CSS3 Notes

CSS3:

* Latest standard for CSS
* Completely backwards compatible w/ earlier versions of CSS
* CSS3 has been split into "modules". It contains the "old CSS specification" (which has been split into smaller pieces). In addition, new modules are added
* Selectors
* Box model
* Backgrounds/Borders
* Image values & replaced content
* Text effects
* 2D/3D Transformations
* Animations
* Multiple column layout
* User interface

Rounded Corners

* border-radius property

background-image:

* Allows for multiple background images
* #example1 {  
      background-image: url(img\_flwr.gif), url(paper.gif);  
      background-position: right bottom, left top;  
      background-repeat: no-repeat, repeat;  
  }
* #example1 {  
      background: url(img\_flwr.gif) right bottom no-repeat, url(paper.gif) left top repeat;  
  } //shorthand

background-origin 🡪 specifies where the background is positioned

background-clip 🡪 specifies the painting area of the background

**GRADIENTS:**

\*CSS3 defines two types of gradients

* Linear Gradients (goes down/up/left/right/diagonally)
* **Syntax:**

background: linear-gradient(direction, color-stop1, color-stop2, ...);

* + Top to Bottom (Default)

background: linear-gradient(red, yellow);

* + Left to Right

background: linear-gradient(to right, red , yellow);

* + Diagonal – specifies both horizontal & vertical starting points

background: linear-gradient(to bottom right, red, yellow);

* Angles:
* Gives you more control over the direction of the gradient
* **Syntax:**

background: linear-gradient(angle, color-stop1, color-stop2);

* ex:

background: linear-gradient(-90deg, red, yellow);

* Multiple Color Stops
* Transparency:
* Used to create fading effects
* Use the **rgba() function** to define the color stops
* The last parameter in the function is a value of 0 to 1 & defines transparency of the color🡪 0 = full transparency & 1 = full color

ex:

background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));

Radial Gradients (defined by their center)

* **Syntax:**

background: radial-gradient(shape size at position, start-color, ..., last-color);

* Default: shape is ellipse, size is farthest-corner, and position is center
* Circle Gradient:

background: radial-gradient(circle, red, yellow, green);

* Differently spaced out color stops:

background: radial-gradient(red 5%, yellow 15%, green 60%);

**TEXT:**

**CSS3 WEB FONTS:**

\*Allow Web Designers to use fonts that are not installed on the user’s computer

\*Your "own" fonts are defined within the CSS3 @font-face rule\*

**TrueType Fonts (TTF)**

TrueType is a font standard developed in the late 1980s, by Apple and Microsoft. TrueType is the most common font format for both the Mac OS and Microsoft Windows operating systems.

**OpenType Fonts (OTF)**

OpenType is a format for scalable computer fonts. It was built on TrueType, and is a registered trademark of Microsoft. OpenType fonts are used commonly today on the major computer platforms.

**The Web Open Font Format (WOFF)**

WOFF is a font format for use in web pages. It was developed in 2009, and is now a W3C Recommendation. WOFF is essentially OpenType or TrueType with compression and additional metadata. The goal is to support font distribution from a server to a client over a network with bandwidth constraints.

* Must first define a name for the font (e.g. myFirstFont), and then point to the font file
* @font-face {  
      font-family: myFirstFont;  
      src: url(sansation\_light.woff);  
  }
* To use the font for an HTML element, refer to the name of the font (myFirstFont) through the font-family property:
* div {  
      font-family: myFirstFont;  
  }
* font-weight: bold;

**CSS3 TRANSFORMS:**

\*An effect that lets an element change shape, size, & position

\*Allows you to translate, rotate, scale, & skew elements

CSS3 supports 2D & 3D transformations

2D Transformation methods:

* translate()
* moves an element from its current position (according to the parameters given for the X-axis and the Y-axis)
* ex: moves element 50px to the right and 100px down from its current position

transform: translate(50px, 100px);

* rotate()
* rotates an element clockwise or counter-clockwise according to a given degree
* ex:

transform: rotate(20deg); //moves element clockwise

transform: rotate(-20deg); //moves element counter-clockwise

* scale()
* increases or decreases the size of an element (according to the parameters given for the width & height)
* ex:

transform: scale(2, 3); //increases element to 2x its width & 3x its height

transform: scale(2, 3); //decreases element to half its original width & height

* skewX()
* skews an element along the X-axis by the given angle
* ex:

transform: skewX(20deg);

* skewY()
* skews an element along the X-axis by the given angle
* ex:

transform: skewY(20deg);

* matrix()
* combines all the 2D transformations into one
* take six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements
* matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())
* ex:

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

3D Transformation Methods:

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

transform: rotateX(150deg);

* rotateY():
* rotates an element around its Y-axis as a given degree
* ex:

transform: rotateY(130deg);

* rotateZ():
* rotates an element around its Z-axis as a given degree
* ex:

transform: rotateZ(90deg);

**CSS3 TRANSITIONS:**

\* Allows you to change property values smoothly (from one value to another), over a given duration

To create a Transition effect 🡪 must specify:

\*The CSS property you want to add an effect to

\*The duration of the effect

* Shows a 100px \* 100px red <div> element. The <div> element has also specified a transition effect for the width property, with a duration of 2 seconds:
* transition effect will start when the specified CSS property (width) changes value

div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-transition: width 2s; /\* Safari \*/  
    transition: width 2s;  
}

* Specifies a new value for the width property when a user mouses over the <div> element:

div:hover {  
    width: 300px;  
}

* Change Several Property Values:

ex: transition: width 2s, height 4s;

* Specify the Speed Curve of the Transition:
* Use property 🡪 transition-timing-function
* Delay the Transition Effect:
* Use property 🡪 transition-delay
* ex: transition-delay: 1s;
* Transition + Transformation:
* ex: adds a transformation to the transition effect

transition: width 2s, height 2s, transform 2s;

**CSS3 ANIMATIONS:**

\* Allows animation of most HTML elements without using JavaScript or Flash

\*Lets an element gradually change from one style to another

To use Animation effect 🡪 must specify some keyframes for the animation

(Keyframes hold what styles the element will have at certain times)

* @keyframes Rule:
* CSS Styles specified inside this rule – animations will gradually change from the current style to the new style at certain times
* In order for the animation to work 🡪 must bind the animation to an element
* ex: binds the "example" animation to the <div> element; animation will last for 4 seconds, & will gradually change the background-color of the <div> element from "red" to "yellow":

/\* The animation code \*/  
@keyframes example {  
    from {background-color: red;}  
    to {background-color: yellow;}  
}  
  
/\* The element to apply the animation to \*/  
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
}

- animation-duration property🡪 defines how long an animation should take to complete

* animation-delay 🡪 specifies a delay for the start of an animation
* animation-iteration-count 🡪 specifies the number of times an animation should run
* animation-direction 🡪 specifies if the animation should be played forwards, backwards, or in alternate cycles
* animation-timing-function 🡪 specifies the speed curve of the animation
* animation-fill-mode 🡪 specifies a style for the target element when the animation is not playing
* Animation Shorthand Property:
* Combines animation-

🡪 name, duration, timing-function, delay, iteration-count, & direction

animation: example 5s linear 2s infinite alternate;

**CSS3 MULTIPLE COLMNS:**

\*CSS3 Multi-Column Layout allows easy definition of multiple column text\*

CSS3 Multi-Column Properties:

* column-count:
* column-gap:
* column-rule-style:
* column-rule-width:
* column-rule-color:
* column-rule:
* column-span:
* column-width:

**CSS3 USER INTERFACE:**

\*New features such as resizing elements, outlines, & box-sizing

Resizing:

* **resize** property: specifies whether or not an element should be resizable by the user
* ex: allows user to resize only the width of the div element

div {  
    resize: horizontal;  
    overflow: auto;  
}

* ex: allows user to resize only the height

resize: vertical;

* ex: allows user to resize both the height & width

resize: both;

Outlining:

* **outline-offset** property: adds space between an outline and the edge or border of an element

\*Outlines differ from borders in three ways:

- An outline is a line drawn around elements, outside the border edge

- An outline does not take up space

- An outline may be non-rectangular