Web Technologies and Programming

Lecture 15

CSS 3

Summary of the previous lecture

- The process to design a page layout
  - Determining requirements
  - Grouping and categorization
- Key element for each page
- Translating design into code
  - Liquid vs Fixed design
  - The Div tag
  - Coding the design
- Coding a page using divs and CSS
- Positioning and resizing divs.

Outline

- **Introduction to CSS3.**
- **CSS3 Properties.**
- **Rounded Corners, Box Shadows.**
- **CSS3 Transformations.**
- **CSS3 Media Queries.**

1. Introduction to CSS 3

- **Draft Published in 1999, Released in 2012.**
- **Backward Compatible with CSS2 and CSS1.**
- **CSS3 has been split into different modules.**
- **It also contains Old CSS Specification. But some old CSS tags has been removed in this version.**
- **Fully Supported in only modern browsers like Google Chrome, Internet Explorer 11 etc.**

1. Introduction to CSS 3

- **Designed to be smaller and faster than other CSS frameworks.**
- **Designed to be easier to learn, and easier to use than other CSS frameworks.**
- **Designed to provide CSS equality for all devices: PC, laptop, tablet, and mobile.**
- **Designed to use standard CSS only**
- **Designed to speed up mobile HTML apps.**

1. Introduction to CSS 3

- **CSS3 is split up into "modules". Some of the most important CSS3 modules are**
  - **Selectors**
  - **Box Model**
  - **Backgrounds and Borders**
  - **Text Effects**
  - **2D/3D Transformations**
  - **Animations**
  - **Multiple Column Layout**
  - **User Interface**

1.1 CSS3 Rounded Corners.

- With CSS3, you can give any element "rounded corners", by using the border-radius property.
- You can specify each corner separately if you wish. Here are the rules:
  - Four values: first value applies to top-left, second value applies to top-right, third value applies to bottom-right, and fourth value applies to bottom-left corner
  - Three values: first value applies to top-left, second value applies to top-right and bottom-left, and third value applies to bottom-right
  - Two values: first value applies to top-left and bottom-right corner, and the second value applies to top-right and bottom-left corner
  - One value: all four corners are rounded equally

1.1 CSS3 Rounded Corners.

```
#rcorners4 {
    border-radius: 15px 50px 30px 5px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}

#rcorners5 {
    border-radius: 15px 50px 30px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}

#rcorners6 {
    border-radius: 15px 50px;
    background: #73AD21;
    padding: 20px;
    width: 200px;
    height: 150px;
}
```

1. Four values - border-radius: 15px 50px 30px 5px:

2. Three values - border-radius: 15px 50px 30px:

3. Two values - border-radius: 15px 50px:

1.2 CSS3 Border Images.

- **The CSS3 border-image property** allows you to specify an image to be used instead of the **normal border around an element**.
- **The property has three parts:**
  - **The image to use as the border**
  - **Where to slice the image**
  - **Define whether the middle sections should be repeated or stretched**

1.2 CSS3 Border Images.

- **The border-image property takes the image and slices it into nine sections, like a tic-tac-toe board. It then places the corners at the corners, and the middle sections are repeated or stretched as you specify.**
- **Note: For border-image to work, the element also needs the border property set!**

1.2 CSS3 Border Images.

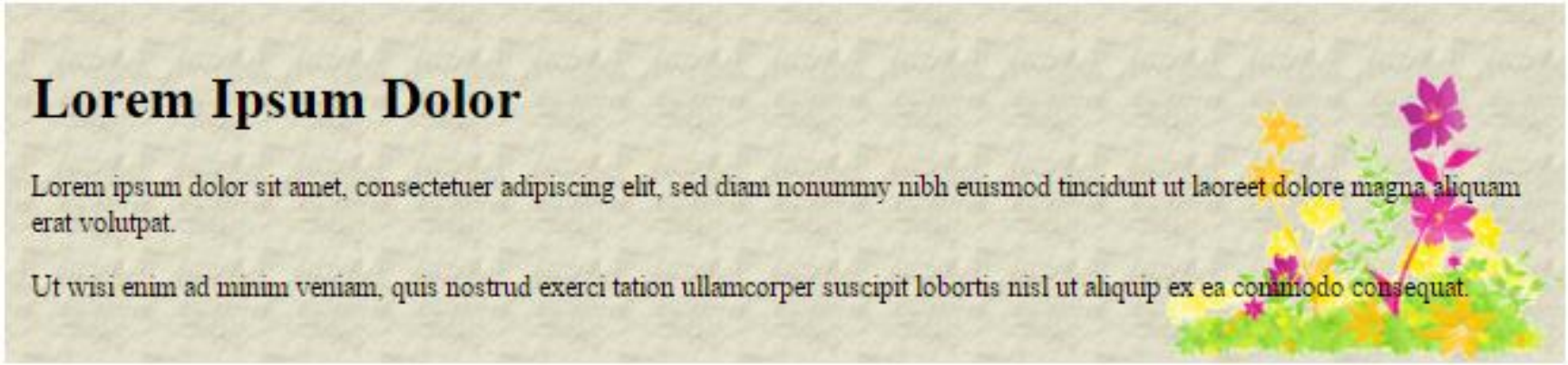
An image as a border!

```
#borderimg {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 30 round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 30 round; /* Opera 11-12.1 */
    border-image: url(border.png) 30 round;
}
```

1.3 CSS3 Background.

- **CSS3 allows you to add multiple background images for an element, through the background-image property.**
- **The different background images are separated by commas, and the images are stacked on top of each other, where the first image is closest to the viewer.**

1.3 CSS3 Background.




```
#example1 {
    background-image: url(img_flwr.gif), url(paper.gif);
    background-position: right bottom, left top;
    background-repeat: no-repeat, repeat;
}
```

1.3 CSS3 Background.

- **Background-size:**
  - The CSS3 background-size property allows you to specify the size of background images.
  - Before CSS3, the size of a background image was the actual size of the image. CSS3 allows us to re-use background images in different contexts.
  - The size can be specified in lengths, percentages, or by using one of the two keywords: contain or cover.
- **Example:**
  - background-size: 100px 80px;
  - background-size: contain;
  - background-size: cover;

1.3 CSS3 Background.

- **Background-origin:**
  - The CSS3 background-origin property specifies where the background image is positioned.
  - **The property takes three different values:**
    - **border-box** - the background image starts from the upper left corner of the border
    - **padding-box** - (default) the background image starts from the upper left corner of the padding edge
    - **content-box** - the background image starts from the upper left corner of the content
  - Example:
    - background-origin: content-box;

1.4 CSS3 Gradients.

- **CSS3 gradients let you display smooth transitions between two or more specified colors.**
- **Elements with gradients look better when zoomed, because the gradient is generated by the browser.**
- CSS3 defines two types of gradients:
  - **Linear Gradients (goes down/up/left/right/diagonally)**
  - **Radial Gradients (defined by their center)**

1.4.1 CSS3 Linear Gradients.

- **To create a linear gradient** you must define at **least two color stops**. Color stops are the colors you want to render smooth transitions among.
- **You can also set a** starting point and a direction (or an angle) along with the gradient effect.
- Syntax:
  - background:linear-gradient(*direction*,*color-stop1*, *color-stop2, ...*);

1.4.1 CSS3 Linear Gradients. (Top to bottom)

```
#grad {
    background: red; /* For browsers that do not support gradients */
    background: -webkit-linear-gradient(red, yellow); /* For Safari 5.1 to 6.0 */
    background: -o-linear-gradient(red, yellow); /* For Opera 11.1 to 12.0 */
    background: -moz-linear-gradient(red, yellow); /* For Firefox 3.6 to 15 */
    background: linear-gradient(red, yellow); /* Standard syntax */
}
```

1.4.1 CSS3 Linear Gradients. (Left to Right)

```
#grad {
  background: red; /* For browsers that do not support gradients */
  background: -webkit-linear-gradient(left, red , yellow); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(right, red, yellow); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(right, red, yellow); /* For Firefox 3.6 to 15 */
  background: linear-gradient(to right, red , yellow); /* Standard syntax */
}
```

1.4.1 CSS3 Linear Gradients. (Diagonal)

```
#grad {
  background: red; /* For browsers that do not support gradients */
  background: -webkit-linear-gradient(left top, red, yellow); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(bottom right, red, yellow); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(bottom right, red, yellow); /* For Firefox 3.6 to 15 */
  background: linear-gradient(to bottom right, red, yellow); /* Standard syntax */
}
```

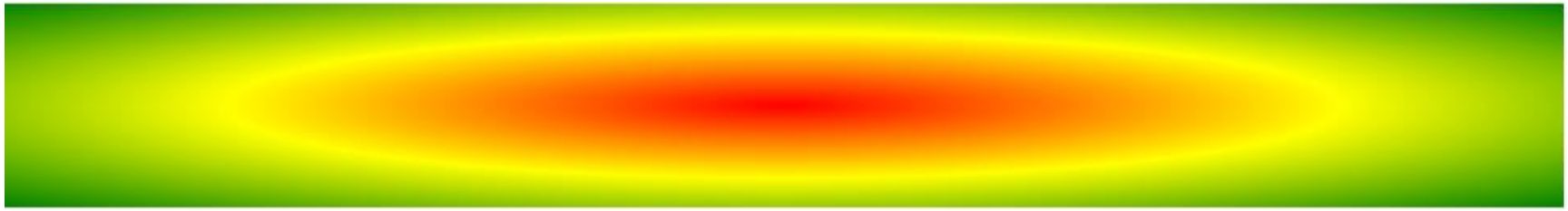
1.4.1 CSS3 Linear Gradients. (Using Angles)

```
#grad {
  background: red; /* For browsers that do not support gradients */
  background: -webkit-linear-gradient(-90deg, red, yellow); /* For Safari 5.1 to 6.0 */
  background: -o-linear-gradient(-90deg, red, yellow); /* For Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(-90deg, red, yellow); /* For Firefox 3.6 to 15 */
  background: linear-gradient(-90deg, red, yellow); /* Standard syntax */
}
```

1.4.2 CSS3 Radial Gradients.

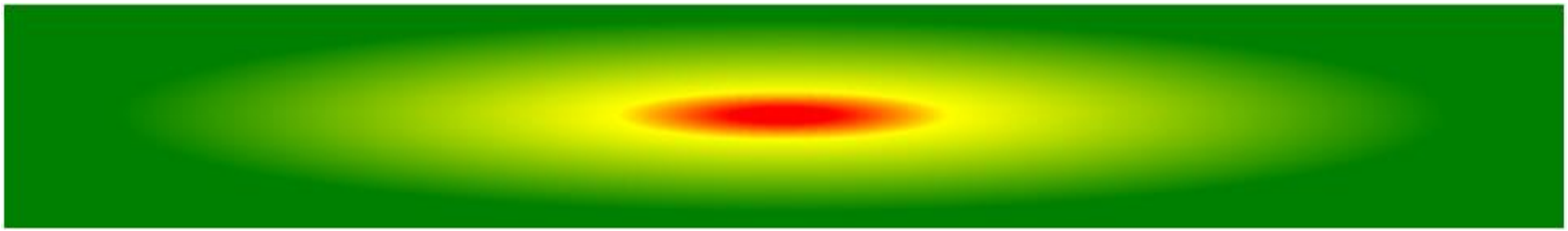
- **A radial gradient is defined by its center.**
- **To create a radial gradient you must also define at least two color stops.**
- **Syntax:**
  - **background: radial-gradient(*shape size* at *position, start-color, ..., last-color*);**

1.4.2 CSS3 Radial Gradients. (Evenly Spaced Color Stops)



```
#grad {
  background: red; /* For browsers that do not support gradients */
  background: -webkit-radial-gradient(red, yellow, green); /* Safari 5.1 to 6.0 */
  background: -o-radial-gradient(red, yellow, green); /* For Opera 11.6 to 12.0 */
  background: -moz-radial-gradient(red, yellow, green); /* For Firefox 3.6 to 15 */
  background: radial-gradient(red, yellow, green); /* Standard syntax */
}
```

1.4.2 CSS3 Radial Gradients. (Differently Spaced Color Stops)



```
#grad {
  background: red; /* For browsers that do not support gradients */
  background: -webkit-radial-gradient(red 5%, yellow 15%, green 60%); /* Safari 5.1-6.0 */
  background: -o-radial-gradient(red 5%, yellow 15%, green 60%); /* For Opera 11.6-12.0 */
  background: -moz-radial-gradient(red 5%, yellow 15%, green 60%); /* For Firefox 3.6-15 */
  background: radial-gradient(red 5%, yellow 15%, green 60%); /* Standard syntax */
}
```

1.5 CSS3 Shadows Effects.

- **We can add shadows to your content using CSS3 Shadow effects.**
- **There are two types of shadows we can use:**
  - **Text-shadow**
  - **Box-shadow**

1.5.1 CSS3 text-shadow.

Text shadow effect!

```
text-shadow: 2px 2px;
```

Text shadow effect!

```
text-shadow: 2px 2px red;
```

Text shadow effect!

```
text-shadow: 2px 2px 5px red;
```

Text shadow effect!

```
text-shadow: 0 0 3px #FF0000;
```

1.5.2 CSS3 box-shadow.

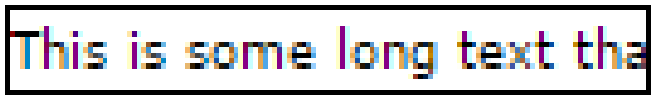
This is a yellow element with a black box-shadow

```
box-shadow: 10px 10px;
```

This is a yellow element with a grey box-shadow

```
box-shadow: 10px 10px grey;
```

1.6 CSS3 text-overflow.




```
text-overflow: clip;
```


```
text-overflow: ellipsis;
```

1.6 CSS3 word-wrapping.

This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

Without word wrap

This paragraph contains a very long word: thisisaveryveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

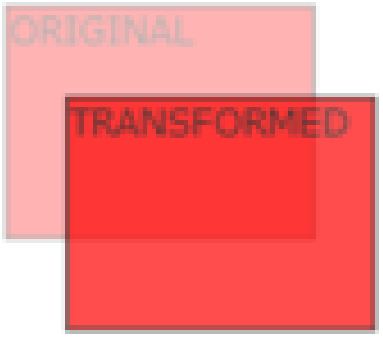
```
word-wrap: break-word;
```

1.7 CSS3 2-D Transforms.

- CSS3 transforms allow you to translate, rotate, scale, and skew elements.
- A transformation is an effect that lets an element change shape, size and position.
- Following are some css properties to apply 2-D Transforms
  - translate()
  - rotate()
  - scale()
  - skewX()
  - skewY()
  - matrix()

1.7.1 CSS3 2-D Transforms (Translate).

- The translate() method moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).




```
-ms-transform: translate(50px,100px);
```

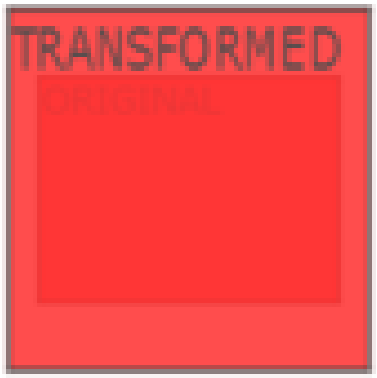
1.7.2 CSS3 2-D Transforms (Rotate).

- The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.


```
-ms-transform: rotate(20deg);
```

1.7.3 CSS3 2-D Transforms (Scale).

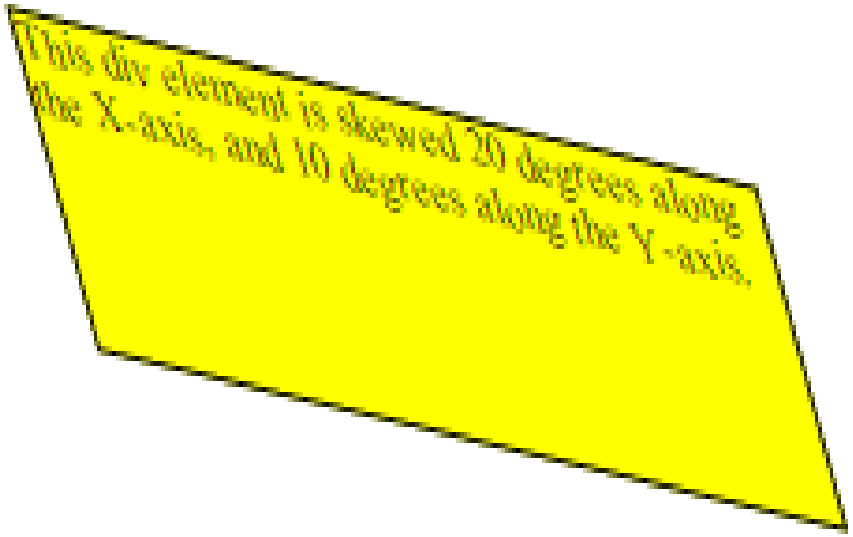
- The scale() method increases or decreases the size of an element (according to the parameters given for the width and height).




```
-ms-transform: scale(2,3);
```

1.7.4 CSS3 2-D Transforms (Skew).

- The skew() method skews an element along the X and Y-axis by the given angles.




```
-ms-transform: skew(20deg, 10deg);
```

1.7.5 CSS3 2-D Transforms (Matrix).

- The matrix() method combines all the 2D transform methods into one.


```
-ms-transform: matrix(1, -0.3, 0, 1, 0, 0);
```

1.8 CSS3 3-D Transforms.

- **CSS3 allows you to format your elements using 3D transformations.**
- Following are some css properties to apply 3-D Transforms
  - **rotateX()**
  - **rotateY()**
  - **rotateZ()**

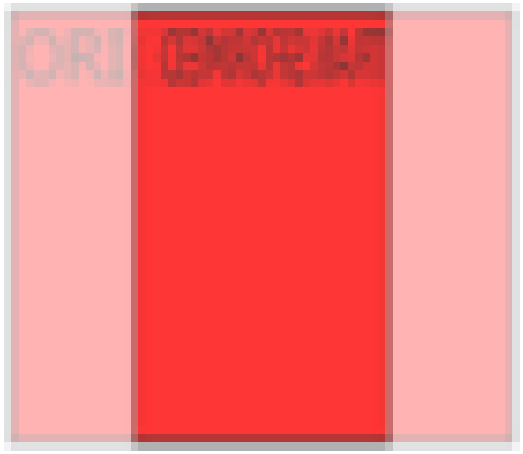
1.8.1 CSS3 3-D Transforms (rotateX).

- The rotateX() method rotates an element around its X-axis at a given degree:


```
transform: rotateX(150deg);
```

1.8.2 CSS3 3-D Transitions (rotateY)

- The rotateY() method rotates an element around its Y-axis at a given degree:



```
transform: rotateY(130deg);
```

1.9 CSS3 Media Queries

- In CSS2, "@media" property made it possible to define different style rules for different media types.
- **For example**, You could have one set of style rules for computer screens, one for printers, one for handheld devices, one for television-type devices, and so on.

1.9 CSS3 Media Queries

- **Media queries can be used to check many things, such as:**
  - **width and height of the viewport**
  - **width and height of the device**
  - **orientation (is the tablet/phone in landscape or portrait mode?)**
  - **resolution**

1.9 CSS3 Media Queries

- **A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.**

```
@media not|only mediatype and (expressions) {
    CSS-Code;
}
```

1.9 CSS3 Media Queries

- **The result of the query is true if the specified media type matches the type of device the document is being displayed on and all expressions in the media query are true.**
- **When a media query is true, the corresponding style sheet or style rules are applied, following the normal cascading rules.**
- **Unless you use the not or only operators, the media type is optional and the all type will be implied.**

1.9 CSS3 Media Queries

- **We can have different style sheets for different screen types.**
- **We can specify this condition in <link> tag, when we are associating our external style sheet. media="mediatype and|not|only (expressions)"**
- **<link rel="stylesheet" href="sample.css">**

1.9 CSS3 Media Queries (Media Types)

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

Summary

- Introduction to CSS3.
- CSS3 modules
  - Selectors
  - Box Model
  - Backgrounds and Borders
  - Text Effects
  - 2D/3D Transformations
  - Animations
  - Multiple Column Layout
  - User Interface
- CSS3 Rounded Corners
- CSS3 Properties.
- CSS3 border-image property
- CSS3 Border Images

Summary

- CSS3 Gradients
  - CSS3 Linear Gradients
  - CSS3 Radial Gradients
- CSS3 text-shadow
- CSS3 text-overflow
- CSS3 word-wrapping
- CSS3 2-D Transforms
    - translate()
    - rotate()
    - scale()
    - skewX()
- CSS3 3-D Transforms
    - rotateX()
    - rotateY()
    - rotateZ()
- CSS3 Media Queries

THANK YOU