

CSS
IS
AWESOME

Module 3 Day 2

Grid and Flexbox

What makes an application?

- Program Data

- ✓ Variables & .NET Data Types
- ✓ Arrays
- ✓ More Collections (list, dictionary, stack, queue)
- ✓ Classes and objects (OOP)

- Program Logic

- ✓ Statements and expressions
- ✓ Conditional logic (if)
- ✓ Repeating logic (for, foreach, do, while)
- ✓ Methods (functions / procedures)
- ✓ Classes and objects (OOP)
- ❑ Frameworks (MVC)

- Input / Output

- User

- ✓ Console read / write

- ❖ [HTML / CSS](#)

- ❑ Front-end frameworks (HTML / CSS / JavaScript)

- Storage

- ✓ File I/O

- ✓ Relational database

- ❑ APIs

CSS Variables

- Used to apply styles consistently and make change easier

```
# style.css ▶ body
1  :root {
2      --top-row-height: 120px;
3      --gutter: 20px;
4      --highlight-color: #00adee;
5      --bg-color: rgba(192, 192, 192, 0.1);
6  }
7
8  body {
9      background-color: var(--bg-color);
10     color: var(--highlight-color, darkblue);
11 }
```

Grid

- Often used to divide a page into sections

- Setup a *Grid Container*

```
display: grid;  
grid-template-columns: 100px 1fr 1fr 1fr  
grid-template-areas: ". name1 name2 name3"  
                    ". name4 name4 name4"  
  
grid-gap: 10px; /* or grid-column-gap / grid-row-gap */  
grid-template-rows  
justify-items, align-items: horizontal/vertical alignment of all items
```

- Assign *Grid Items* to a grid area

- Grid Items are direct children of the grid container in the HTML

```
grid-area: name1;  
grid-column-start/end, grid-row-start/end: <linenumber>  
justify-self / align-self: horizontal/vertical alignment
```

Grid Examples and Docs

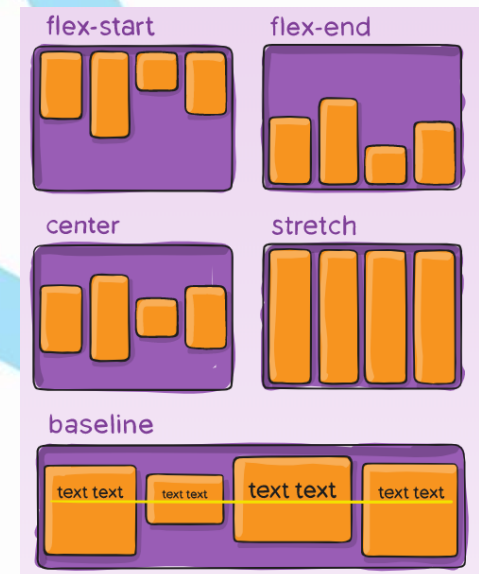
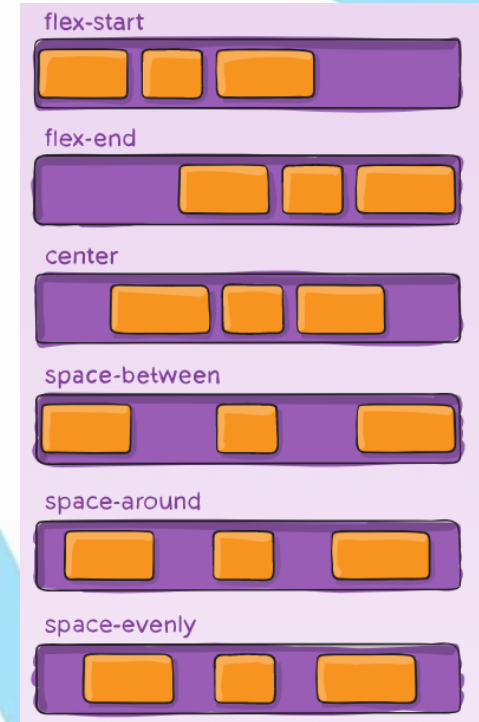
- FireFox Grid Tools
- <https://www.mozilla.org/en-US/developer/css-grid/>
- <https://gridbyexample.com/>
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS Grid Layout/Basic Concepts of Grid Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)

Flexbox

- Usually used to display a series of similar items, such as products, images or cards
- Has a direction: row or column (horizontal / vertical “main” axis)
- Define a flex-container
- The container’s direct children are the flex items

Flex Container

- Display: flex;
- Flex-direction: row, column, row-reverse, column-reverse
- Flex-wrap: wrap, nowrap
- Justify-content: alignment along “main” axis
- Align-items: alignment along the “cross” axis



Flex Items

- Order: n ; sequence of items in the container
- Flex-basis: default size
- Flex-grow, flex-shrink
- Align-self: Overrides parent align-items for this one item

Flex Examples and Docs

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS Flexible Box Layout/Basic Concepts of Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)