

Cascading Style Sheets - CSS

UFCFES-30-1

What are we covering in this lecture?

What is CSS?

Why we need it?

How to use it?

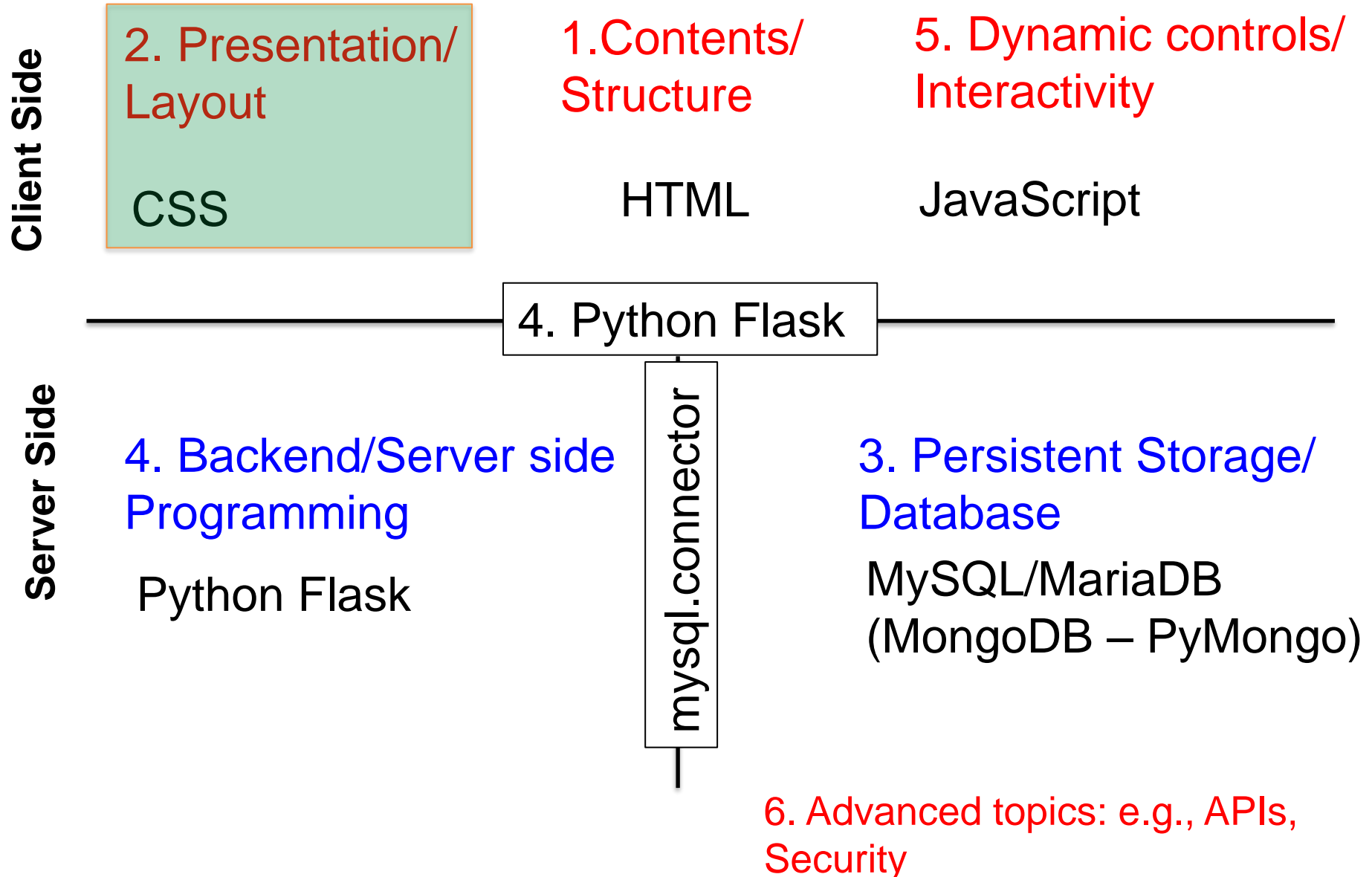
Div tag

CSS Selectors

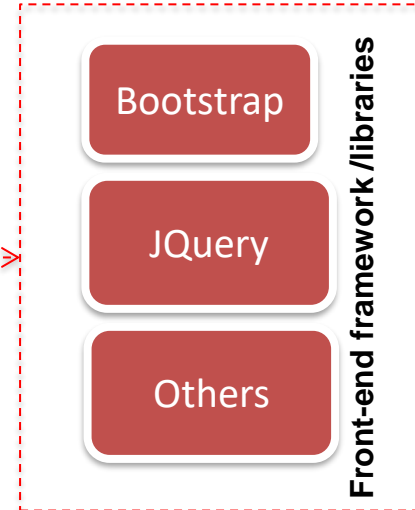
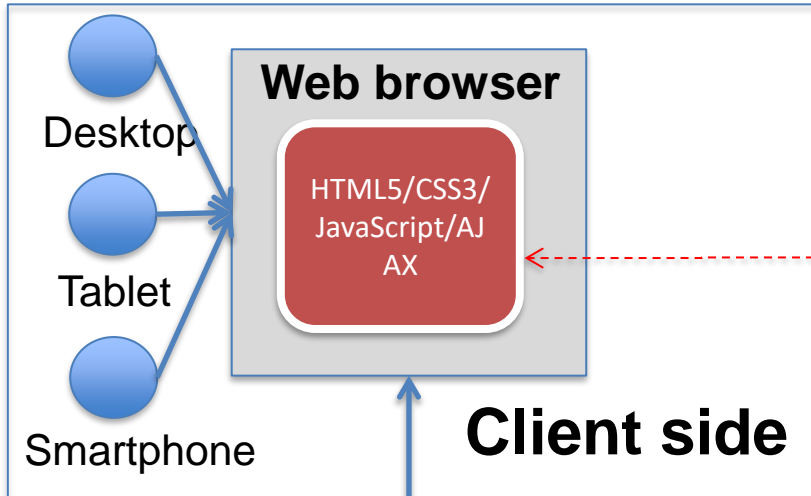
- Element
- Classes
- IDs

Layout techniques

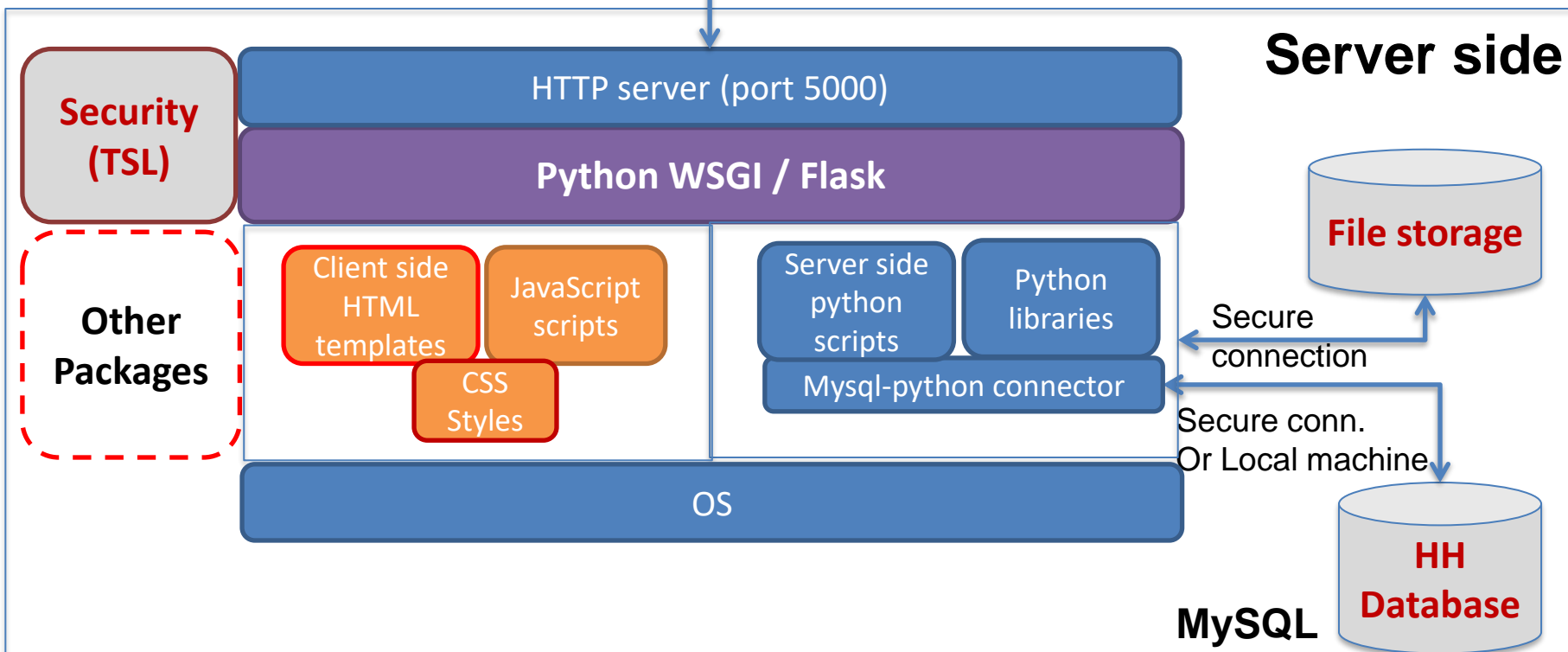
Overall Learning Map



High Level View



Network
connection



Sources

- **Online**
 - **W3Schools:** <http://www.w3schools.com/css/default.asp> or http://www.w3schools.com/html/html_css.asp
 - Another Useful reference for CSS:
 - <http://www.htmlhelp.com/reference/css/style-html.html>
- Books
 - **HTML5 & CSS3 for dummies by David Karlins, 2013 [UWE library – online]**
 - Beginning HTML5 and CSS3 by Richard C., Oli S., Christopher M., Divya M., Springer Verlag, 2012 [UWE library – online]
 - **[Tip: you should get into habit of using library]**

CSS

- What is CSS?
 - Solved **formatting problems** in HTML 3.2 – e.g. font tag
 - **Separation of concern** – **contents** and **formatting**
- Why/When is it needed?
 - Easy **formatting for multiple web pages**, **Time saving**, **Consistent look & feel** across multiple webpages
 - **Example:** Imagine if you have to change font type or style of paragraphs or apply consistent look and feel of images across website
- How to use/apply it?
 - We'll learn in later slides
 - CSS **classes, ids, inline, internal external styles, style properties, div**

CSS

- It is not programming like Python
- Formatting or presentation e.g. layout, colours, fonts, etc.
- Let's you build responsive web apps
- You'll learn webpage design skills
- **Requires a lot of practice**

An Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
  <h1>My First CSS Example</h1>
```

```
  <p>We are trying to learn CSS in Web programming.</p>
```

```
</body>
```

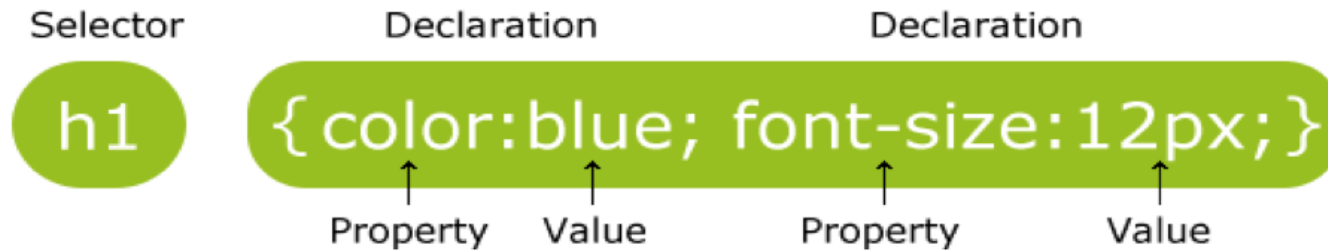
```
</html>
```

My First CSS Example

We are trying to learn CSS in Web programming.

This is HTML document. No styling is applied to this page

Simple CSS



- **Selector** points to HTML element you want to style
- The definition of the **style** is via '**rules**' such as **color: red;**
- Rules are always declared using '**property: value**' pairs like the example above
- Rules can be laid out how you wish (i.e. on one line, multiple lines etc)
- but correct syntax e.g., the **colons** and **semicolons** are required – if you omit them the style will not work

Image source:

http://www.w3schools.com/css/css_syntax.asp

Types of CSS Selectors

- Element Selector
- ID Selector
- Class Selector

CSS Element selector

- **Element selector** – element tag
 - Can be used with element tag and will be applied on all same elements
 - `p { text-align: center; color: red; }`

An Example

```
< ! DOCTYPE   html >
< html >
< head >
    < style >
        body {
            background-color: #00ff00 ;
        }

        h1 {
            color : white ;
            text-align : center ;
        }

        p {
            font-family : "Times New Roman";
            font-size : 20 px ;
        }
    < / style >
< / head >
< body >
    < h1 > My   First   CSS   Example < / h1 >
    < p > We   are   trying   to   learn   CSS   in   Web   programming . < / p >
< / body >
< / html >
```

An Example - Output

My First CSS Example

We are trying to learn CSS in Web programming.

CSS Selectors –ID Selector

- **Id selector** – id attribute of element tag
 - **Id** specifies a single element defined with #
 - Id attribute is needed; Ids are unique; Each element can have one ID; Each page can have only one element with that ID;
 - Can be applied on one specific element with same Id
 - **#paraid1 {text-align: center; color: red; }**
 - In HTML body - **<p id="paraid1">Hello World!</p>**

CSS id Selector

The id selector uses the id attribute of the HTML element, and is defined with a "#".

The style rule below will be applied to the element with `id="para1"`:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #para1
      {
        text-align:center;
        color:red;
      }
    </style>
  </head>
  <body>
    <p id="para1">Hello World!</p>
    <p>This paragraph is not affected by the style.</p>
  </body>
</html>
```

CSS - Class

- **Class selector** – class attribute of element tag
 - Class attribute is required; It is defined with a "."
 - Can be applied on multiple elements with same class in a web page; This allows you to set a particular style for many HTML elements with the same class.
 - You can use multiple classes on the same element.
 - `.center { text-align: center; color: green; }`
 - In HTML body –
 - `<h1 class="center">Green and center-aligned heading</h1>`
 - `<p class="center">Green and center-aligned paragraph.</p>`
 - Element specific class selector
 - `p.center { text-align: center; color: red; }`
 - In HTML body –
 - `<h1 class="center">No Effect on This Heading</h1>`
 - `<p class="center">Red and center-aligned paragraph.</p>`
- **Grouping selectors** - `h1, h2, p { text-align: center; color: red; }`

CSS class Selector

In the example below, all HTML elements with `class="center"` will be center-aligned:

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .center
      {
        text-align:center;
      }
    </style>
  </head>
  <body>
    <h1 class="center">Center-aligned heading</h1>
    <p class="center">Center-aligned paragraph.</p>
  </body>
</html>
```

CSS Usage

- There are three ways of inserting a style sheet:
- **READ:** http://www.w3schools.com/css/css_howto.asp
 - External style sheet
 - Internal style sheet
 - Inline style

CSS Usage - 1

- **External:** each page must link to the style sheet using the `<link>`

- `<head>`

```
<link rel="stylesheet" type="text/css"
href="mystyle.css"/>
</head>
```

- Stylesheet e.g., `mystyle.css`

```
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

CSS Usage – 2 and 3

- **Internal:** used when a single document has a unique style
 - It is defined in the **head** section of an HTML page
 - **<head>**

```

<style>
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
</style>
</head>

```
- **Inline:** mixes content with presentation “**tight coupling**”
 - **<p style="color:sienna;margin-left:20px">This is a paragraph.</style>**

CSS Usage?

Which usage style is more appropriate and why?

CSS with HTML: Layout Techniques: Table, Div, Flow, Flex...

- Different techniques are used for multi-column layouts - http://www.w3schools.com/html/html_layout.asp
- **HTML tables:** legacy systems used to structure elements in tables but now **do not use it for presentation/layout**
- **DIV tag:** The `<div>` tag is placed around blocks of HTML elements and can be used to apply a style to all the tags in that block of text

```
<div class="s1">
  <h1> ... </h1>
  <p> ... </p>
</div>
```

- This allows you to chop up your HTML document into individually styled sections
- HTML5 places considerable importance on the use of `<div>` to break up content – try to use `<div>` where ever there is a change of style. Examples at: http://www.w3schools.com/html/html_layout.asp

Layout Techniques - Float

- **CSS float property:** easy to learn and use but less flexible - http://www.w3schools.com/css/css_float.asp
 - **float** – is used for **positioning and formatting content** e.g. let an image or other element float left to the text in a container. Elements after a floating element will flow around it. E.g.
 - `img { float: right; margin-left: 10px; }` float is useful to wrap text around images
 - **clear** – is used to control the behaviour of floating element. It specifies on which side of an element floating elements are not allowed to float.
 - `img { clear : left; }` Means no floating elements allowed on left side
 - `p { clear : both; }` Means no floating elements allowed on both sides of <p>
 - clear should be used after float has been used (i.e. float left and then clear left)
 - See more about **clearfix** here: https://www.w3schools.com/css/css_float_clear.asp
 - You should also look at position property - https://www.w3schools.com/css/css_positioning.asp

Float example

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  img {
6    float: right;
7  }
8  </style>
9  </head>
10 <body>
11 <h2>Float Right Example</h2>
12 <p>This is a paragraph..</p>
13 <p>
14   The Faculty of Environment and Technology at UWE, is situated in the
15   beautiful city of Bristol. We are a highly successful and very popular
16   Faculty of over 6000 students and 450 staff members.
17   The Faculty of Business and Law comprises Bristol Business School and
18   Bristol Law School who provide a range of undergraduate, postgraduate,
19   higher research and professional courses covering a wide variety of
20   business and law subjects.
21 </p>
22 </body>
23 </html>
```

Float Right Example

This is a paragraph..

The Faculty of Environment and Technology at UWE, is situated in the beautiful city of Bristol. We are a highly successful and very popular Faculty of over 6000 students and 450 staff members. The Faculty of Business and Law comprises Bristol Business School and Bristol Law School who provide a range of undergraduate, postgraduate, higher research and professional courses covering a wide variety of business and law subjects.



Float Right Example

This is a paragraph..

The Faculty of Environment and Technology at UWE, is situated in the beautiful city of Bristol. We are a highly successful and very popular Faculty of over 6000 students and 450 staff members. The Faculty of Business and Law comprises Bristol Business School and Bristol Law School who provide a range of undergraduate, postgraduate, higher research and professional courses covering a wide variety of business and law subjects.



Layout Techniques – Flexbox

- **CSS flexbox:** predictably works for different screen sizes and display devices but **IE 10 compatibility is an issue!**
 - Makes it easier to design flexible responsive structure without using **float** and **positioning**.
 - Creates a **flex container** by setting **display** property to either **flex** (rendered as block) or **inline-flex** (rendered as inline) e.g.,
 - `.flex-container {display:flex;}`
 - Inside a flex container there can be one or more **flex items**
 - `<div class="flex-container">`
 - `<div> 1 </div>`
 - `<div> 2 </div>`
 - `</div>`
 - **Note:** Everything outside a flex container and inside a flex item is rendered as usual. Flexbox defines how flex items are laid out inside a flex container.
 - **Self-study** flexbox at : http://www.w3schools.com/css/css3_flexbox.asp
 - See examples at: https://www.w3schools.com/html/html_layout.asp

Flex-box example

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  |
5  <style>
6  .flex-container {
7  |   display: flex;
8  |   background-color: lightgreen;
9  | }
10 .flex-container > div {
11 |   background-color: red;
12 |   margin: 10px;
13 |   padding: 10px;
14 |   font-size: 20px;
15 | }
16 </style>
17 </head>
18 <body>
19 <h1>Flex Container Example </h1>
20 <div class="flex-container">
21 |   <div>FET</div>
22 |   <div>FBL</div>
23 |   <div>HAS</div>
24 </div>
25 </body>
26 </html>

```

Flex Container Example



FET FBL HAS

Layout Techniques - Flexbox

- Layout model used in modern web pages
- Dual axis, single dimension model
- **flex-direction** determines the direction in which items are laid out; the **main axis**
 - Possible values are **row**, **row-reverse**, **column**, **column-reverse**

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

Flex Container Example



Layout Techniques - Flexbox

- **Flex-wrap** property specifies whether the flex items should wrap or not.

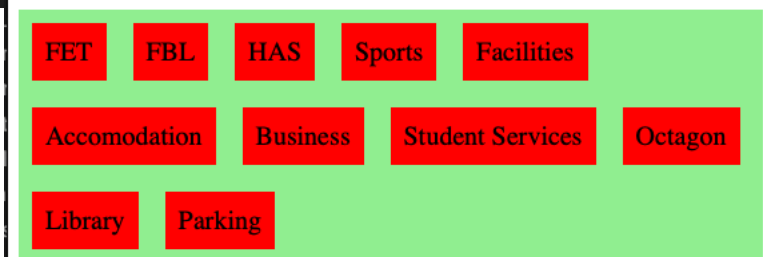
```
<h1>Flex Container Example </h1>
<div class="flex-container">
  <div>FET</div>
  <div>FBL</div>
  <div>HAS</div>
  <div>Sports</div>
  <div>Facilities</div>
  <div>Accomodation</div>
  <div>Business</div>
  <div>Student Services</div>
  <div>Octagon</div>
  <div>Library</div>
  <div>Parking</div>
</div>
```

Flex Container Example



```
.flex-container {
  display: flex;
  background-color: lightgreen;
  flex-direction: row;
  flex-wrap: wrap;
}
```

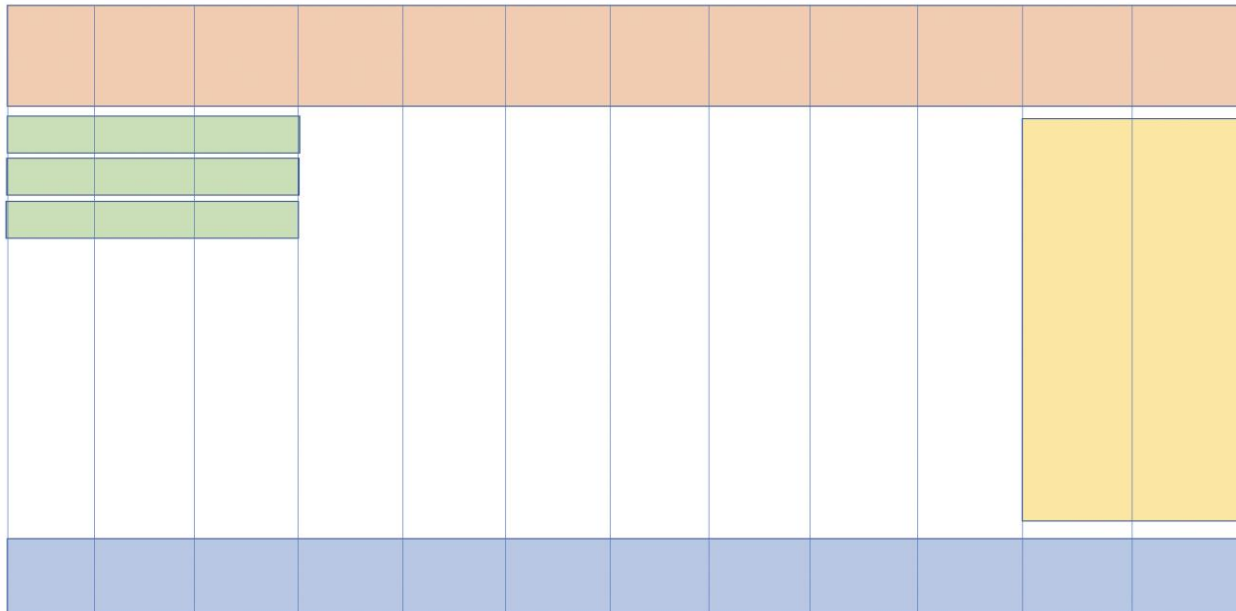
Flex Container Example



- Try flex-container properties: flex-flow, justify-content, align-item, align-content
- https://www.w3schools.com/css/css3_flexbox_container.asp

Layout Techniques – Grid view

- **CSS Grid View:** layout in rows and column makes it easier to design web pages and no need for float and positioning.
 - Does not work in IE and Edge 15 or earlier!



- We'll cover this with RWD.
https://www.w3schools.com/css/css_rwd_grid.asp

Summary

- So far we covered basics of CSS
- How to use CSS
- Style rules
- Selector, Ids, classes, Div
- Layouts
- There are **number of CSS properties** for different CSS rules e.g., background, text, fonts, tables, links, list, box model, border, outline, margin, padding, etc. **Not all can be covered in a lecture!!**
- **IMPORTANT READ: You must get yourself familiarise with different CSS properties to use them when needed – A list is here:**
<http://www.w3schools.com/cssref/default.asp>
- Check some nice CSS Templates by using float, flexbox and grid layout module - https://www.w3schools.com/css/css_templates.asp
- Next – More advanced CSS features