

Web Programming (CSci 130)

Department of Computer Science
College of Science and Mathematics
California State University Fresno
H. Cecotti

Learning outcomes

- In this class, you will get more knowledge about:
 - CSS syntax
 - Focus on **attributes** and **properties**
 - CSS examples
- At the end of this class (or end of the week), you must know
 - How to put CSS code in an HTML page
 - Inline, Internal, External
 - Understand the meaning of the **units** and **parameters** in HTML files
 - Have a knowledge about what it is possible to achieve with CSS
 - Practice will come throughout this course with additional elements
 - **NEED to practice in order to remember the commands**

CSS3

■ Learning outcomes

- Syntax of CSS3
- Examples with CSS3

■ Remarks

- It is critical to grasp the concepts of the syntax and the parameters that can be used in order to “guess” what can be achieved.
 - To be in a position where you know what is possible to be done, and to be able to search the right parameters to do
- We will have several sessions with CSS
 - To get used with the syntax, to try multiple examples, to get comfortable with it

Introduction

- CSS (Cascading Style Sheet)
 - Different levels (CSS1, CSS2, CSS3)
 - Profiles
 - Subset of CSS built for a particular device
 - Mobile interface, printers, TV...
 - Media type: CSS2
- Goal:
 - To fill the gap of HTML that is used for the structure of the document
 - Presentation of the document

From CSS1 to CSS3

■ CSS1

- Official W3C recommendation (1996)
- Support for:
 - Font properties (typeface, emphasis, ...)
 - Text color, backgrounds
 - Text attributes
 - Alignment of text, images, tables,...

■ CSS2

- Official W3C recommendation (1998)
 - Super set of CSS1
- New properties
 - Absolute, relative, fixed elements (e.g. background)
 - Media types
 - Aural Style Sheets (for blind, print impaired communities)
 - Bidirectional text, shadows

CSS3

■ CSS3

- Latest evolution of the CSS language
- Extension of CSS 2.1
- Many novelties
 - Rounded corners
 - More shadows
 - Gradients, animations...(many elements to replace predefined images)



The CSS syntax

- Selector
 - { list of properties with values }
 - P { }
- The **id** selector
 - #something { }
- The **class** selector
 - p.myparagraph { background-color: red; }
- Grouping selector
 - Li ul ol { background-color: blue; }
- Comments
 - /* it is a comment */

In the HTML document

- Be organized, structured...
- Several solutions to get the CSS code in the HTML
 - **Inline:** by using the style attribute in HTML elements
 - To avoid !
 - style="color:red;"
 - **Internal:** by using a <style> element in the <head> section
 - **External :** by using an external CSS file in the <head> section
 - <link rel="stylesheet" href="styles.css">
- Remarks
 - Inline takes the **highest** priority when you have several style declaration in the documents.
 - Priority: Inline > Internal > External

The main parameters...

- It can be quite frustrating to place an element exactly the way you want
 - Warning: You may think you do something correct, and it looks as you want, but adding more elements, you may realize it is not so much true
 - We are dealing with (empty) spaces between blocks, so it can be tricky to align blocks.
 - Use the information from the browser (F12) so check where is what.
- You must know (for Midterm, Final,...):
 - Fonts
 - **Border**
 - **Padding**
 - Extra space **inside** the control
 - **Margin**
 - Extra space **around** the control
- CSS3
 - More about knowledge than problem solving
 - Syntax, Type of command to use
 - → Personal work needed to remember the syntax and the command
 - Expect a significant number of questions related to CSS in the midterm: you know or you don't ☹

Class and ID selectors

- When to use **class** selectors and **id** selectors?
 - Use **classes** whenever possible, in large part because you can **reuse** them.
 - Some advocate not using **ids** at all, the choice comes down to you as you develop your sites. In any case, here are two of the
- **id** selectors:
 - Their associated styles can't be reused on other elements
 - an **id** may appear on only one element in a page.
 - This can lead to repeating styles on other elements, rather than **sharing** them via a **class**.
 - They are more specific than **class** selectors.
 - if you ever need to override styling that was defined with an **id** selector, you will need to write a CSS rule that's even more specific.
 - your CSS can get longer and more complicated than necessary on a substantial website.
- Reason some people like using **ids**
 - They will know at a glance if an element is unique.
- Recommendations
 - search for opportunities to combine shared styles into one or more **classes**
 - you can reuse them
 - keep **id** selectors to a minimum if you do use them
 - You may find your style sheets shorter and easier to manage.
 - **Factorize as much as you can!**

```
Class
└── .very {
    color: red;
}

ID
└── #gaudi {
    color: red;
}
```

Units

- Units are critical to understand the different values in the CSS files
 - % : relative to another value
 - International units
 - cm : centimeters, mm: millimeters
 - em: a relative measurement for the height of a font in em spaces
 - em : size of a given font (font = 12pt → em = 12 pt)
 - ex: measurement relative to a font's x-height
 - in: inches
 - pc = pica = 12pt
 - pt : points (1/72 inch) dpi: 72
 - px: pixels
 - Consider the resolution of the screen
 - vh, vw: 1% of viewport (viewport = the browser window)
 - the area where the browser renders the site. screen minus the reserved space of the browser chrome.
 - size an element based on that viewport, like a sidebar.

Units

■ Example with %

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  font-size:12px;
}
div {
  font-size: 200%;
  border: 1px solid black;
}
</style>
</head>
<body>
<p>The fontsize of this paragraph is 12px.</p>
<div>The fontsize of this div element is 200%.</div>
<p>The % unit sets the font-size relative to the current font-size.
The div is inside the body.</p>
</body>
</html>
```

Units

- Viewport width (vw) and viewport height (vh)
 - Viewport-percentage values introduced in CSS3
 - vh unit
 - Used to fill the background of the viewport
 - vw unit
 - Used to fill viewport backgrounds for horizontally-scrolling sites
 - Height property
 - sets the background to the full height of the viewport
 - vh unit with % → 100vh == 100% of the viewport
 - Width property
 - sets the background to the full width of the viewport (%)
 - The container div's width property:
 - value of 200vw
 - → 200% of the viewport width
 - → to contain the 2 floated divs.

```
.viewport-1 {  
    background-size: cover;  
    background-image: url('../mypictures/landscape01.jpg');  
    height: 100vh;  
}  
.viewport-2 {  
    background-size: cover;  
    background-image: url('../mypictures/landscape01.jpg');  
    height: 100vh;  
}
```

```
#container {  
    width: 200vw;  
}  
.viewport-1 {  
    float: left;  
    background-size: cover;  
    background-image: url('../mypictures/landscape01.jpg');  
    width: 100vw;  
    height: 100vh;  
}  
.viewport-2 {  
    float: left;  
    background-size: cover;  
    background-image: url('../mypictures/landscape02.jpg');  
    width: 100vw;  
    height: 100vh;  
}
```

Colors

■ Colors with different formats

➤ Hex Code

- #RRGGBB `p{color:#FF0000;}`

➤ Short Hex Code

- #RGB `p{color:#6A7;}`

➤ RGB % r

- `gb(rrr%,ggg%,bbb%)` `p{color:rgb(50%,50%,50%);}`

➤ RGB Absolute

- `rgb(rrr,ggg,bbb)` `p{color:rgb(0,0,255);}`

➤ keyword

- aqua, black, red, blue. `p{color:teal;}`

			
aqua #00FFFF	black #000000	blue #0000FF	fuchsia #FF00FF
			
gray #808080	green #008000	lime #00FF00	maroon #800000
			
navy #000080	olive #808000	purple #800080	red #FF0000
			
silver #C0C0C0	teal #008080	white #FFFFFF	yellow #FFFF00

Colors

■ CSS color values

Color Value	Description and Notes
<code>rgb(red-value, green-value, blue-value)</code>	RGB (red, green, blue) color model values can be a number from 0 to 255 or a percentage (but not a combination of numbers and percentages) <code>rgb(0, 0, 0)</code> and <code>rgb(0%, 0%, 0%)</code> are black <code>rgb(255, 255, 255)</code> and <code>rgb(100%, 100%, 100%)</code> are white
<code>rgba(red-value, green-value, blue-value, alpha)</code>	RGB color model, plus alpha transparency color values are the same as for RGB syntax the fourth parameter, alpha , is a decimal greater than or equal to 0.0 (full transparency) and less than or equal to 1.0 (full opacity)
<code>hsl(hue-value, saturation-value, lightness-value)</code>	HSL (hue, saturation, lightness) color model the hue value is expressed as the angle of a color circle (a number from 0 to 360); 0 and 360 = red, 120 = green, 240 = blue, with other colors represented in between the saturation value is expressed as a percentage; 0% is gray, and 100% is full saturation of the color the lightness value is expressed as a percentage; 0% is black, 100% is white, and 50% is "normal"
<code>hsla(hue-value, saturation-value, lightness-value, alpha)</code>	HSL color model, plus alpha transparency color values are the same as for HSL syntax the fourth parameter, alpha , is a decimal greater than or equal to 0.0 (full transparency) and less than or equal to 1.0 (full opacity)

Lab time

■ Represent the colors given in the previous slide in an HTML page

➤ Where the table of colors is represented as a table

○ We have

- 4 columns
- 8 rows (4 for colors and 4 for the labels)
- In each cell:
 - A square (div) having a color - centered
 - A name (example: green) - centered
 - A code in hexadecimal - centered

			
aqua #00FFFF	black #000000	blue #0000FF	fuchsia #FF00FF
			
gray #808080	green #008000	lime #00FF00	maroon #800000
			
navy #000080	olive #808000	purple #800080	red #FF0000
			
silver #C0C0C0	teal #008080	white #FFFFFF	yellow #FFFF00

Background

- background-color property
 - to set the background color of an element.
- background-image property
 - to set the background image of an element.
- background-repeat property
 - to control the repetition of an image in the background.
- background-position property
 - to control the position of an image in the background.
- background-attachment property
 - to control the scrolling of an image in the background.
- background property
 - a shorthand to specify a number of other background properties.

Fonts

- The font-family property
 - to change the face of a font.
- The font-style property
 - to make a font italic or oblique.
- The font-variant property
 - to create a small-caps effect.
- The font-weight property
 - to increase or decrease how bold or light a font appears.
- The font-size property
 - to increase or decrease the size of a font.
- The font property
 - a shorthand to specify a number of other font properties.

Text

- color property: to set the color of a text.
- direction property: to set the text direction.
- letter-spacing property: to add or subtract space between the letters that make up a word.
- word-spacing property: to add or subtract space between the words of a sentence.
- text-indent property: to indent the text of a paragraph.
- text-align property: to align the text of a document.
- text-decoration property: to underline, overline, and strikethrough text.
- text-transform property: to capitalize text or convert text to uppercase or lowercase letters.
- white-space property: to control the flow and formatting of text.
- text-shadow property: to set the text shadow around a text.

Images

- **border** property:
 - to set the width of an image border.
- **height** property:
 - to set the height of an image.
- **width** property:
 - to set the width of an image.

Some effects for paragraph

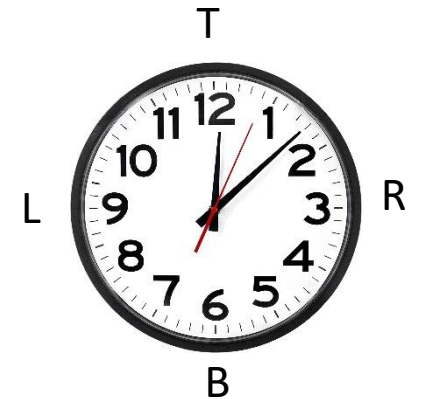
■ p

- Uppercase (everything is uppercase)
- Lowercase (everything is lowercase)
- Capitalize (first letter of each word is Uppercase)

```
div.a {  
    text-transform: uppercase;  
}  
  
div.b {  
    text-transform: lowercase;  
}  
  
div.c {  
    text-transform: capitalize;  
}
```

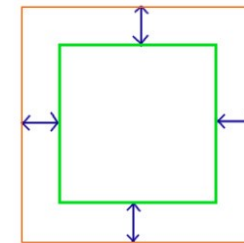
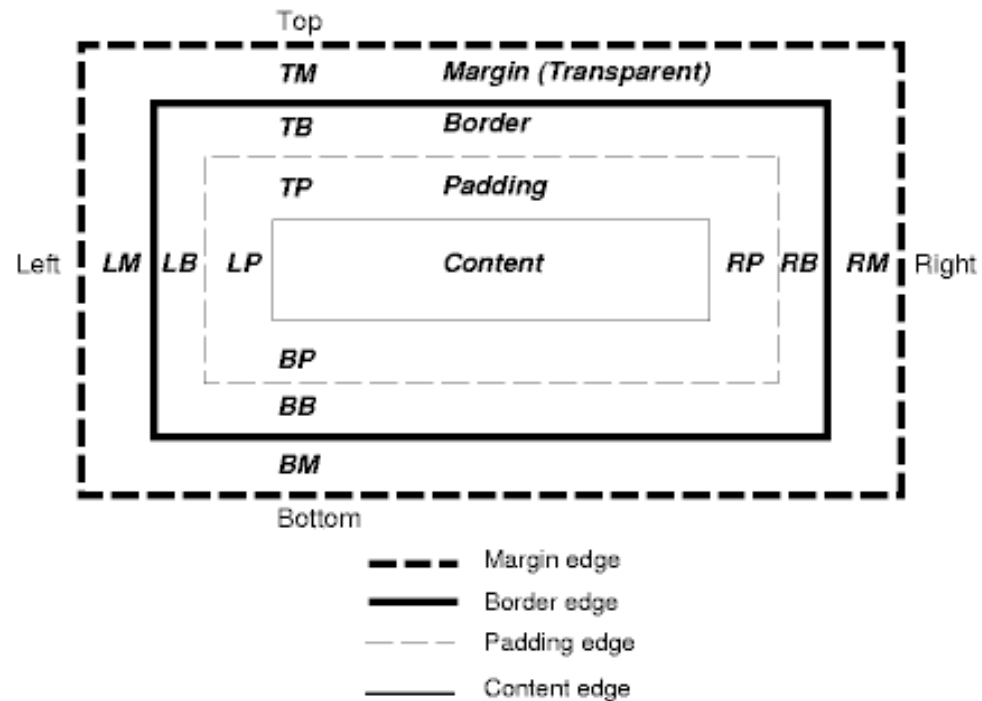
Margins and Paddings

- On all four sides of an element
 - To set all individual **margins** at once:
 - margin-top , margin-right , margin-bottom , and margin-left
 - **TRBL**
 - Need to remember for questions in the midterm 😊
- **Margin:**
 - the space **outside** something
 - how much distance the element wants to keep with other elements around it
- **Padding:**
 - the space **inside** something
 - space between the **content** and the **border**
 - how much distance an element wants to keep with the elements inside it.

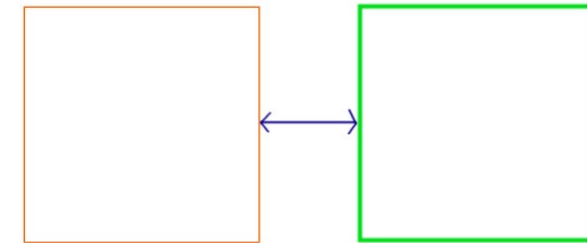


Margins and Paddings

■ Example:



Padding



Margin

Media specific style sheets

- Canvas

- See example in the files....

CSS Properties and Values

- The 8 following slides are given as a **reference**
 - You don't need to know everything
 - Practice and you will remember a good number
 - Some will be asked during the first midterm

CSS Properties and Values (1)

■ Property/Values	Description and Notes
background any combination of the values for background-attachment , background-color , background-image , background-repeat , and/or background-position , or inherit	For changing the background color and image of elements. initial value depends on individual properties; not inherited; percentages allowed for background-position To display multiple background images, separate the combined background values with a comma; if you are specifying a background-color , this should be included as a part of the last background.
background-attachment either scroll , fixed , or inherit	For determining if and how background images should scroll. initial value: scroll ; not inherited If you are displaying multiple background images, you can apply a separate background-attachment value to each by separating the values with commas.
background-color either a color, transparent , or inherit	For setting just the background color of an element. initial value: transparent ; not inherited

CSS Properties and Values (2)

Property/Values	Description and Notes
background-image either a URL, a CSS gradient (see Table B.4), none , or inherit	For setting just the background image of an element. Initial value: none ; not inherited To display multiple background images, separate the image values with a comma.
background-position either one or two percentages or lengths (or one percentage and one length) or one of top , center , or bottom and/or one of left , center , or right , or use inherit	For setting the physical position of a specified background image. Initial value: 0% 0% ; If a single percentage is set, it is used for the horizontal position, and the initial value of the vertical is set to 50% ; If only one keyword is used, the initial value of the other is center ; applies to block-level and replaced elements; not inherited; percentages refer to the size of the box itself If you are displaying multiple background images, you can apply a separate background-position value to each by separating the values with commas.
background-repeat one of repeat , repeat-x , repeat-y , no-repeat , or inherit	For determining how and if background images should be tiled. Initial value: repeat ; not inherited If you are displaying multiple background images, you can apply a separate background-repeat value to each by separating the values with commas.

Property/Values	Description and Notes
background-size either one or two percentages or lengths, or auto , or use cover or contain	For specifying the size of background images. Initial value: auto ; not inherited If you are displaying multiple background images, you can apply a separate background-size value to each by separating the values with commas.
border any combination of the values of border-width , border-style , and/or a color, or inherit	For defining all aspects of a border on all sides of an element. Initial value depends on individual properties; not inherited
border-color from one to four colors, transparent , or inherit	For setting only the color of the border on one or more sides of an element. Initial value: the element's color property; not inherited
border-radius	For giving a box rounded corners. Initial value: 0 ; not inherited
border-top-right-radius , border-bottom-right-radius , border-bottom-left-radius , border-top-left-radius	For setting the border-radius values for only one corner of the box. Initial value: 0 ; not inherited Note: older versions of Firefox use a different syntax for individual corners: -moz-border-radius-topright , -moz-border-radius-bottomright , -moz-border-radius-bottomleft , and -moz-border-radius-topleft .

CSS Properties and Values (3)

Property/Values	Description and Notes
border-spacing either one or two lengths or inherit	For specifying the amount of space between borders in a table. Initial value: 0 ; may be applied only to table elements; Inherited
border-style one to four of the following values: none , dotted , dashed , solid , double , groove , ridge , inset , outset , inherit	For setting only the style of a border on one or more sides of an element. Initial value: none ; not Inherited
border-top , border-right , border-bottom , border-left any combination of a single value each for border-width , border-style , and/or a color, or use inherit	For defining all three border properties at once on only one side of an element. Initial value depends on individual values; not Inherited
border-top-color , border-right-color , border-bottom-color , border-left-color one color or inherit	For defining just the border's color on only one side of an element. Initial value: the value of the color property; not Inherited
border-top-style , border-right-style , border-bottom-style , border-left-style one of none , dotted , dashed , solid , double , groove , ridge , inset , outset , or inherit	For defining just the border's style on only one side of an element. Initial value: none ; not Inherited

Property/Values	Description and Notes
border-top-width , border-right-width , border-bottom-width , border-left-width one of thin , medium , thick , or a length	For defining just the border's width on only one side of an element. Initial value: medium ; not Inherited
border-width one to four of the following values: thin , medium , thick , or a length	For defining the border's width on one or more sides of an element. Initial value: medium ; not Inherited
bottom either a percentage, length, auto , or inherit	For setting the distance that an element should be offset from its parent element's bottom edge. Initial value: auto ; not Inherited; percentages refer to height of containing block
box-shadow optional inset followed by two to four length values, followed by a color	For adding one or more drop shadows to a box. The length values refer (in order) to: position to the right of the box (negative values position to the left), position down from the box (negative values position up), blur radius (negative values are not allowed), and spread distance (negative values cause the shadow to contract). Each subsequent box-shadow value should be separated from its predecessor with a comma. Initial value: none ; Inherited
clear one of none , left , right , both , or inherit	For preventing elements from wrapping around on one or both sides of floated elements. Initial value: none ; may only be applied to block-level elements; not Inherited

CSS Properties and Values (4)

Property/Values	Description and Notes
clip one of auto , rect , or inherit	For displaying only a portion of an element. Initial value: auto ; applies only to absolutely positioned elements
color a color or inherit	For setting the text color of an element. Initial value: parent's color, some colors are set by browser; Inherited
cursor one of auto , crosshair , default , pointer , progress , move , e-resize , ne-resize , nw-resize , n-resize , se-resize , sw-resize , s-resize , w-resize , text , wait , help , a URL, or inherit	For setting the cursor's shape. Initial value: auto ; Inherited
display one of inline , block , inline-block , list-item , run-in , compact , table , inline-table , table-row-group , table-header-group , table-footer-group , table-row , table-column-group , table-column , table-cell , table-caption , ruby , ruby-base , ruby-text , ruby-base-group , ruby-text-group , none , inherit	For determining how and if an element should be displayed. Initial value: usually inline or block ; not Inherited

Property/Values	Description and Notes
float one of left , right , none , inherit	For determining which side of the parent element an element will float to. Initial value: none ; may not be applied to positioned elements or generated content; not inherited
font If desired, any combination of the values for font-style , font-variant , and font-weight followed by the required font-size , an optional value for line-height , and the also-required font-family , or use inherit	For setting at least the font family and size, and optionally the style, variant, weight, and line height of text. Initial value depends on individual properties; Inherited; percentages allowed for values of font-size and line-height ; font-size and font-family are required for the font property to work
font-family one or more quotation mark-enclosed font names followed by an optional generic font name, or use inherit	For choosing the font family for text. Initial value: depends on browser; Inherited
font-size an absolute size, a relative size, a length, a percentage, or inherit	For setting the size of text. Initial value: medium ; the computed value is inherited; percentages refer to parent element's font size
font-style either normal , italic , oblique , or inherit	For making text italic. Initial value: normal ; Inherited
font-variant either normal , small-caps , or inherit	For setting text in small caps. Initial value: normal ; Inherited

Ruby annotations: short runs of text presented alongside base text, primarily used in East Asian typography as a guide for pronunciation or to include other annotations

CSS Properties and Values (5)

Property/Values	Description and Notes
font-weight either normal , bold , bolder , lighter , 100 , 200 , 300 , 400 , 500 , 600 , 700 , 800 , 900 , or inherit	For applying, removing, and adjusting bold formatting. Initial value: normal ; the numeric values are considered keywords and not integers (you can't choose 150, for example); Inherited
height either a length, a percentage, auto , or inherit	For setting the height of an element. Initial value: auto ; may be applied to all elements except non-replaced inline elements, table columns, and column groups; not inherited
left either a length, a percentage, auto , or inherit	For setting the distance that an element should be offset from its parent element's left edge. Initial value: auto ; may only be applied to positioned elements; not inherited; percentages refer to width of containing block
letter-spacing either normal , a length, or inherit	For setting the amount of space between letters. Initial value: normal ; Inherited
line-height either normal , a number, a length, a percentage, or inherit	For setting the amount of space between lines of text. Initial value: normal ; Inherited; percentages refer to the font size of the element itself

Property/Values	Description and Notes
list-style any combination of the values for list-style-type , list-style-position , and/or list-style-image , or use inherit	For setting a list's marker (regular or custom) and its position. Initial value depends on initial values of individual elements; may only be applied to list elements; Inherited
list-style-image either a URL, none , or inherit	For designating a custom marker for a list. Initial value: none ; may only be applied to list elements; overrides list-style-type ; Inherited
list-style-position either inside , outside , or inherit	For determining the position of a list's marker. Initial value: outside ; may only be applied to list elements; Inherited
list-style-type either disc , circle , square , decimal , lower-roman , upper-roman , lower-alpha , upper-alpha , none , or inherit	For setting a list's marker. Initial value: disc ; may only be applied to list elements; not used if list-style-type is valid; Inherited
margin one to four of the following: length , percentage , or auto , or inherit	For setting the amount of space between one or more sides of an element's border and its parent and/or sibling elements. Initial value depends on browser and on value of width ; not inherited; percentages refer to width of containing block
margin-top , margin-right , margin-bottom , margin-left either a length, a percentage, auto , or inherit	For setting the amount of space between only one side of an element's border and its parent and/or sibling elements. Initial value: 0 ; not inherited; percentages refer to width of containing block; the values for margin-right and margin-left may be overridden if sum of width , margin-right , and margin-left are larger than parent element's containing block

CSS Properties and Values (6)

Property/Values	Description and Notes
max-height, max-width either a length, a percentage, none , or inherit	For setting the maximum height and/or width of an element, respectively. Initial value: none ; may not be applied to inline elements or table elements; not inherited; percentages refer to height/width of containing block
min-height, min-width either a length, a percentage, or inherit	For setting the minimum height and/or width of an element, respectively. Initial value: 0 ; may not be applied to inline elements or table elements; not inherited; percentages refer to height/width of containing block
opacity any decimal value from 0.0 (fully transparent) to 1.0 (fully opaque)	For making an element translucent or invisible. Initial value: 1 ; not inherited
orphans either an integer or inherit	For specifying how many lines of an element may appear alone at the bottom of a page. Initial value: 2 ; may only be applied to block-level elements; inherited; only for use with print media
overflow either visible , hidden , scroll , auto , or inherit	For determining where extra content should go if it does not fit in the element's content area. Initial value: visible ; may only be applied to block-level and replaced elements; not inherited

Property/Values	Description and Notes
padding one to four lengths or percentages, or inherit	For specifying the distance between one or more sides of an element's content area and its border. Initial value depends on browser; not inherited; percentages refer to width of containing block
padding-top, padding-right, padding-bottom, padding-left either a length, a percentage, or inherit	For specifying the distance between one side of an element's content area and its border. Initial value: 0 ; not inherited; percentages refer to width of containing block
page-break-after, page-break-before either always , avoid , auto , right , left , or inherit	For specifying when page breaks should or should not occur. Initial value: auto ; may only be applied to block-level elements; not inherited; only for use with print media
page-break-inside either avoid , auto , or inherit	For keeping page breaks from dividing an element across pages. Initial value: auto ; may only be applied to block-level elements; inherited; only for use with print media
position either static , relative , absolute , fixed , or inherit	For determining how an element should be positioned with respect to the document's flow. Initial value: static ; not inherited
right either a length, a percentage, auto , or inherit	For setting the distance that an element should be offset from its parent element's right edge. Initial value: auto ; may only be applied to positioned elements; not inherited; percentages refer to width of containing block

CSS Properties and Values (7)

Property/Values	Description and Notes
table-layout one of fixed , auto , or inherit	For choosing the algorithm that should be used to determine the widths of cells. Initial value: auto ; not inherited
text-align one of left , right , center , justify , a string, or inherit	For aligning text. Initial value depends on browser and writing direction; may only be applied to block-level elements; inherited
text-decoration any combination of underline , overline , line-through , and blink , or none or inherit	For decorating text (mostly with lines). Initial value: none ; not inherited
text-indent either a length, a percentage, or inherit	For setting the amount of space the first line of a paragraph should be indented. Initial value: 0 ; may only be applied to block-level elements; inherited; percentages refer to width of containing block
text-overflow one of clip , ellipsis , or "string"	For specifying how text overflow must be handled when it is not visible. Initial value: clip
text-shadow two to four length values, followed by a color	For adding one or more drop shadows to the text of an element. The length values refer (in order) to: position to the right of the text (negative values position to the left), position down from the box (negative values position up), blur radius (negative values are not allowed), and spread distance (negative values cause the shadow to contract). Each subsequent text-shadow value should be separated from its predecessor with a comma. Initial value: none ; inherited

Property/Values	Description and Notes
text-transform either capitalize , uppercase , lowercase , none , or inherit	For setting the capitalization of an element's text. Initial value: none ; inherited
transform none or a list of transform functions (matrix , translate , translateX , translateY , scale , scaleX , scaleY , rotate , skew , skewX , skewY)	For transforming the shape, size, or orientation of an element. Initial value: none ; not inherited; transform functions are applied in the same order they are listed
transform-origin either one or two percentages or lengths (or one percentage and one length) or one of top , center , or bottom and/or one of left , center , or right	For defining the origin of any transforms applied to an element. Initial value: 50% 50% ; not inherited; only applies to block-level and inline-level elements; percentages refer to the size of the element's box
transition a space-separated shorthand for defining (in order) transition-property , transition-duration , transition-timing-function , and transition-delay	For defining a transition effect on an element. Initial value depends on the individual property; applies to all elements, including the :before and :after pseudo-elements; the order of the values is important to this property

CSS Properties and Values (8)

Property/Values	Description and Notes
transition-property none, all, or a comma-separated list of CSS properties	For identifying the CSS properties defined on an element that should have a transition applied to it. Initial value: all ; not inherited; applies to all elements, including the :before and :after pseudo-elements
transition-duration a time value in seconds or milliseconds	For defining the time that a transition takes to complete. Initial value: 0s (zero seconds); not inherited; applies to all elements including the :before and :after pseudo-elements
transition-timing-function ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier(number, number, number, number)	For describing how the intermediate values used during a transition are to be calculated. Initial value: ease ; applies to all elements, including the :before and :after pseudo-elements
transition-delay a time value in seconds or milliseconds	For defining when a transition will start. Initial value: 0s (zero seconds); not inherited; applies to all elements, including the :before and :after pseudo-elements
top either a length, a percentage, auto , or inherit	For setting the distance that an element should be offset from its parent element's top edge. Initial value: auto ; may only be applied to positioned elements; not inherited; percentages refer to height of containing block
vertical-align either baseline , sub , super , top , text-top , middle , bottom , text-bottom , a percentage, a length, or inherit	For aligning elements vertically. Initial value: baseline ; may only be applied to inline-level and table cell elements; not inherited; percentages refer to the element's line-height property

Property/Values	Description and Notes
visibility either visible , hidden , collapse , or inherit	For hiding elements without taking them out of the document's flow. Initial value: inherit , which rather makes the fact that it's not inherited a moot point
white-space either normal , pre , nowrap , pre-wrap , pre-lined , or inherit	For specifying how white space should be treated. Initial value: normal ; may only be applied to block-level elements; inherited
widows either an integer or inherit	For specifying how many lines of an element may appear alone at the top of a page. Initial value: 2 ; may only be applied to block-level elements; inherited; only for use with print media
width either a length, a percentage, auto , or inherit	For setting the width of an element. Initial value: auto ; may not be applied to inline elements, table rows, or row groups; not inherited; percentages refer to width of containing block
word-spacing either normal , a length, or inherit	For setting the distance between words. Initial value: normal ; inherited
z-index either auto , an integer, or inherit	For setting the depth of an element with respect to overlapping elements. Initial value: auto ; may only be applied to positioned elements; not inherited

Conclusion

- CSS
 - Powerful approach to adapt the presentation of an HTML file
- Many professional developers have their own blog and share information about their HTML/CSS tricks
 - You must browse the net to get **new ideas** about presentations
- Reading
 - Book: HTML5 and CSS3, 7th Edition (or more recent edition)
 - Quiz to try
 - https://www.w3schools.com/css/css_quiz.asp
 - https://www.tutorialspoint.com/css/css_online_quiz.htm(maybe the first midterm will a few questions from these tests...)
- Next session
 - Focus on
 - **Selectors**
 - **Events**