

# OIM3690 - Web Technologies



# Lists

# Types of Lists

- Unordered lists
  - Bulleted list of items
- Ordered Lists
  - Numbered list of items
- Description Lists
  - List of terms and their descriptions

# Syntax

- Unordered lists:
  - defined by `<ul>` and `</ul>` tags
  - each item on the list: `<li>` and `</li>` tags
  - **Syntax:**

```
<ul>  
  <li>...</li>  
  <li>...</li>  
</ul>
```

- Ordered lists:
  - defined by `<ol>` and `</ol>`
- Description lists:
  - defined by `<dl>` and `</dl>`

# List examples:

```
<h2>Ingredients for Lemonade:</h2>
<ul>
  <li>1 3/4 cups white sugar</li>
  <li>1 1/2 cups lemon juice</li>
  <li>8 cups water</li>
</ul>

<h2>To-Do List Today:</h2>
<ol>
  <li>wake up</li>
  <li>eat</li>
  <li>sleep again</li>
</ol>

<h2>All About Web:</h2>
<dl>
  <dt>Web page</dt>
  <dd>A document which can be displayed in a web browser</dd>
  <dt>Website</dt>
  <dd>
    A collection of web pages which are grouped together and usually
    connected together in various ways.
  </dd>
  <dt>web server</dt>
  <dd>A computer that hosts a website on the Internet.</dd>
</dl>
```

## Ingredients for Lemonade:

- 1 3/4 cups white sugar
- 1 1/2 cups lemon juice
- 8 cups water

## To-Do List Today:

1. wake up
2. eat
3. sleep again

## All About Web:

### Web Page

A document which can be displayed in a web browser

### Web Site

A collection of web pages which are grouped together and usually connected together in various ways.

### web server

A computer that hosts a website on the Internet.

# Nesting lists

```
<h3>Web technologies</h3>
<ul>
  <li>Basics
    <ol>
      <li>HTML
        <ul>
          <li>baisc elements</li>
          <li>tables</li>
          <li>forms</li>
        </ul>
      </li>
      <li>CSS</li>
    </ol>
  </li>
  <li>Scripting
    <ol>
      <li>JavaScript</li>
      <li>Web APIs</li>
    </ol>
  </li>
</ul>
```

## Web technologies

- Basics
  1. HTML
    - baisc elements
    - tables
    - forms
  2. CSS
- Scripting
  1. JavaScript
  2. Web APIs
  3. Web Frameworks

# Using CSS with lists

- You can format the text in a list using CSS
  - font, color, text, background, alignment etc.
- One specific formatting – using an image for the bullet in unordered lists
  - `ul {list-style-type: disc | circle | square;}`
    - You can set the bullet to any one of disc/circle/square
  - `ul {list-style-image: url (image-file-name);}`
    - e.g. if the bullet image you want is 😊 (*smiley.gif*)

```
ul {list-style-image: url(images/smiley.gif)};
```

# Exercise: creating lists

- Create *ex05.html*
- Add a nested unordered list of the courses you took last semester and this semester.
  - Add "My Courses" as `h2`.
  - Use "smiley.gif" as the bullet image for courses.
- Add an ordered list of the countries that ranked top 4 in 2018 FIFA World Cup.
  - Add "2022 FIFA World Cup Ranking" as `h2`.
  - Add national flag image before each country name (not as bullet image)
    - You can find the images under [resources/templates/images](#)
    - Resize the flags, e.g. `height="30"`
- Check it using Chrome.



# More on CSS

# Block-level elements vs. inline elements

- Read about [block-level elements](#) and [inline elements](#)
  - What ***block-level elements*** have we used?
  - What about ***inline elements***?
  - What is the difference?

# Using Images as Background

- Using CSS, we can set background images for web pages or even part of web pages using the following properties:

- `background-image: url("image_filename.jpg")`
- `background-attachment: fixed | scroll`
- `background-repeat: repeat | norepeat | repeat-x | repeat-y`
- `background-position: center | top | bottom | left | right | (x,y)`

- You can set these independently or **all** at once:

- Example:

```
body {background: url(file.jpg) center fixed repeat};
```

- More `background-image` property [examples](#)

# Positioning Images

- `float : none | left | right`
  - Allows the image to move to one side and the rest of the content flows around it on the other side
  - `float` implies the use of the ***block*** layout.
- `clear: both | left | right | none`
  - Allows the side of the image to be clear of any other *floating* content (where the computed value of `float` is not `none` )
  - i.e., prevents content from flowing around on that side
- `float` and `clear` are not only for images.
  - See examples using `float` for [Horizontal Navigation Bar](#)

# More Positioning

- `float` does not let you specify where exactly you want the image to be
- The combination of the following two will allow you to do this:

```
{position: relative | absolute | static}  
{left: auto | distance-value (pt or px)}  
{top: auto | distance-value (pt or px)}
```

- more `position` property [examples](#)
- `z-index`
  - allows you to decide whether the element must be above or below other elements that overlap its position
  - `z-index` property [example](#)
    - [a deeper explanation](#) on `z-index`

## Exercise: *ex05.html* (cont.)

- Add "*music.gif*" as the background image for the page.
- Add the image "*beaver.jpg*" and position it so that it is at - the bottom of the page and horizontally centered.
- Add the words "*Go Beavers!*" (as `h3`) and place it **over** the beaver image. Hints:
  - using different `z-index` for `img` and `h3`
  - using `h3 {position: relative; top: ?; z-index: ?}`
- Save the file *ex05.html*
- Update *sitemap.html* and **commit/push** to GitHub.

# Questions?

