

CSS3: Expand your Horizons

Instructor: Donnie Seigler

Day One

I. New CSS3 Selectors

A. Selectors are Patterns Used to Select the Element(s) You Want to Style

1. **p** - Tag Name or Element
2. **#id** - Element ID
3. **.classname** - Element Class
4. ***** - Wildcard

B. Combining Selectors

1. **div, p** - Select for `<div>` and `<p>`
2. **div p** - Select for All `<p>` Inside of `<div>`
3. **div > p** - Select for All `<p>` with a Parent `<div>`
4. **div + p** - Select for All `<p>` Placed Immediately After `<div>`
5. **p ~ ul** - Select for All `` that are Preceded by a `<p>`

C. Selecting for Attribute Values

1. **[target]** - Select for All Elements with a **target** Attribute
2. **[target=_blank]** - Select for All Elements with **target="_blank"**
3. **[title~=flower]** - Select for All Elements with a **title** Attribute Containing the Word "flower"
4. **[title]=flower]** - Select for All Elements with a **title** Attribute Starting with the Word "flower"
5. **a[href^="https"]** - Select for All `<a>` with **href** Attribute Value Starting with "https"

6. `a[href$=".pdf"]` - Select for All `<a>` with `href` Attribute Value Ending with ".pdf"
7. `a[href*="compuskills"]` - Select for All `<a>` with `href` Attribute Value Containing "compuskills"

D. Select Elements Based on Their State Using CSS Psuedo Classes

1. `p::after` - Insert Something After the Content in Each `<p>`
2. `p::before` - Insert Something Before the Content in Each `<p>`
3. `input:checked` - Selects for All Checked `<input>` Elements
4. `input:disabled` - Selects for All Disabled `<input>` Elements
5. `p:empty` - Selects for All `<p>` Elements that Have No Children
6. `input:enabled` - Selects for All Enabled `<input>` Elements
7. `p:first-child` - Selects for All `<p>` Elements that are the First Child of Their Parent
8. `p::first-letter` - Selects the First Letter of All `<p>` Elements
9. `p::first-line` - Selects the First Line of All `<p>` Elements
10. `p:first-of-type` - Selects for All `<p>` Elements that are the First `<p>` Elements of its Parent
11. `input:focus` - Selects for the `<input>` Element that is in Focus
12. `input:in-range` - Selects for All `<input>` Elements with Values in a Specified Range
13. `input:invalid` - Selects for All `<input>` Elements with an Invalid Value
14. `p:lang(en)` - Selects for All `<p>` Elements with a `lang` Attribute Equal to "en"
15. `p:last-child` - Selects for All `<p>` Elements that is the Last Child of its Parent
16. `p:last-of-type` - Selects for All `<p>` Elements that are the Last `<p>` Element of its Parent
17. `:not(p)` - Selects for All Elements that are Not a `<input>` Element

18. **p:nth-child(2)** - Selects for All `<p>` Elements that are the Second Child of its Parent
19. **p:nth-last-child(2)** - Selects for All `<p>` Elements that are the Second Child of its Parent, Counting from the Last Child
20. **p:nth-last-of-type(2)** - Selects for All `<p>` Elements that are the Last `<p>` Elements of its Parent, Counting from the Last Child
21. **p:nth-of-type(2)** - Selects for All `<p>` Elements that are the Second `<p>` Element of its Parent
22. **p:only-of-type** - Selects for All `<p>` Elements that are the Only `<p>` Element of its Parent
23. **p:only-child** - Selects for All `<p>` Elements that are the Only Child of its Parent
24. **input:optional** - Selects for All `<input>` Elements with No **required** Attribute
25. **input:out-of-range** - Selects for All `<input>` Elements that are Outside of a Specified Range
26. **input:read-only** - Selects for All `<input>` Elements with the **readonly** Attribute Specified
27. **input:read-write** - Selects for All `<input>` Elements with the **readonly** Attribute NOT Specified
28. **input:required** - Selects for All `<input>` Elements with the **required** Attribute Specified
29. **:root** - Selects the Document's Root Element, Which is Always the `<html>` Element
30. **::selection** - Selects the Portion of an Element that is Selected by a User
31. **#news:target** - Selects the Current Active `#news` Element when the `<a>` with that Name is Clicked
32. **input:valid** - Selects for All `<input>` Elements with a Valid Value

II. Text & Fonts

A. CSS1 & CSS2 Text Properties Not Covered

1. **direction**
 - a. Specifies the Text Direction/Writing Direction
 - b. **ltr** - The Writing Direction is Left-to-Right, is Default
 - c. **rtl** - The Writing Direction is Right-to-Left
2. **letter-spacing**
 - a. Increases or Decreases the Space Between Characters in a Text
 - b. **normal** - No Extra Space Between Characters, is Default
 - c. **length** - Defines an Extra Space Between Characters (negative values are allowed)
3. **text-indent**
 - a. Specifies the Indentation of the First Line in a Text-Block
 - b. **length** - Defines a Fixed Indentation in px, pt, cm, em, etc.
Default Value is 0
 - c. **%** - Defines the Indentation in % of the Width of the Parent Element
4. **text-transform**
 - a. Controls the Capitalization of Text
 - b. **none** - No Capitalization, Text Renders as it is, is Default
 - c. **capitalize** - Transforms the First Character of Each Word to Uppercase
 - d. **uppercase** - Transforms All Characters to Uppercase
 - e. **lowercase** - Transforms All Characters to Lowercase
5. **unicode-bidi**

- a. Used Together with the Direction Property to Set or Return Whether the Text Should be Overridden to Support Multiple Languages in the Same Document
- b. **normal** - Does Not Use an Additional Level of Embedding, is Default
- c. **embed** - Creates an Additional Level of Embedding
- d. **bidirectional-override** - Creates an Additional Level of Embedding, Reordering Depends on the Direction Property

6. vertical-align

- a. Sets the Vertical Alignment of an Element
- b. **baseline** - Align the Baseline of the Element with the Baseline of the Parent Element, is Default
- c. **length** - Raises or Lowers an Element by the Specified Length, Negative Values are Allowed
- d. **%** - Raises or Lower an Element in a Percent of the "line-height" Property, Negative Values are Allowed
- e. **sub** - Aligns the Element as if it was Subscript
- f. **super** - Aligns the Element as if it was Superscript
- g. **top** - The Top of the Element is Aligned with the Top of the Tallest Element on the Line
- h. **text-top** - The Top of the Element is Aligned with the Top of the Parent Element's Font
- i. **middle** - The Element is Placed in the Middle of the Parent Element
- j. **bottom** - The Bottom of the Element is Aligned with the Lowest Element on the Line
- k. **text-bottom** - The Bottom of the Element is Aligned with the Bottom of the Parent Element's Font

7. white-space

- a. **normal** - Sequences of Whitespace Will Collapse into a Single Whitespace, Text Will Wrap When Necessary & it is Default
- b. **nowrap** - Sequences of Whitespace Will Collapse into a Single Whitespace, Text Will Never Wrap to the Next Line & Text Continues on the Same Line Until a `
` is Encountered
- c. **pre** - Whitespace is Preserved by the Browser & Text Will Only Wrap on Line Breaks
- d. **pre-line** - Sequences of Whitespace Will Collapse into a Single Whitespace & Text Will Wrap When Necessary, and on Line Breaks
- e. **pre-wrap** - Whitespace is Preserved by the Browser & Text Will Wrap When Necessary, and on Line Breaks

8. word-spacing

- a. Increases or decreases the space between words in a text
- b. **normal** - Defines Normal Space Between Words (0.25em), is Default
- c. **length** - Defines an Additional Space Between Words (in, px, pt, cm, em, etc), Negative Values are Allowed

B. CSS3 Text Properties

1. text-overflow

- a. Specifies How Overflowed Content that is not Displayed Should be Signaled to the User
- b. **clip** - Clips the Text, Default
- c. **ellipsis** - Renders an Ellipsis (...) to Represent Clipped Text

2. word-wrap

- a. Allows Long Words to be Able to be Broken and Wrap Onto the Next Line

- b. **normal** - Break Words Only at Allowed Break Points
 - c. **break-word** - Allows You to Force the Text to Wrap Even if it Means Splitting it in the Middle of a Word
3. **word-break**
- a. Specifies Line Breaking Rules for Non-CJK Scripts (Chinese, Japanese, Korean)
 - b. **normal** - Is Default Value & Break Words According to Their Usual Rules
 - c. **break-all** - Lines May Break Between Any Two Letters
 - d. **keep-all** - Breaks are Prohibited Between Pairs of Letters
4. **CSS3 Text Properties Not Yet Supported or Little Support Among Browsers**
- a. **text-align-last** - Specifies How to Align the Last Line of a Text
 - b. **text-emphasis-style** - Applies Emphasis Marks to the Element's Text
 - c. **text-emphasis-color** - Defines the Foreground Color of the Emphasis Marks
 - d. **text-emphasis** - A Shorthand for Setting **text-emphasis-style** & **text-emphasis-color** in One Declaration
 - e. **text-justify** - Specifies How Justified Text Should be Aligned & Spaced

C. CSS3 Web Fonts

1. The **@font-face** Rule

- a. Web Fonts Allow Web Designers to Use Fonts that are Not Installed on the User's Computer
- b. Include the Font File on Your Web Server and it Will be Automatically Downloaded to the User When Needed
- c. Your "Own" Fonts are Defined Within the CSS3 **@font-face** Rule

2. Font Formats

- a. TrueType Fonts (TTF) - Supported by All Browsers
- b. OpenType Fonts (OTF) - Supported by All Browsers
- c. The Web Open Font Format (WOFF) - Supported by All Browsers
- d. The Web Open Font Format (WOFF 2.0)
- e. SVG Fonts/Shapes
- f. Embedded OpenType Fonts (EOT)

3. Defining Your Font

- a. Define a Name for Your Font Using font-family
- b. Point to the Location of Your Font File Using src
- c. Syntax
- d. `@font-face {`

```
font-family: fontname;
```

```
src: url(path/to/file);
```

```
}
```

4. Using Bold Text

- a. You Must Add Another `@font-face` Rule Containing Descriptors for Bold Text
- b. Use font-weight to Define Bold Fonts
- c. Browsers Will Use this Whenever a Piece of Text with the font-family Should Render as Bold
- d. This Way You Can Have Many `@font-face` Rules for the Same Font

D. Text Shadow Effects

1. The Four Values of text-shadow

- a. Horizontal Shadow (+ right/- left)
- b. Vertical Shadow (+ bottom/- top)
- c. Blur Radius

- d. Color
- 2. Multiple Text Shadows
 - a. To Add More than One Shadow to the Text Add a Comma-separated List of Shadows

iii. Multiple Columns

A. Multiple Columns

- 1. allows easy definition of multiple columns of text
- 2. All Browsers Support Multiple Columns, but Only IE & Edge Without Browser Extensions
- 3. CSS3 Multi-Column Properties
 - a. column - Shorthand Property for Setting column-width & column-count
 - b. column-count - Specifies the Number of Columns an Element Should be Divided Into
 - c. column-gap - Specifies the Gap Between the Columns
 - d. column-rule-style - Specifies the Style of the Rule Between Columns
 - e. column-rule-width - Specifies the Width of the Rule Between Columns
 - f. column-rule-color - Specifies the Color of the Rule Between Columns
 - g. column-rule - Shorthand Property for Setting All the column-rule-* Properties Above (width, style, color)
 - h. column-span - Specifies How Many Columns an Element Should Span Across
 - i. column-width - Specifies a Suggested, Optimal Width for the Columns

IV. CSS3 User Interface Properties

A. CSS3 box-sizing Property

1. Allows Us to Include the padding & border in an Element's Total width & height
2. box-sizing Values
 - a. **content-box** - Default: The width & height Properties (and min/max properties) Includes Only the Content (border, padding and margin are not included)
 - b. **border-box** - The width & height Properties (and min/max properties) Includes Content, Padding & Border, but NOT the Margin
3. You Can Set the box-sizing Property to All Elements Using the * Selector

B. CSS3 resize Property

1. Specifies Whether or Not an Element Should be Resizable by the User
2. Not Supported by IE or Edge
3. CSS3 resize Property Values
 - a. **none** - Default Value, User Cannot Resize the Element
 - b. **both** - User Can Adjust Both the Height & Width of the Element
 - c. **horizontal** - User Can Adjust the Width of the Element
 - d. **vertical** - User Can Adjust the Height of the Element

C. CSS2 outline Property

1. A Line that is Drawn Around Elements (outside the borders) to Make the Element "Stand Out"
2. A Shorthand Property Sets All the Outline Properties in One Declaration
 - a. Set in Order of outline-color, outline-style & outline-width
 - b. outline-color - Sets the Color of the Outline

- c. `outline-style` - Sets the Style of the Outline
 - d. `outline-width` - Sets the Width of the Outline
 - e. The Default Value for a Missing Property Will be Inserted, if Any
3. The `outline` Property is Similar to the `border` Property but is Not a Part of the Element's Dimensions

D. CSS3 `outline-offset` Property

- 1. Adds Space Between an Outline and the Edge or Border of an Element
- 2. The `outline` Property Value is a Length with a Default Value of 0
- 3. Not Supported by IE or Edge

v. CSS3 Media Queries

A. Media Queries are a Popular Technique for Delivering a Tailored Style Sheet to Tablets, iPhone & Androids Known as Responsive Design

B. Media Query Syntax

- 1. `@media not|only mediatype and (expressions) {`
`CSS-Code;`
`}`
- 2. 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
- 3. When a Media Query is True the Corresponding Style Sheet or Style Rules are Applied, Following the Normal Cascading Rules
- 4. Unless You Use the `not` or `only` Operators the Media Type is Optional and the `all` Type Will be Implied
- 5. Example that Will be True if Any Screen is 480px or Less in Width:
- 6. `@media screen and (max-width: 480px) {`
`CSS-Code;`
`}`

C. You Can Have an Alternate CSS Section Right Inside Your Style Sheet

D. Media Queries Can be Used to Check Many Things Such As

1. Width & Height of the Viewport
2. Width & Height of the Device
3. Orientation (is the tablet/phone in landscape or portrait mode?)
4. Resolution

E. CSS3 Media Types

1. **all** - Used for All Media Type Devices
2. **print** - Used for Printers
3. **screen** - Used for Computer Screens, Tablets, Smart-phones etc.
4. **speech** - Used for Screen Readers that "Reads" the Page Out Loud

F. Media Features

1. **aspect-ratio** - The Ratio Between the Width & Height of the Viewport
2. **color** - The Number of Bits per Color Component for the Output Device
3. **color-index** - The Number of Colors the Device Can Display
4. **device-aspect-ratio** - The Ratio Between the Width & Height of the Device
5. **device-height** - The Height of the Device
6. **device-width** - The Width of the Device
7. **grid** - Whether the Device is a Grid or Bitmap
8. **height** - The Viewport Height
9. **max-aspect-ratio** - The Maximum Ratio Between the Width & Height of the Display Area
10. **max-color** - The Maximum Number of Bits per Color Component for the Output Device
11. **max-color-index** - The Maximum Number of Colors the Device Can Display

12. **max-device-aspect-ratio** - The Maximum Ratio Between the Width & Height of the Device
13. **max-device-height** - The Maximum Height of the Device
14. **max-device-width** - The Maximum Width of the Device
15. **max-height** - The Maximum Height of the Display Area
16. **max-monochrome** - The Maximum Number of Bits per Color on a Monochrome (greyscale) Device
17. **max-resolution** - The Maximum Resolution of the Device Using dpi or dpcm
18. **max-width** - The Maximum Width of the Display Area
19. **min-aspect-ratio** - The Minimum Ratio Between the Width & Height of the Display Area
20. **min-color** - The Minimum Number of Bits per Color Component for the Output Device
21. **min-color-index** - The Minimum Number of Colors the Device Can Display
22. **min-device-aspect-ratio** - The Minimum Ratio Between the Width & Height of the Device
23. **min-device-width** - The Minimum width of the Device
24. **min-device-height** - The Minimum Height of the Device
25. **min-height** - The Minimum Height of the Display Area
26. **min-monochrome** - The Minimum Number of Bits per Color on a Monochrome (greyscale) Device
27. **min-resolution** - The Minimum Resolution of the Device using dpi or dpcm
28. **min-width** - The Minimum Width of the Display Area
29. **monochrome** - The Number of Bits per Color on a Monochrome (greyscale) Device

30. **orientation** - The Orientation of the Viewport (landscape or portrait mode)
31. **overflow-block** - How Does the Output Device Handle Content that Overflows the Viewport Along the Block Axis
32. **overflow-inline** - Can Content that Overflows the Viewport Along the Inline Axis be Scrolled
33. **resolution** - The Resolution of the Output Device Using dpi or dpcm
34. **scan** - The Scanning Process of the Output Device
35. **update-frequency** - How Quickly Can the Output Device Modify the Appearance of the Content
36. **width** - The Viewport Width

Day Two

I. CSS3 Borders

A. CSS3 Rounded Corners

1. With the CSS3 border-radius Property You Can Give Any Element Rounded Corners
2. The border-radius Property is Actually a Shorthand Property
 - a. border-top-left-radius
 - b. border-top-right-radius
 - c. border-bottom-right-radius
 - d. border-bottom-left-radius
3. Variations Using the Shorthand border-radius Property
 - a. Four Values: The Order is Top Left, Top Right, Bottom Right, Bottom Left

- b. **Three Values: The Order is Top-Left, Top Right & Bottom Left, Bottom Right**
 - c. **Two Values: The Order is Top Left & Bottom Right, Top Right & Bottom Left**
 - d. **One Values: All Four Corners are Rounded Equally**
- 4. Creating Elliptical Corners**
- a. **Use 2 Values to Define the Radii of a Quarter Ellipse**
 - b. **The First Value is Horizontal Radius**
 - c. **The Second Value is the Vertical Radius**
 - d. **If the Second Value is Omitted it is Copied from the First**
 - e. **If Either Length is Zero the Corner is Square Rather than Rounded**
 - f. **Percentages for the Horizontal Radius Refer to the Width of the Border Box**
 - g. **Percentages for the Vertical Radius Refer to the Height of the Border Box**
 - h. **Use 50% on Each Corner to Create a Perfect Ellipse or Circle**
 - i. **Individual Corner Properties Separate Each Value by a Space**
 - j. **For the Shorthand Property Separate Each Value by a "/"**
 - k. **You Can Use 2 Elliptical Values in the Shorthand Property with Each Pair Separated by a Space and the Pairs Separated by a "/"**

B. CSS3 Border Images

- 1. Set an Image to be Used as the Border Around an Element with the border-image Property**
- 2. The border-image Property is a Shorthand Property**
 - a. **border-image-source - Specifies the Path to the Image to be Used as a Border**
 - b. **border-image-slice - Specifies How to Slice the Border Image**

- c. border-image-width - Specifies the Widths of the Border Image
 - d. border-image-outset - Specifies the Amount by Which the Border Image Area Extends Beyond the Border Box
 - e. border-image-repeat - Specifies Whether the Border Image Should be Repeated, Rounded or Stretched
 - f. Generic Syntax Property Order
 - g. border-image: source slice width outset repeat;
3. The border-image Property Takes the Image and Slices it into 9 Sections Like a Tic-Tac-Toe Board
 4. It Places the Corners at the Corners and the Middle Sections are Repeated or Stretched as You Specify
 5. For border-image to Work the Element Also Needs the border Property Set
 6. More on the border-image-slice Property
 - a. The Image is Always Sliced into Nine Sections: Four Corners, Four Edges & the Middle
 - b. The "Middle" Part is Treated as Fully Transparent Unless the **fill** Keyword is Set
 - c. The border-image-slice Property Can Take from One to Four Values
 - d. If the Fourth Value is Omitted it is the Same as the Second
 - e. If the Third Value is Omitted it is the Same as the First
 - f. If the Second Value is Omitted it is the Same as the First
 - g. **number** - The Number(s) Represent Pixels (do not use px units) for Raster Images or Coordinates for Vector Images
 - h. **%** - Percentages are Relative to the Height or Width of the Image
 - i. **fill** - Causes the Middle Part of the Image to be Displayed
 7. More on the border-image-repeat Property

- a. Specifies How the Images for the Sides and the Middle Part of the Border Image are Scaled and Tiled
- b. You Can Specify Two Values Here, One for the Sides and One for the Middle
- c. If the Second Value is Omitted it is Assumed to be the Same as the First
- d. **stretch** - Default Value, the Image is Stretched to Fill the Area
- e. **repeat** - The Image is Tiled (repeated) to Fill the Area
- f. **round** - The Image is Rescaled and Tiled (repeated) to Fill the Area with a Whole Number of Tiles
- g. **space** - The Image is Tiled (repeated) to Fill the Area, but if it doesn't Fill the Area with a Whole Number of Tiles the Extra Space is Distributed Around the Tiles

II. CSS3 Backgrounds

A. CSS3 Backgrounds Allow Greater Control of the Background Element

B. CSS3 Adds Support for Multiple Backgrounds

1. You Can Add Multiple Background Images with the `background-image` Property
2. Different Background Images are Separated by Commas
3. Images are Stacked on Top of Each Other Where the First Image is Closest to the Viewer
4. Multiple Background Images Can be Specified Using Either the Individual Background Properties or the `background` Shorthand Property

C. CSS3 Background Size

1. The CSS3 `background-size` Property Allows You to Specify the Size of Background Images

2. Before CSS3 the Size of a Background Image was the Actual Size of the Image
3. CSS3 Allows Us to Re-use Background Images in Different Contexts
4. The Size Can be Specified in Lengths, Percentages, or by Using the Keywords: **contain** or **cover**
 - a. **auto** - Default Value, the background-image Contains its Natural Width & Height
 - b. **length** - Sets the Width & Height of the Background Image. The First Value Sets the Width, the Second Value Sets the Height. If Only One Value is Given, the Second is Set to **auto**
 - c. **%** - Sets the Width & Height of the Background Image in Percent of the Parent Element. The First Value Sets the Width, the Second Value Sets the Height. If Only One Value is Given, the Second is Set to **auto**
 - d. **cover** - Scale the Background Image to be as Large as Possible so that the Background Area is Completely Covered by the Background Image. Some Parts of the Background Image May Not be in View Within the Background Positioning Area
 - e. **contain** - Scale the Image to the Largest Size Such that Both its Width & Height Can Fit Inside the Content Area
5. The background-size Property also Accepts Multiple Values for Background Size
6. Use a Comma-separated List of Values When Working with Multiple Backgrounds
7. `background-size: 50px, 130px, auto;`

D. Full Size Background Image

1. If You Want a Background Image that Covers the Entire Browser Window at All Times

- a. Fill the Entire Page with the Image (no white space) Using the background-image Property
- b. Scale Image as Needed Using background-size Set to **cover**
- c. Center Image on Page Using background-position Set to **center**
- d. Do Not Cause Scrollbars by Setting background-attachment to **fixed**
- e. Make Sure the Image Doesn't Repeat by Setting background-repeat to **no-repeat**

E. CSS3 background-origin Property

1. Specifies Where the Background Image is Positioned
2. The Property Takes Three Different Values
 - a. **border-box** - The Background Image Starts from the Upper Left Corner of the Border
 - b. **padding-box** - The Background Image Starts from the Upper Left Corner of the Padding Edge (default)
 - c. **content-box** - The Background Image Starts from the Upper Left Corner of the Content Area

F. CSS3 background-clip Property

1. Specifies the Painting Area of the Background
2. The Property Takes Three Different Values
 - a. **border-box** - The Background is Painted to the Outside Edge of the Border (default)
 - b. **padding-box** - The Background is Painted to the Outside Edge of the Padding
 - c. **content-box** - The Background is Painted Within the Content Area

iii. CSS3 Colors

A. RGBA Colors

1. **RGBA Color Values are an Extension of RGB Color Values with an Alpha Channel Specifying the Opacity for a Color**
2. **An RGBA Color Value is Specified with: `rgba(red, green, blue, alpha)`**
3. **The Alpha Parameter is a Number Between 0.0 (fully transparent) and 1.0 (fully opaque)**

B. HSL Colors

1. **HSL Stands for Hue, Saturation and Lightness**
2. **An HSL Color Value is Specified with: `hsl(hue, saturation, lightness)`**
 - a. **0 (or 360) is Red**
 - b. **120 is Green**
 - c. **240 is Blue**
3. **Saturation is a Percentage Value: 100% is the Full Color**
4. **Lightness is Also a Percentage; 0% is Dark (black) and 100% is White**

C. HSLA Colors

1. **HSLA Color Values are an Extension of HSL Color Values with an Alpha Channel Specifying the Opacity for a Color**
2. **An HSLA Color Value is Specified with: `hsla(hue, saturation, lightness, alpha)` Where the Alpha Parameter Defines the Opacity**
3. **The Alpha Parameter is a Number Between 0.0 (fully transparent) and 1.0 (fully opaque)**

D. Opacity

1. **The CSS3 `opacity` Property Sets the Opacity for a Specified RGB Value**
2. **The `opacity` Property Value Must be a Number Between 0.0 (fully transparent) and 1.0 (fully opaque)**
3. **Note that the Content of the Element Will Also be Opaque**

iv. More on CSS3 Gradients

- A. CSS3 Gradients Let You Display Smooth Transitions Between Two or More Specified Colors
- B. CSS3 Gradients Reduce Download Time and Bandwidth Usage vs Using Images
- C. Elements with Gradients Look Better when Zoomed Because the Gradient is Generated by the Browser
- D. CSS3 Defines Two Types of Gradients
 - 1. Linear Gradients (goes down/up/left/right/diagonally)
 - 2. Radial Gradients (defined by their center)
- E. CSS3 Linear Gradients
 - 1. To Create a Linear Gradient You Must Define at Least Two Color Stops
 - 2. Color Stops are the Colors You Want to Render Smooth Transitions Among
 - 3. You Can Also Set a Starting Point and a Direction (or an angle) Along with the Gradient Effect
 - 4. Linear Gradient Syntax:
 - 5. `background: linear-gradient(direction, color-stop1, color-stop2, ...);`
 - 6. Keywords For Direction
 - a. **to bottom** - Starts at Top and Goes to Bottom (default)
 - b. **to top** - Starts at Bottom and Goes to Top
 - c. **to right** - Starts at Left and Goes to Right
 - d. **to left** - Starts at Right and Goes to Left
 - e. You can make a gradient diagonally by specifying both the horizontal and vertical starting positions
 - f. **to bottom right** - Starts at Top Left Corner and Goes to Bottom Right Corner
 - 7. Using Angles

- a. For More Control Over the Direction of the Gradient You Can Define a Specific Angle
 - b. The Angle is Specified as an Angle Between a Horizontal Line and the Gradient Line
 - c. The Angle is Specified Using Degrees and Written as 45deg
 - d. Syntax Using Angles
 - e. `background: linear-gradient(angle, color-stop1, color-stop2);`
8. You Can Use Multiple Color Stops with as Many Colors as You Want
9. Using Transparency
- a. CSS3 Gradients Support Transparency Which Can be Used to Create Fading Effects.
 - b. Add Transparency by Using the `rgba()` Function to Define the Color Stops
 - c. The Last Parameter in the `rgba()` Function Can be a Value from 0 to 1 Defines the Transparency Value of the Color
10. Controlling Color Stops
- a. Add a Space then a Percentage After a Color Stop to Set it's Starting Point
 - b. Horizontal & Vertical Gradient Lines Begin at Edge of Element on Starting Side to End of Element on Ending Side
 - c. Each Color Stop Starting Point is a Percentage On the Gradient Line
 - d. The Diagonal Gradient Line is Determined by the Starting Corner of the Element
11. Repeating a `linear-gradient`
- a. The `repeating-linear-gradient()` Function is Used to Repeat Linear Gradients

b. Use Color Stop Starting Points to Control Length of Color

c. Example Syntax:

d. `background: repeating-linear-gradient(color1, color2 10%, color3 20%);`

F. CSS3 Radial Gradients

1. A Radial Gradient is Defined by its Center

2. To Create a Radial Gradient You Must Also Define at Least Two Color Stops

3. Syntax

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

a. Shape is **ellipse** by Default

b. Size is **farthest-corner** by Default

c. Position is **center** by Default

d. Evenly Spaced Color Stops by Default

5. Set Shape

a. The Shape Parameter Defines the Shape

b. It Can Take the Value **circle** or **ellipse**

c. The Default Value is **ellipse**

6. Use of Different Size Keywords

a. The Size Parameter Defines the Size of the Gradient

b. **closest-side**

c. **farthest-side**

d. **closest-corner**

e. **farthest-corner**

7. Repeating a **radial-gradient**

a. The **repeating-radial-gradient()** Function is Used to Repeat Radial Gradients

- b. Use Color Stop Starting Points to Control Length of Color
- c. The Radial Gradient Line Starts at Center and Goes Out to Furthest Corner
- d. Example Syntax:
- e. `background: repeating-linear-gradient(color1, color2 10%, color3 20%);`

G. Multiple Box Shadows

- 1. Works Just Like the text-shadow Property
- 2. Define the Box Shadow using the Property box-shadow
- 3. Each Shadow Effect Must be Separated by a Comma
- 4. Example Syntax:
- 5. `box-shadow: 10px 10px 5px grey, 2px 2px 2px black;`

v. CSS3 2D Transforms

- A. CSS3 Transforms Allow You to Translate, Rotate, Scale & Skew Elements
- B. A Transformation is an Effect that Lets an Element Change Shape, Size & Position
- C. CSS3 Supports 2D & 3D Transformations
- D. Use the CSS3 Property transform in Current Browsers, for Safari Use `-webkit-transform`
- E. CSS3 2D Transform Methods
 - 1. The `translate()` Method
 - a. Moves an Element from its Current Position to the Parameters Given for the X-axis and the Y-axis
 - b. Use `translate(x,y)` Method to Move the Element Along the X-axis (x) & the Y-axis (y)
 - c. Use `translateX(n)` Method to Move the Element Along the X-axis Only

- d. Use **translateY(n)** Method to Move the Element Along the Y-axis Only
 - e. Negative Values Can be Used
2. The **rotate()** Method
- a. Rotates an Element Clockwise or Counter-clockwise According to a Given Degree
 - b. Using Negative Values Will Rotate the Element Counter-clockwise
 - c. Use the **rotate(angle)** Method to Rotate the Element
3. The **scale()** Method
- a. Increases or Decreases the Size of an Element by a Factor of Width & Height
 - b. Use the **scale(x,y)** Method to Scale Both the Elements Width (x) & Height (y)
 - c. Use the **scaleX(n)** Method to Scale the Element's Width Only
 - d. Use the **scaleY(n)** Method to Scale the Element's Height Only
4. The **skew()** Method
- a. Skews an Element Along the X-axis and Y-axis by the Given Angles
 - b. Use the **skew(x-angle,y-angle)** Method to Skew the Element Along Both the X-axis (x-angle) & the Y-axis (y-angle)
 - c. The **skewX(angle)** Method to Skew the Element Along the X-axis Only
 - d. The **skewY(angle)** Method to Skew the Element Along the Y-axis Only
5. The **matrix()** Method
- a. Combines All the 2D Transform Methods into One

b. The `matrix()` Method Take Six Parameters, Containing
Mathematic Functions, Which Allows You to Rotate, Scale, Move
(translate), and Skew Elements

c. The parameters are as follow:

```
matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY()  
)
```

d. If you Are Interested in The [Mathematics Behind the
matrix\(\) Method](#) or to Use the [Matrix Tool](#)

F. You Can Specify Multiple 2D Transformations Separated by a Space

G. `transform: rotate(90deg) scale(2, 3) translate(50px 100px);`

VI. CSS3 3D Transforms

A. CSS3 Allows You to Format Your Elements with 3D Transformations Also
Using the transform Property

B. The `rotateX()` Method

1. The `rotateX()` Method Rotates an Element Around its X-axis at a Given
Degree

C. The `rotateY()` Method

1. The `rotateY()` Method Rotates an Element Around its Y-axis at a Given
Degree

D. The `rotateZ()` Method

1. The `rotateZ()` Method Rotates an Element Around its Z-axis at a Given
Degree

E. You Can Specify Multiple 3D Transformations Separated by a Space

F. `transform: rotateX(90deg) rotateY(90deg) rotateZ(90deg);`

VII. Transitions

- A. The CSS3 transition Property Allows You to Change Property Values Smoothly Over a Given Duration
- B. To Create a Transition Effect You Must Specify Two Things
 - 1. The CSS Property You Want to Add an Effect To
 - 2. The Duration of the Effect
- C. If the Duration is Not Specified the Transition Will Have No Effect Because the Default Value is 0
- D. The Transition Effect Will Start When the Specified CSS Property Changes Value
- E. Multiple Transition Effects Can Be Specified Separated by Commas
- F. `transition: width 2s, height 2s;`
- G. Specify the Speed Curve of the Transition
 - 1. The transition-timing-function Property Specifies the Speed Curve of the Transition Effect
 - 2. The transition-timing-function Property Can Have the Following Values
 - a. **ease** - Specifies a Transition Effect with a Slow Start, then Fast and then End Slowly (default)
 - b. **linear** - Specifies a Transition Effect with the Same Speed from Start to End
 - c. **ease-in** - Specifies a Transition Effect with a Slow Start
 - d. **ease-out** - Specifies a Transition Effect with a Slow End
 - e. **ease-in-out** - Specifies a Transition Effect with a Slow Start and End
 - f. **cubic-bezier(n,n,n,n)** - Lets You Define Your Own Values in a Cubic-bezier Function
- H. Delay the Transition Effect
 - 1. The transition-delay Property Specifies a Delay (in seconds) for the Transition Effect

I. Create a Transition Effect with 2D & 3D Transformations

J. The transition Property is a Shorthand Property

1. transition-property
2. transition-duration
3. transition-timing-function
4. transition-delay
5. Example Syntax
6. `transition: width 2s linear 1s;`

VIII. CSS3 Animations

A. CSS3 Animations Allows You to Animate Most HTML Elements Without Using JavaScript or Flash!

B. What are CSS3 Animations?

1. An Animation Lets an Element Gradually Change from One Style to Another
2. You Can Change as Many CSS Properties as You Want, as Many Times as You Want
3. To Use CSS3 Animation You Must First Specify Some Keyframes for the Animation
4. Keyframes Hold What Styles the Element Will Have at Certain Times

C. The @keyframes Rule

1. When You Specify CSS Styles Inside the @keyframes Rule the Animation Will Gradually Change from the Current Style to the New Style at Certain Times
2. You Can Specify When the Style Will Change by Using the Keywords **from** & **to** (which represents 0% (start) and 100% (complete))
3. @keyframes example {
`from {background-color: red;}`

```
to {background-color: yellow;}  
}
```

4. It is Also Possible to Use Percent. By Using Percent You Can Add as Many Style Changes as You Like

5. @keyframes example {

```
0% {background-color: red;}  
25% {background-color: yellow;}  
50% {background-color: blue;}  
100% {background-color: green;}  
}
```

6. To Get an Animation to Work You Must Bind the Animation to an Element Using the animation-nameProperty

7. If the animation-duration Property is not Specified the Animation Will Have No Effect Because the Default Value is 0

D. Delay an Animation

1. The animation-delay Property Specifies a Delay for the Start of an Animation

E. Set How Many Times an Animation Should Run

1. The animation-iteration-count Property Specifies the Number of Times an Animation Should Run

2. Use the Value **infinite** to Make the Animation Continue Forever

F. Run Animation in Reverse Direction or Alternate Cycles

1. The animation-direction Property is Used to Let an Animation Run in Reverse Direction or Alternate Cycles

2. **normal** - The Animation Should be Played as Normal (default)

3. **reverse** - The Animation Should Play in Reverse Direction

4. **alternate** - The Animation Will be Played as Normal Every Odd Time and in Reverse Direction Every Even Time

5. **alternate-reverse** - The Animation Will be Played in Reverse Direction Every Odd Time and in a Normal Direction Every Even Time
6. The **reverse** & **alternate-reverse** Values are Not Supported in Safari

G. Specify the Speed Curve of the Animation

1. The animation-timing-function Property Specifies the Speed Curve of the Animation
2. The animation-timing-function Property Can Have the Following Values
 - a. **ease** - Specifies an Animation with a Slow Start, then Fast, then End Slowly (default)
 - b. **linear** - Specifies an Animation with the Same Speed from Start to End
 - c. **ease-in** - Specifies an Animation with a Slow Start
 - d. **ease-out** - Specifies an Animation with a Slow End
 - e. **ease-in-out** - Specifies an Animation with a Slow Start and End
 - f. **cubic-bezier(n,n,n,n)** - Lets You Define Your Own Values in a Cubic-bezier Function

H. Animation Shorthand Property

- I. `animation: name duration timing-function delay iteration-count direction;`