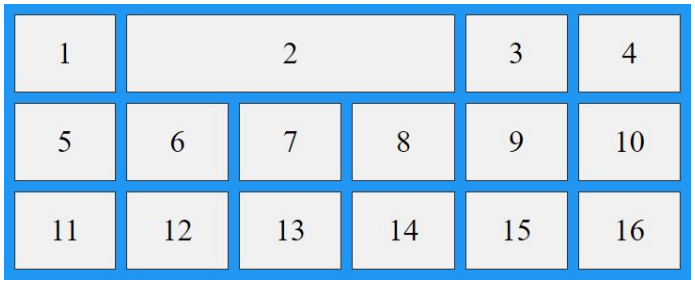
[DAN.IT] EDUCATION

[Grid Items]

→ To make "item2" start on column 2 and span 3 columns:

```
.grid-container {
    display: grid;
    grid-template-columns: auto auto auto auto auto auto;
    grid-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
}

.item2 {
    grid-column: 2 / span 3;
}
```



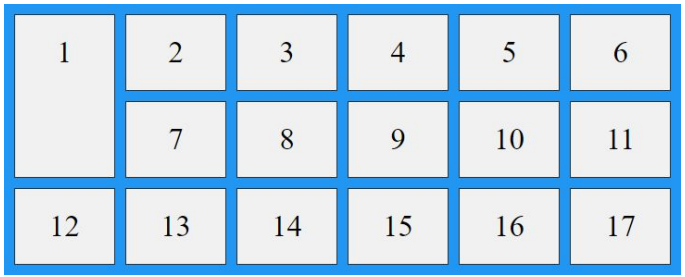
[DAN.IT] EDUCATION

The grid-row Property

- The grid-row property defines on which row to place an item
- To place an item, you can refer to *line numbers*, or use the keyword "span" to define how many rows the item will span
- For example, to make "item1" start on row 1 and span 2 rows:

```
.grid-container {
    display: grid;
    grid-template-columns: auto auto auto auto auto auto;
    grid-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
}

.item1 {
    grid-row: 1 / span 2;
}
```



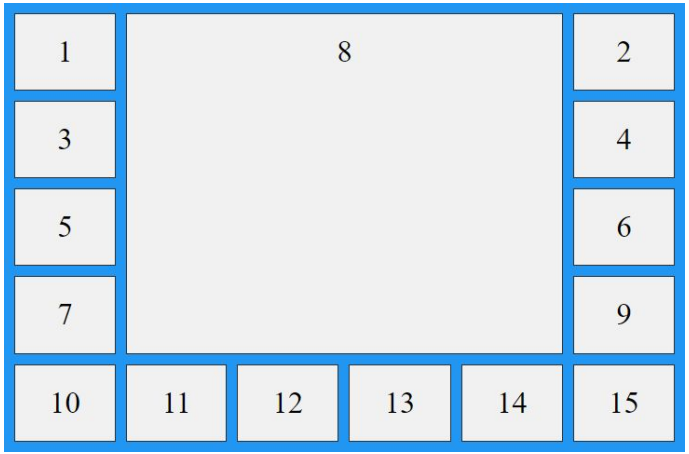
[DAN.IT] EDUCATION

The grid-area Property

- The grid-area property can be used as a shorthand property for the *grid-row-start, grid-column-start, grid-row-end* and the *grid-column-end* properties
- For example, to make "item8" start on row-line 1 and column-line 2, and end on row-line 5 and column line 6:

```
.grid-container {
    display: grid;
    grid-template-columns: auto auto auto auto auto auto;
    grid-gap: 10px;
    background-color: #2196F3;
    padding: 10px;
}

.item8 {
    grid-area: 1 / 2 / 5 / 6;
}
```



Control questions

1. What is grid layout?
2. What does grid layout makes easier?
3. What is the difference in the flexbox and grid approaches?
4. When will you use flexbox and when grid layouts?