Chapter 3
Cascading Style Sheets (CSS)

INT150 Web Technologies (2/2024)

Sanit Sirisawatvatana

Topics

- What will you learn after finishing this class:
    - Cascading Style Sheets (CSS)
    - The benefits of CSS
    - How to use the Style Sheets
    - Concepts of CSS
    - Style Sheet Hierarchy
    - CSS for formatting Text

Cascading Style Sheets (CSS)

- Cascading Style Sheets (CSS)
    - is the W3C standard for defining the presentation of documents written in HTML and any XML language.
- Presentation refers to the way the document is displayed or delivered to the user on:
    - a computer screen
    - a cell phone display,
    - printed on paper
    - read aloud by a screen reader
- CSS is a separate language with its own syntax.

The benefits of CSS

- Precise type and layout controls
    - can achieve print-like precision using CSS
- Less work
    - one style sheet can change the appearance of an entire site
- More accessible sites
    - can mark up the content meaningfully, making it more accessible for non-visual or mobile devices.
- Reliable browser support
    - Every browser supports CSS Level 2 and many cool parts of CSS Level 3

How Style Sheets Work

1. Start with a HTML document

2. Write style rules how certain elements should look.

3. Attach the style rules to the HTML document. When the browser display the HTML document, it follows the style rules for rendering elements.

CSS Rule sets

- A style sheet is made up of one or more style instructions (called rules or rule sets) that describe how an element or group of elements should be displayed.

- Each rule *selects* elements and *describes* how it should look.

- Examples

```
h1 { color: green; }

p   { font-size: small; font-family: sans-serif; }
```

CSS Rule Set

- Syntax

declaration

selector { property: value; }

selector { property1 : value1 ;
property2 : value2 ;
property3 : value3 ;
} declaration block

selector { property1: value1; property2: value2; property3: value3; }

- selector identifies the element or elements to be affected.
- declaration provides the rendering instructions. It is made up of a property (such as color) and its value (green), separated by a colon (:) and a space.

  If the semicolon is omitted, the declaration and the one following it will be ignored.

CSS Rule Set

- Declaration block contains the curly brackets and one or more declarations separated by semicolon (;).

- CSS ignores whitespace and line returns within the declaration block.

- Each declaration should be written in its own line. This makes it easier to find the properties applied to the selector and to tell when the style rule ends.

- Values are dependent on the property. When using a property, it is important to know which values it accepts.

Comments in Style Sheets

- **Syntax:**

  /* comment goes here */

- Sometime it is helpful to leave comments in a style sheet.

- Content between the /* and */ will be ignored when the style sheet is parsed (even within a rule).

```
body {
        font-size: small;
        /* font-size: large;  */
}
```

- One use for comments is to label sections of style sheet to make things easier to find later.

  /* Footer Styles */

- CSS comments are also useful for temporarily hiding style declaration in the design process.

Attaching the styles to HTML document

Three ways to apply the style information to an HTML document:

1. Inline style sheets - by using the style attribute in HTML documents

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

2. Internal style sheets (or Embedded style sheets) - by using a <style> element in the <head> section.

```
<head>
  <title>Title Go Her</title>
  <style>
    h1 { color: blue; }
  </style>
</head>
```

3. External style sheets - by using external CSS file. To use an external style sheet, add a link to it in the <head> section of the HTML page. In this way, all the files in a website may share the same style sheet.

```
<head>
  <title>Title Go Her</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

https://www.w3schools.com/tags/tag_link.asp

Concepts of CSS

- Fundamental concepts of CSS that control how CSS is applied to HTML and how conflicts are resolved.
  - Cascading (Last Rule)
  - Specificity
  - Inheritance

https://www.w3.org/TR/CSS21/cascade

https://dev.to/frontenddude/important-css-concepts-to-learn-57j3

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance

Cascade

- Style sheets cascade
  - The order of CSS rules is important. When two rules apply that have equal specificity the one that comes last in CSS is the one that will be used.

```
h1 {
   color: red;
}
h1 {
   color: blue;
}
```

<h1>This is my heading.</h1>

This is my heading.

Demo

Specificity

- Specificity
    - is how the browser decides which rule applies if multiple rules have different selectors, but could still apply to the same element.
- Basic measure of how specific a selector's selection will be:
    - An element selector is less specific (or lower score)
    - A class selector is more specific (or higher score)
    - The higher score, the rule is applied to.

```
.main-heading {
    color: red;
}
h1 {
    color: blue;
}
```

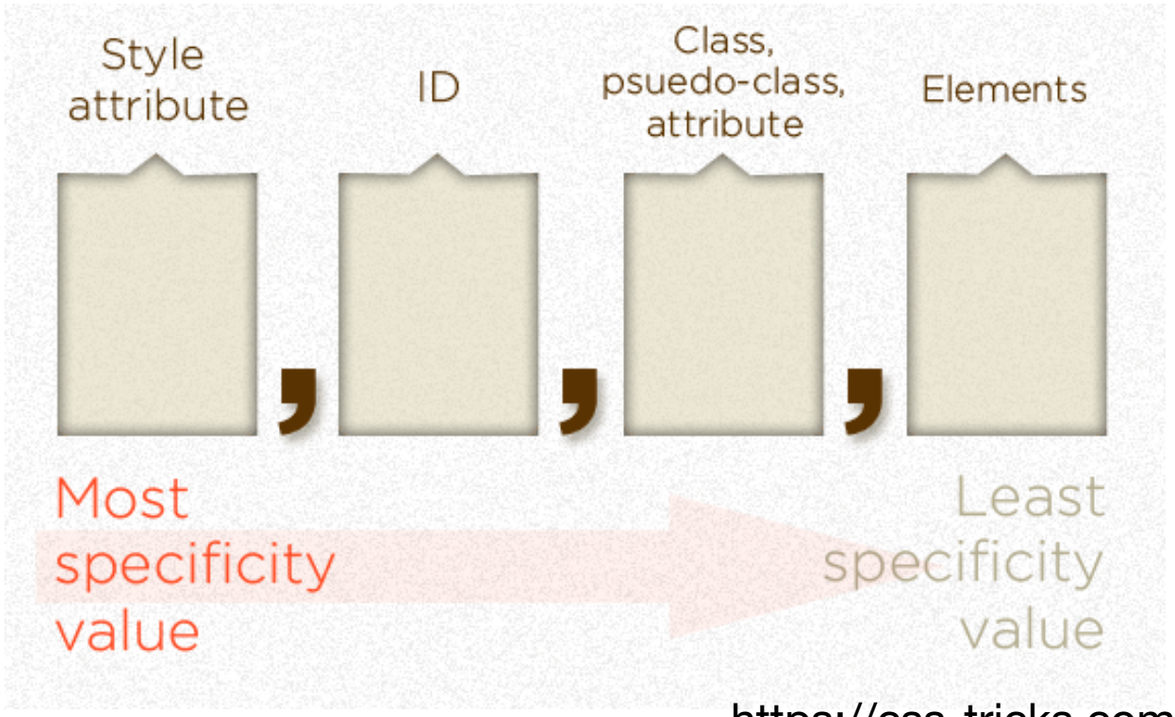
This is my heading.

```
<h1 class="main-heading">This is my heading.</h1>
```

Specificity Ranking
1. Inline style (highest)
2. IDs
3. Class and Pseudo Class's
4. Element selectors (lowest)

Demo

CSS Specificity Value



https://css-tricks.com/specifics-on-css-specificity/

- If the element has inline styling, that automatically1 wins (1,0,0,0 points)
- For each ID value, apply 0,1,0,0 points
- For each class value (or pseudo-class or attribute selector), apply 0,0,1,0 points
- For each element reference, apply 0,0,0,1 point

https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity

Inheritance

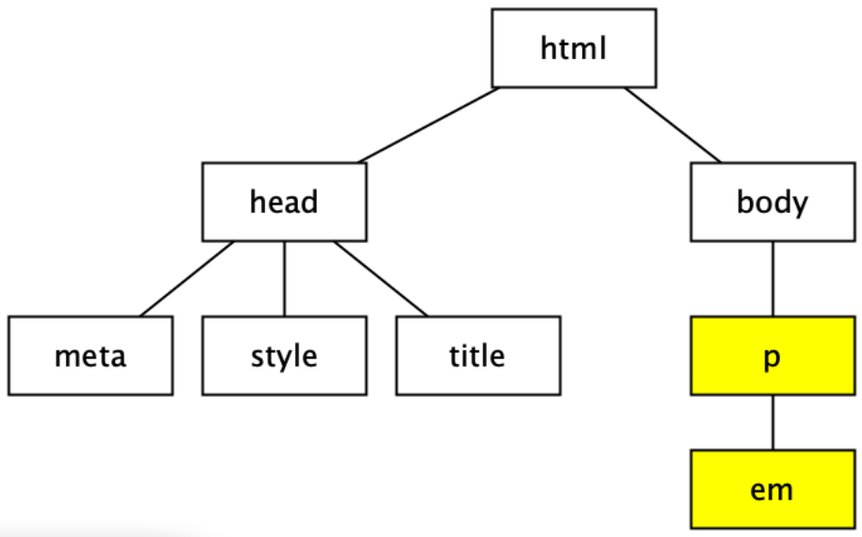
- Inheritance is the mechanism by which certain CSS properties are automatically passed down from a parent element to its child elements.
    - Like parents pass down traits to their children such as eye color or hair color.

- This helps maintain consistency and reduces the need to repeatedly define styles for nested elements.

Inheritance

- Inheritance
  - Some CSS property values set on parent elements are inherited by their child elements, some aren't.

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="utf-8" />
                <title>HTMLiveCode</title>
                <style type="text/css">
                  p {
                      color: blue;
                  }
                </style>
        </head>
        <body>
                <p>This is <em>my</em> paragraph.</p>
        </body>
</html>
```



This is *my* paragraph.

Demo

Important Note

- Some style sheet properties inherit and others do not.

- In general, properties related to the styling of text -- font size, color, style, etc. -- are passed down (inherited).

- Properties such as borders, margins, backgrounds and so on, that affect the box area around the element tend not to be passed down.

- Any property applied to a specific element will override the inherited values for that property.

Assigning Importance

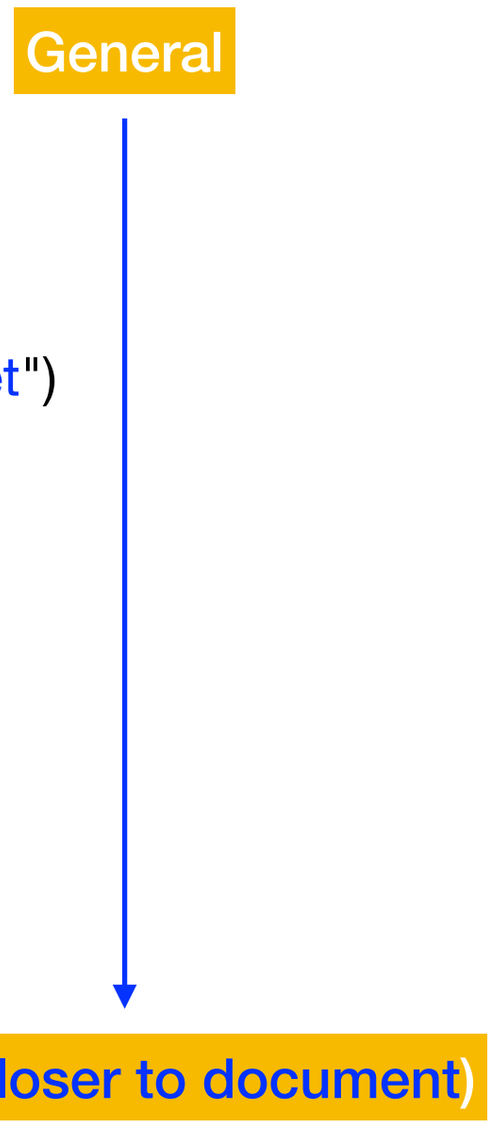
- To prevent a specific rule from being overridden, include the !important indicator just after the property value and before semicolon for the rule.

```
p { color: blue !important ; }
```

Style Sheet Hierarchy

- Style information can come from various sources, listed here from general to specific. Items lower in the list will override items above them:

1. Browser default settings (as "user agent style sheet")
2. User style settings (set in a browser as "reader style sheet")
3. Linked external style sheet (with link element as "author style sheet")
4. Imported style sheet (with @import function)
5. Embedded style sheets (with style element)
6. Inline style information (with style attribute in opening tag)
7. Any style rule marked !important by the author
8. Any style rule marked !important by the reader (user)



5 override 3

```
/* css/mystyle.css. */
p {
        color: blue;
}
```

3. Linked external style sheet

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="utf-8" />
                <title>HTMLiveCode</title>
                <link rel="stylesheet" href="css/mystyle.css">
                <style type="text/css">
                        p {
                                color: red;
                        }
                </style>
        </head>

        <body>
                <p>This is paragraph.</p>
        </body>
</html>
```

5. Embedded style sheets (with style element)

This is paragraph.

7 override 3

```
/* css/mystyle.css. */
p {
        color: blue !important ;
}
```

7. Any style rule marked !important by the author

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="utf-8" />
                <title>HTMLiveCode</title>
                <link rel="stylesheet" href="css/mystyle.css">
                <style type="text/css">
                        p {
                                color: red;
                        }
                </style>
        </head>

        <body>
                <p>This is paragraph.</p>
        </body>
</html>
```

3. Linked external style sheet

This is paragraph.

6 override 5

```
/* css/mystyle.css. */
p {
        color: blue;
}
```

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="utf-8" />
                <title>HTMLiveCode</title>
                <link rel="stylesheet" href="css/mystyle.css">
                <style type="text/css">
                        p {
                                color: red;
                        }
                </style>
        </head>

        <body>
                <p style="color: purple;">This is paragraph.</p>
        </body>
</html>
```

5. Embedded style sheets (with style element)

6. Inline style information

This is paragraph.

7 override 6

```
/* css/mystyle.css. */
p {
        color: blue;
}
```

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="utf-8" />
                <title>HTMLiveCode</title>
                <link rel="stylesheet" href="css/mystyle.css">
                <style type="text/css">
                        p {
                                color: red !important ;
                        }
                </style>
        </head>

        <body>
                <p style="color: purple;">This is paragraph.</p>
        </body>
</html>
```

7. Any style rule marked !important by the author

6. Inline style information

This is paragraph.

Grouped selectors

- Grouped selectors
  - allows one rule with the same style properties to a number of elements by separating them with commas.
  - makes future edits more efficient and results in a smaller file size.

    h1, h2, p, em { font-family: sans-serif; color: blue; }

CSS: Formatting Text

- Several properties to control the appearance of text in HTML documents.
  - font-related properties
    - font-family
    - font-size
    - font-weight
    - font-style
    - font-variant
    - font
  - text properties
    - color
    - line-height
    - text-indent
    - text-align
    - text-decoration
    - text-transform
    - letter-spacing
    - word-spacing
    - text-shadow

font-family

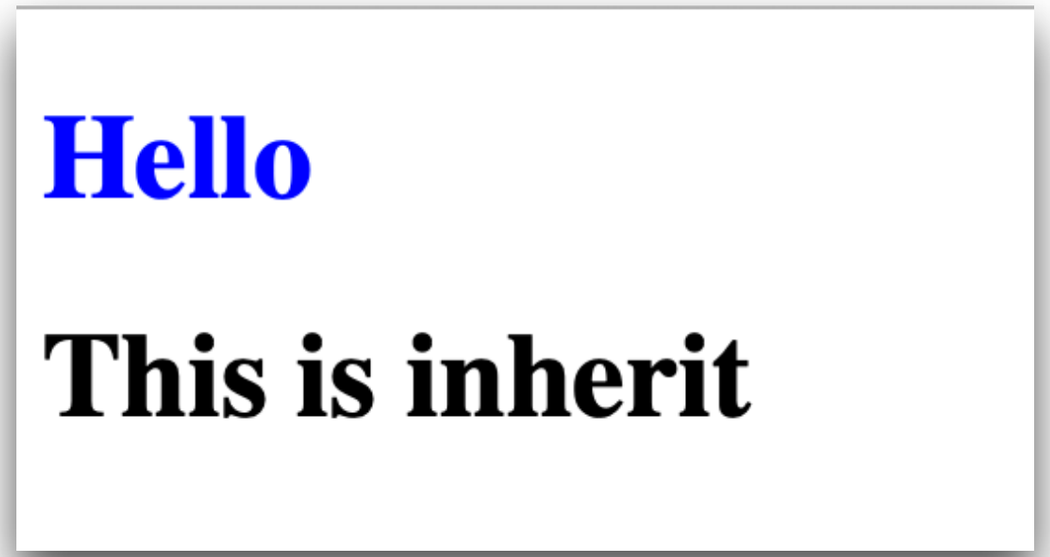
Values:	one or more font or generic font family names, separated by comma | inherit
Default:	depends on the browser
Applies to:	all elements
Inherits:	yes

```
body { font-family: Arial; }

tt     { font-family: Courier, monospace; }

p      { font-family: “Duru Sans”, Verdana, sans-serif; }
```

https://tympanus.net/codrops/css_reference/inherit/

Value: inherit

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>HTMLiveCode</title>
        <style>
            h1 {
                color: blue;
            }

            .extra h1{
                color: inherit;
            }
        </style>
    </head>
    <body>
        <h1>Hello</h1>
        <div class="extra">
            <h1>This is inherit</h1>
        </div>
    </body>
</html>
```



http://web.sit.kmutt.ac.th/sanit/int102/demo/demo_inherit.html

Syntax notes

- All font names, with the exception of generic font families, must be capitalized.

- Use commas to separate multiple font names.

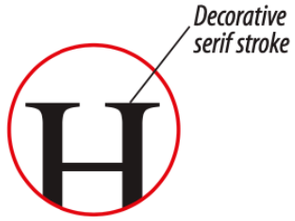
- Font names that contain a character space must appear within quotation marks. (such as "Duru Sans")

- The font-family property should hold several font names as a list of "back-up" fonts. If the browser does not support the first font, it tries the next font, and so on.

- Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

Generic font families

Serif



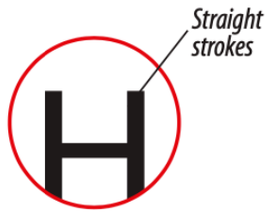
Hello
Times

Hello
Georgia

Hello
Times New Roman

Hello
Lucida (Mac)

Sans-serif



Hello
Veranda

Hello
Trebuchet MS

Hello
Arial

Hello
Arial Black

Monospace

Monospace font (equal widths)

Proportional font (different widths)

Hello
Courier

Hello
Courier New

Hello
Andale Mono

Cursive

Hello
Apple Chancery

Hello
Comic Sans

Hello
Snell

Fantasy

Hello
Imapct

HELLO
Stencil

HELLO
Mojo

font-size

Values:	length unit | percentage | xx-small | small | medium | large | x-large | xx-large | smaller | larger | inherit
Default:	medium
Applies to:	all elements
Inherits:	yes

```
h1 { font-size: 1.5em; }  /* length unit */
h1 { font-size: 150%; }   /* percentage value */
h1 { font-size: x-large; } /* absolute keywords */
strong { font-size: larger }  /* relative keyword (larger or smaller) */
```

font-size

- The font-size value can be an absolute or relative size
- Absolute size
  - set the text to a specified size
  - does not allow a user to change the text size in all browsers
  - absolute size is useful when physical size of the output is known or for print (All of absolute units are outdate)
- Relative size
  - is the preferred value in contemporary web design
  - Set the size relative to surrounding elements
  - allows a user to change the text size in browsers

https://www.w3schools.com/css/css_font_size.asp

CSS Units of Measurement

- CSS provides a variety of units of measurement:

Relative Units	
~~px~~	~~pixel (CSS 2.1) because it varies with display resolution~~
em	a unit of measurement equal to the current font size
ex	x-height; approximately the height of a lowercase "x" in the font
rem	root em (CSS 3), equal to the em size of root element(html)
Absolute Units	
px	pixel equal 1/96 of an inch (CSS 3)
pt	points (1/72 inch in CSS 2.1)
pc	picas (1 pica = 12 points)
mm	milimeters
cm	centimeters
in	inches

https://snook.ca/archives/html_and_css/font-size-with-rem

set font-size with em

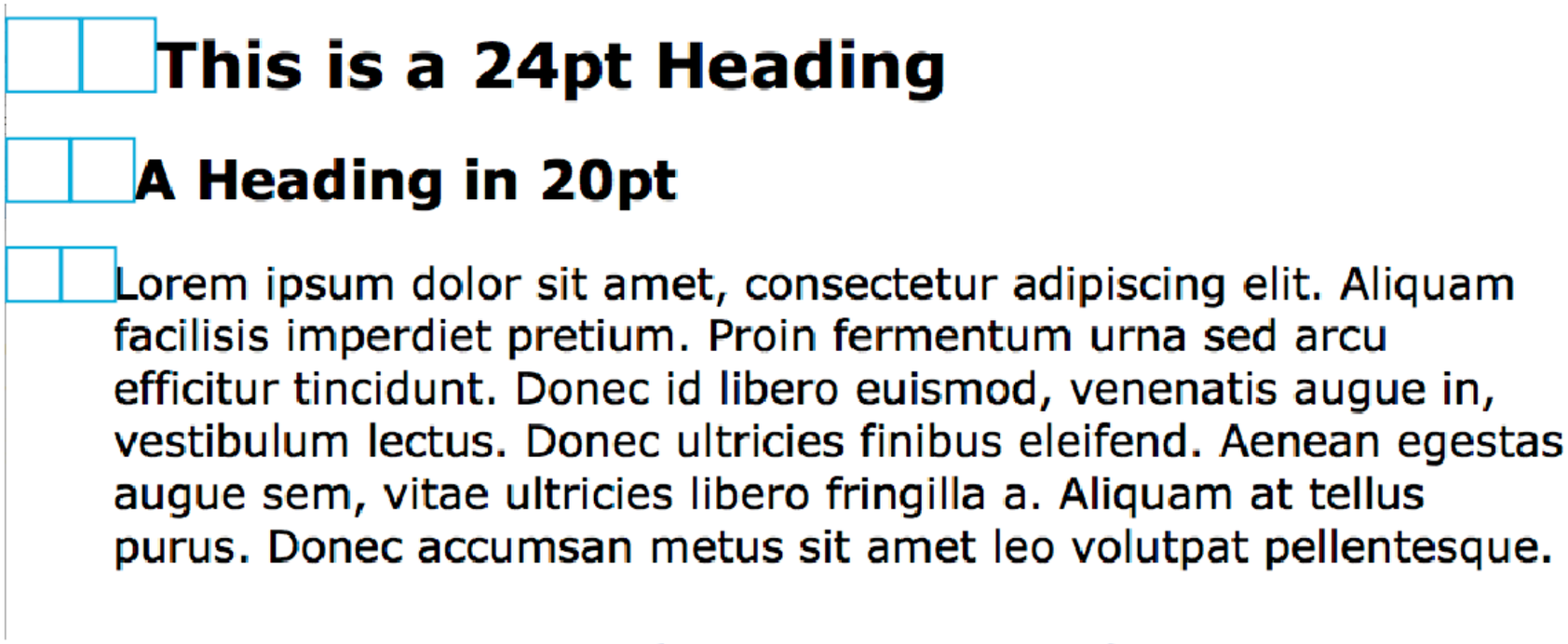
- To allow users to resize the text in the browser menu, many developers use em instead of pixels.
- The em size unit is recommended by the W3C.
- 1em is equal to the current font size. The default text size in browser is 16px. so 1em is 16px.
- The em value works like a scaling factor, similar to a percentage.
- The em unit on the font-size property is relative to the parent element's font size (unlike all other properties, where they're relative to the font size on the element).

```
body { font-size: 100%; }        /* assume the default of 16px */

h1     { font-size: 2.5em }      /* 2.5x16p = 40px */

h2     { font-size: 1.875em }    /* 1.875x16p = 30px */

p      { font-size: 0.875em }    /* 0.875x16p = 14px */
```

Em used on other properties than font-size

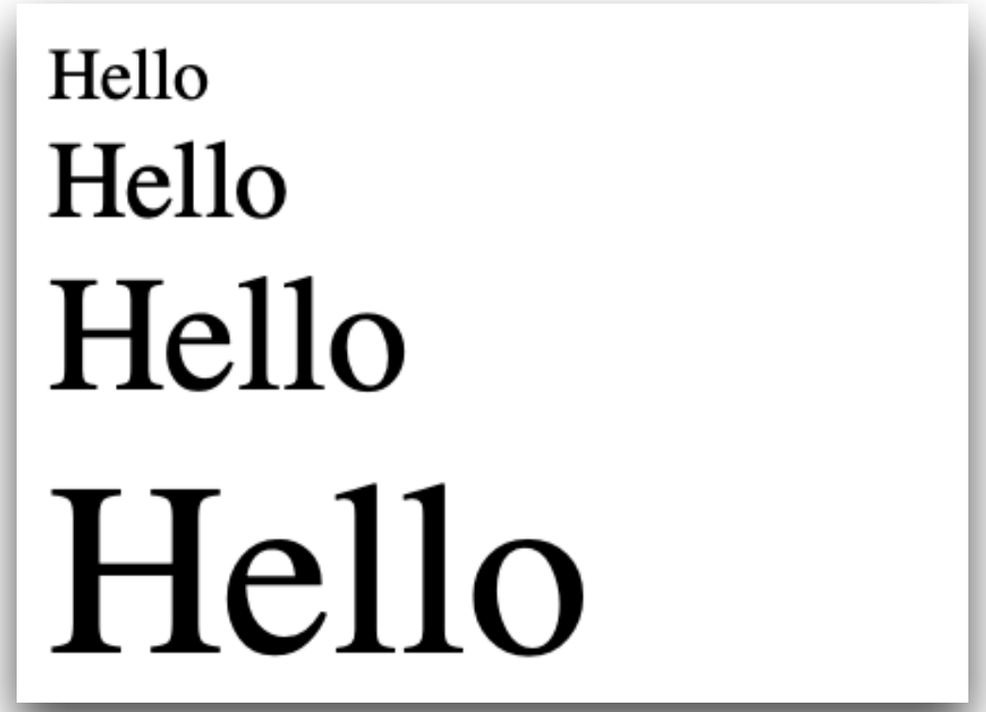
- When em units are used on other properties than font-size, the value is relative to the element's own font-size.

- Em measurements are always relevant to the element's font size. An em for one element may not be the same for another.



set font-size with em

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>HTMLiveCode</title>
        <style>
            div { font-size: 1.5em ;
            }
        </style>
    </head>
    <body>
        <span>Hello</span>
        <div>Hello</div>
        <div><div>Hello</div></div>
        <div><div><div>Hello</div></div></div>
    </body>
</html>
```



https://web.sit.kmutt.ac.th/sanit/int102/demo/demo_fontsize.html

font-weight (boldness)

Values:	normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit
Default:	normal (equivalent of 400)
Applies to:	all elements
Inherits:	yes

```
body { font-weight: normal; }
p    { font-weight: bold; }
p    { font-weight: 900; }
```

https://htmldog.com/references/css/properties/font-weight/

font-style (italics)

Values:	normal | italic | oblique | inherit
Default:	normal
Applies to:	all elements
Inherits:	yes

```
p    { font-style: normal; }
p    { font-style: italic; }
p    { font-style: oblique; }
```

https://en.wikipedia.org/wiki/Oblique_type

font-variant (small caps)

Values:	normal | small-caps | inherit
Default:	normal
Applies to:	all elements
Inherits:	yes

```
p.normal { font-variant: normal;}
p.small  { font-variant: small-caps; }
```

font-variant (small caps)

```
<!DOCTYPE html>
<html>
	<head>
		<meta charset="utf-8" />
		<title>HTMLiveCode</title>
		<style type="text/css">
		p.normal {
			font-variant: normal;
		}

		p.small {
			font-variant: small-caps;
		}
		</style>
	</head>

	<body>
		<p class="normal">My name is Hege Refsnes.</p>
		<p class="small">My name is Hege Refsnes.</p>
	</body>
</html>
```

My name is Hege Refsnes.

My name is Hege Refsnes.

Demo

font (shortcut font)

- CSS provides the shorthand font property that compiles all the font-related properties into one rule.
- At minimum, the font property must include a font-size value and a font-family value, in that order.
- Omitting one or putting them in the wrong order causes the entire rule to be invalid.
- Once, the font property met the size and family requirements, the other values are optional and may appear in any order prior to the font-size.
- When style weight, or variant are omitted, they revert back to normal.
- The value appearing just after font-size, separated by a slash is line-height of the text line.

font (shortcut font)

Values:	font-style font-weight font-variant font-size/line-height font-family | inherit
Default:	depends on default value for each property listed
Applies to:	all elements
Inherits:	yes

```
h3    { font: oblique  bold  small-caps 1.5em/1.8em Verdana,
          sans-serif; }
p     { font: 1em sans-serif; } /* minimum properties */
```

Changing Text color: color

Values:	color value (name or numeric) | inherit
Default:	depeds on the browser and user's preferences
Applies to:	all elements
Inherits:	yes

```
h1    { color: gray; }
h1    { color: #666666; }
h1    { color: #666; }
h1    { color: rgb(102,102,102); }
```

Color Names

- CSS2.1 define 17 standard color names:

black	white	purple
lime	navy	aqua
silver	maroon	fuchsia
olive	blue	orange
gray	red	green
yellow	teal	

- The updated CSS3 color module allows name from a larger set of 140 color name to be specified in style sheets.

color

- To change the color of all the text in a document by applying the color property to the body element.

  body { color: fuchsia; }

- To apply properties to several elements at once, use grouped selectors:

  p, ul, td, th { color: navy; }

- To apply a rule to a particular paragraph or paragraphs, use the following three sectors:

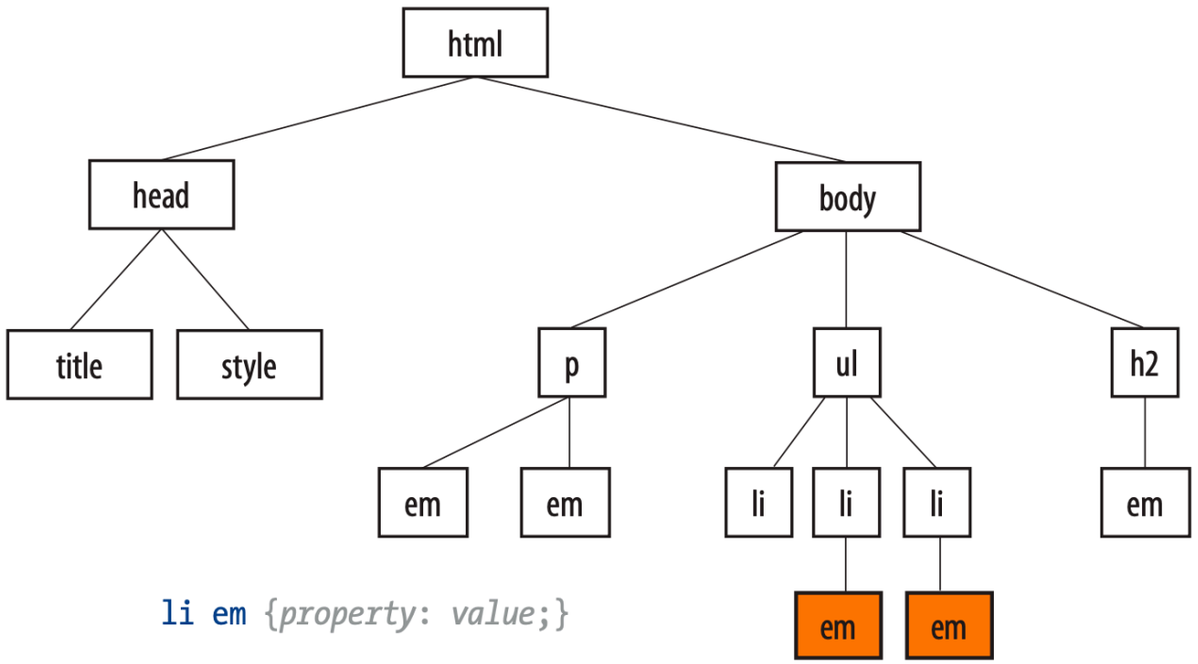
  - Descendant selectors,
  - ID selectors
  - class selectors

Descendant selectors

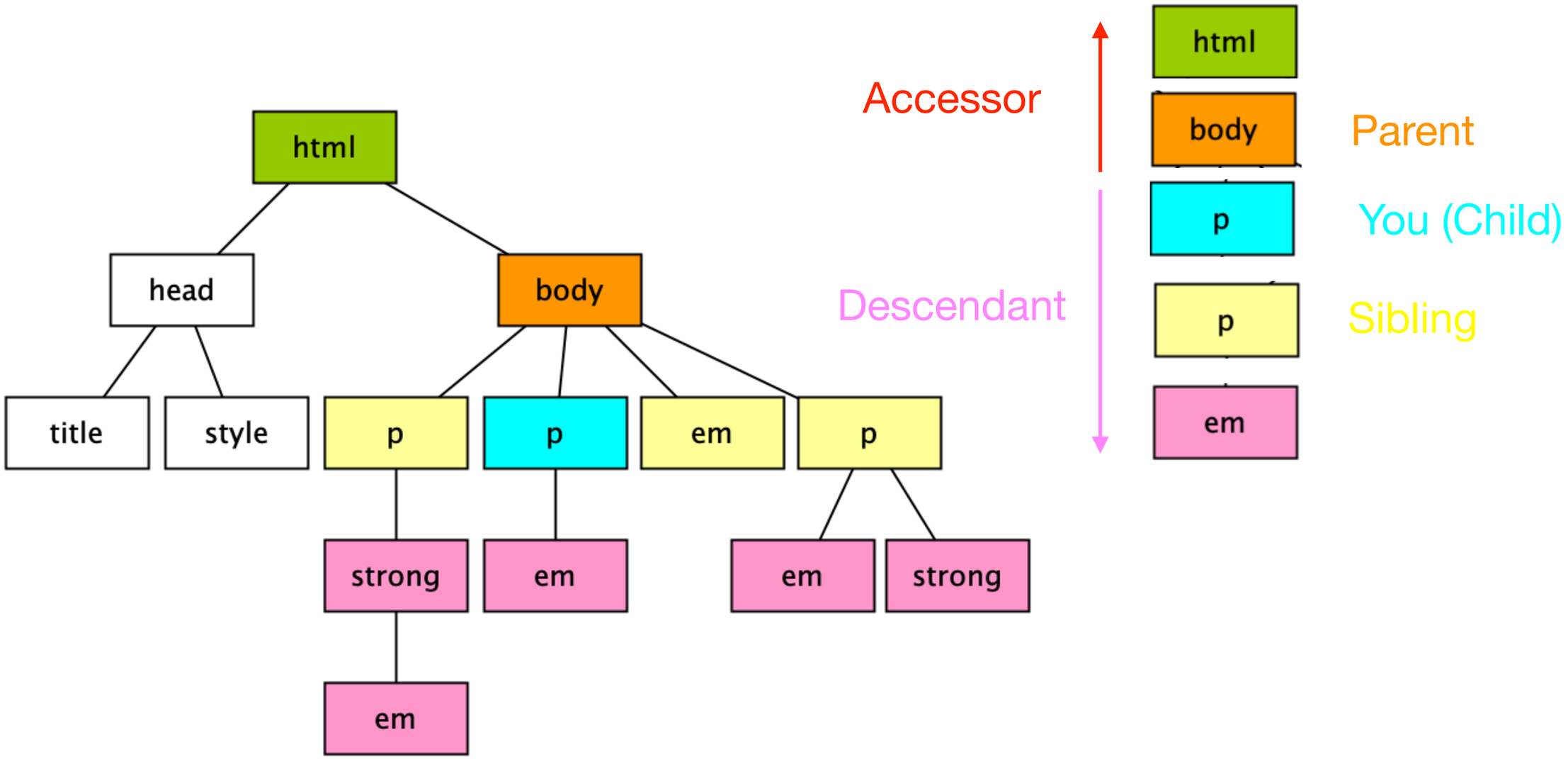
- A descendant selector targets elements that are contained within (and therefore are descendants of) another element.
- Descendant selectors are indicated in a list separated by a character space.

  li em { color: olive; }

  Only em elements within li element are selected. The other em elements are unaffected.



Relationship



Descendant selectors

- Descendant selectors can be grouped in a comma-separated list.

```
h1 em, h2 em, h3 em { color: red; }
```

- Descendant selectors can be nested.

```
ol a em { font-variant: small-caps; }
```

Contextual Sectors

- Contextual selector selects the element based on its context or relation to another element.
- Four types of contextual selectors (called combinators in CSS3 specification)
  - Descendant selectors (character space,' ')

    li em { color: olive; }
  - Child selectors (greater-than symbol,>)

    p > em { font-weight: bold; }
  - Adjacent/Next sibling selectors (plus sign,+)

    h1 + p { font-style: italic; }
  - General/subsequent sibling selectors (~) (New in CSS3)

    h1 ~ h2 { font-weight: normal; }

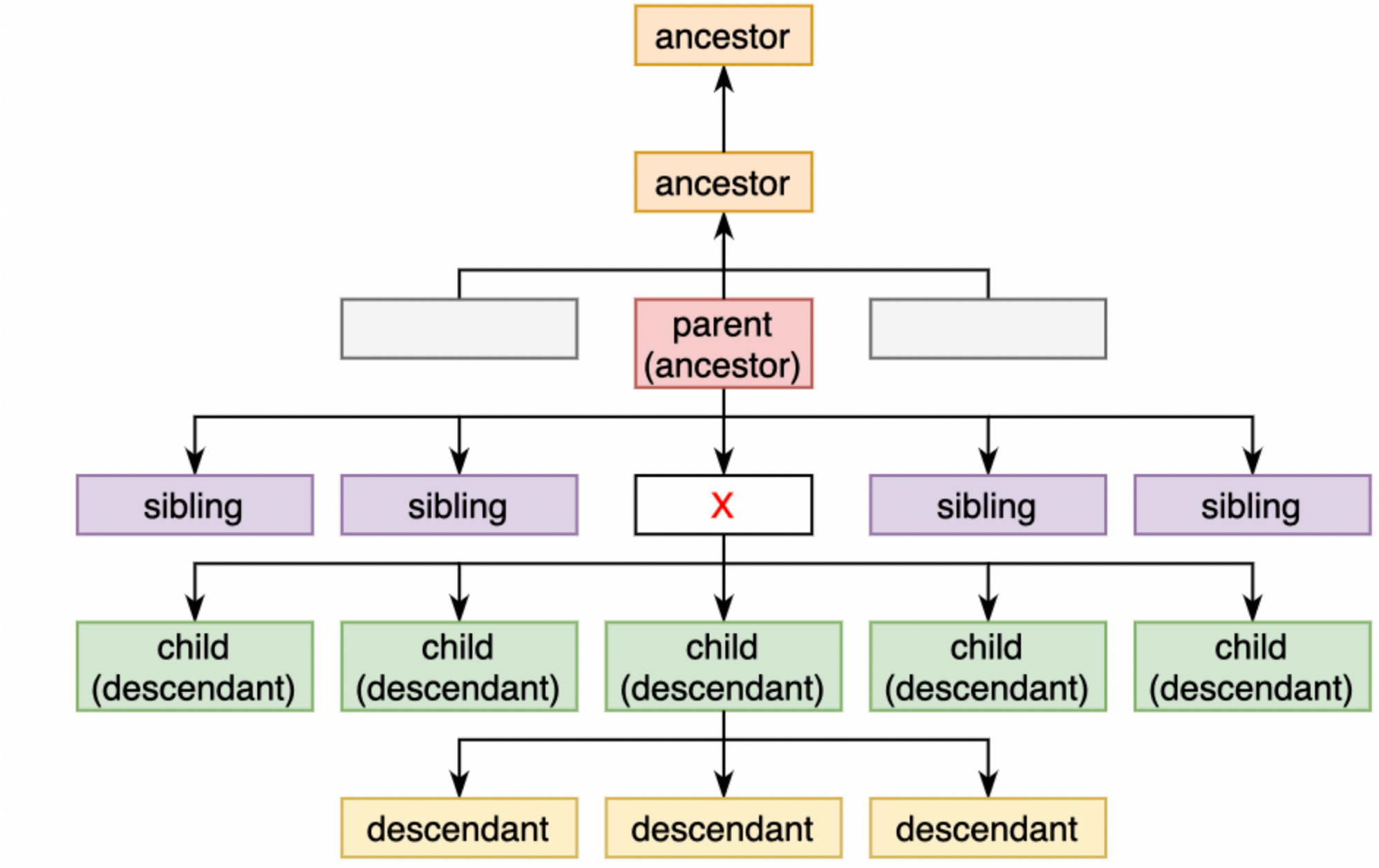
https://www.w3schools.com/css/css_combinators.asp

Contextual Sectors

Selector	Example	Example description
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects all <p> elements that are placed immediately after elements
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element

https://itf-web-advanced.netlify.app/html_css/selectors.html#combinator-selectors

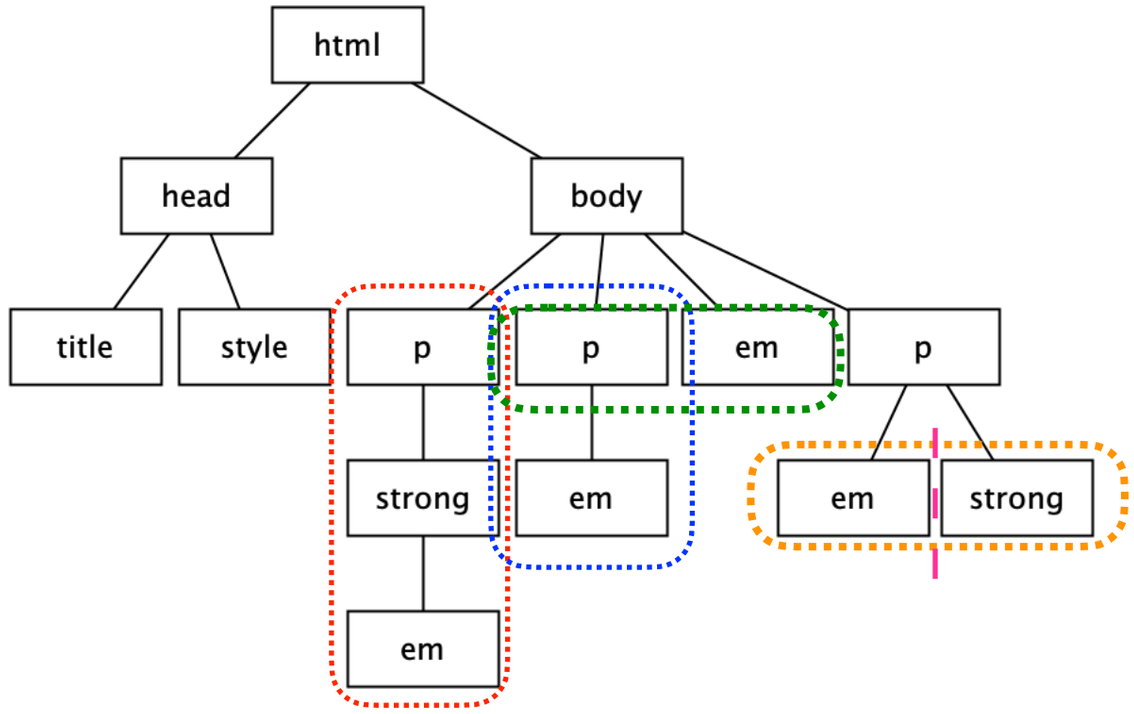
Contextual Sectors



Contextual Sectors

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>Contextual Selectors</title>
        <style type="text/css">
            p  em { color: red; }
            p > em { color: blue; }
            p + em { color: green; }
            em ~ strong { color: orange; }
        </style>
    </head>

    <body>
        <p>
            <strong><em>descendant selector</em></strong>
        </p>
        <p>
            <em> child selector</em>
        </p>
        <em>adjacent sibling selector</em>
        <p>
            <em>child selector</em>
            <strong>general sibling selector</strong>
        </p>
    </body>
</html>
```

Demo Demo2



descendant selector

child selector

adjacent sibling selector

child selector **general sibling selector**

ID Selectors

- The id attribute gives an element a unique identifying name.
- The id attribute can be used with any HTML element.
- It is commonly used to give meaning to the generic div and span elements.
- ID selectors can be used to target elements by their id values.
- The symbol that identifies ID selectors is the hash symbol (#).

```
li#catalog1234 { color: red; }  /* list item using an ID selector */

<li id="catalog1234">Happy Face T-shirt</li>

#catalog1234 { color: red; }    /*id selector */

#links li { margin-left: 10px; }   /*contextual selector with ID selector*/
```

Class selectors

- The class identifier is used to classify elements into a conceptual group.
- Multiple elements can share a class name but an element may belong to more than one class.
- A class selector is used to target elements belonging to the same class.
- Class name are indicated with a period (.) at the beginning of the selector.

```
p.special  { color: orange; }  /* all paragraphs with class "special"  */

.special { color: orange }      /* all elements with class "special"  */
```

The Universal Selector

- CSS2 introduced a universal element selector (*) that matches any element.

```
*   { color: gray; }  /* set the foreground of every element */
```

```
#intro  * { color: gray; } /* select all elements in an "intro" section */
```

- The universal selector causes problems with form controls in some browsers. If the page contains form inputs, the safest way is to avoid the universal selector.

line-height

Values:	number | length measurement | percentage | normal | inherit
Default:	normal
Applies to:	all elements
Inherits:	yes

- The line-hight property defines the minimum distance from baseline to baseline in text

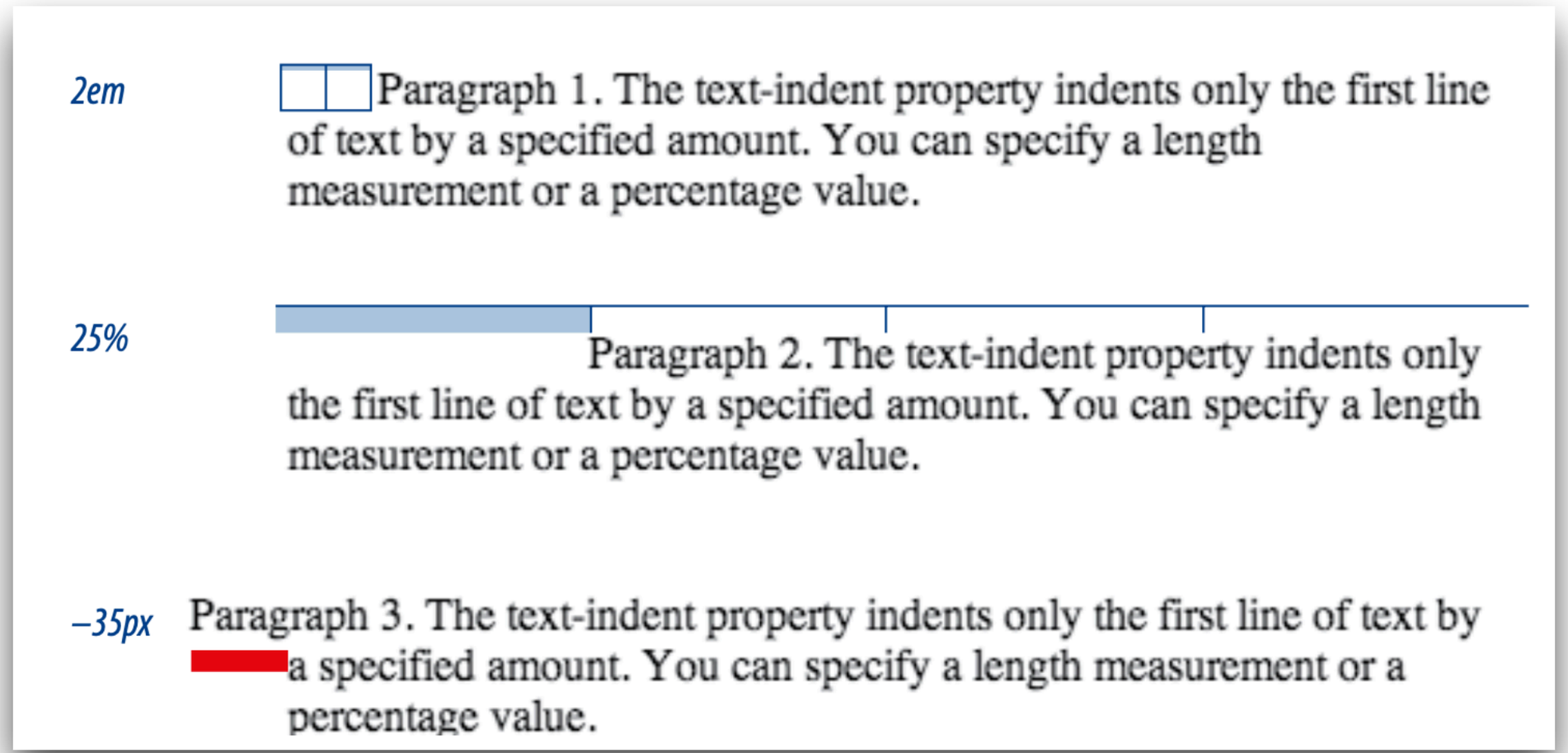
```
p   { line-height: 2; } /* 2*current font size */

p   { line-height: 2em; }

p   { line-height: 200%; }
```

text-indent

Values:	length measurement | percentage | inherit
Default:	0
Applies to:	block-level elements, table cells, and inline blocks
Inherits:	yes

```
p#1    { text-indent: 2em; }

p#1    { text-indent: 25%; }

p#1    { text-indent: -35px; }
```

text-indent


2em
Paragraph 1. The text-indent property indents only the first line of text by a specified amount. You can specify a length measurement or a percentage value.
25%
Paragraph 2. The text-indent property indents only the first line of text by a specified amount. You can specify a length measurement or a percentage value.
–35px
Paragraph 3. The text-indent property indents only the first line of text by a specified amount. You can specify a length measurement or a percentage value.

text-align

Values:	left | right | center | justify | inherit
Default:	left for languages that read left to right right for language that read right to left
Applies to:	block-level elements, table cells, and inline blocks
Inherits:	yes

```
h1    { text-align: left; }
h1    { text-align: right; }
h1    { text-align: center; }
p     { text-align: justify; }
```

text-align

text-align: left

Paragraph 1. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page. The resulting text behavior of the various values should be fairly intuitive.

text-align: right

Paragraph 2. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page.The resulting text behavior of the various values should be fairly intuitive.

text-align: center

Paragraph 3. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page.The resulting text behavior of the various values should be fairly intuitive.

text-align: justify

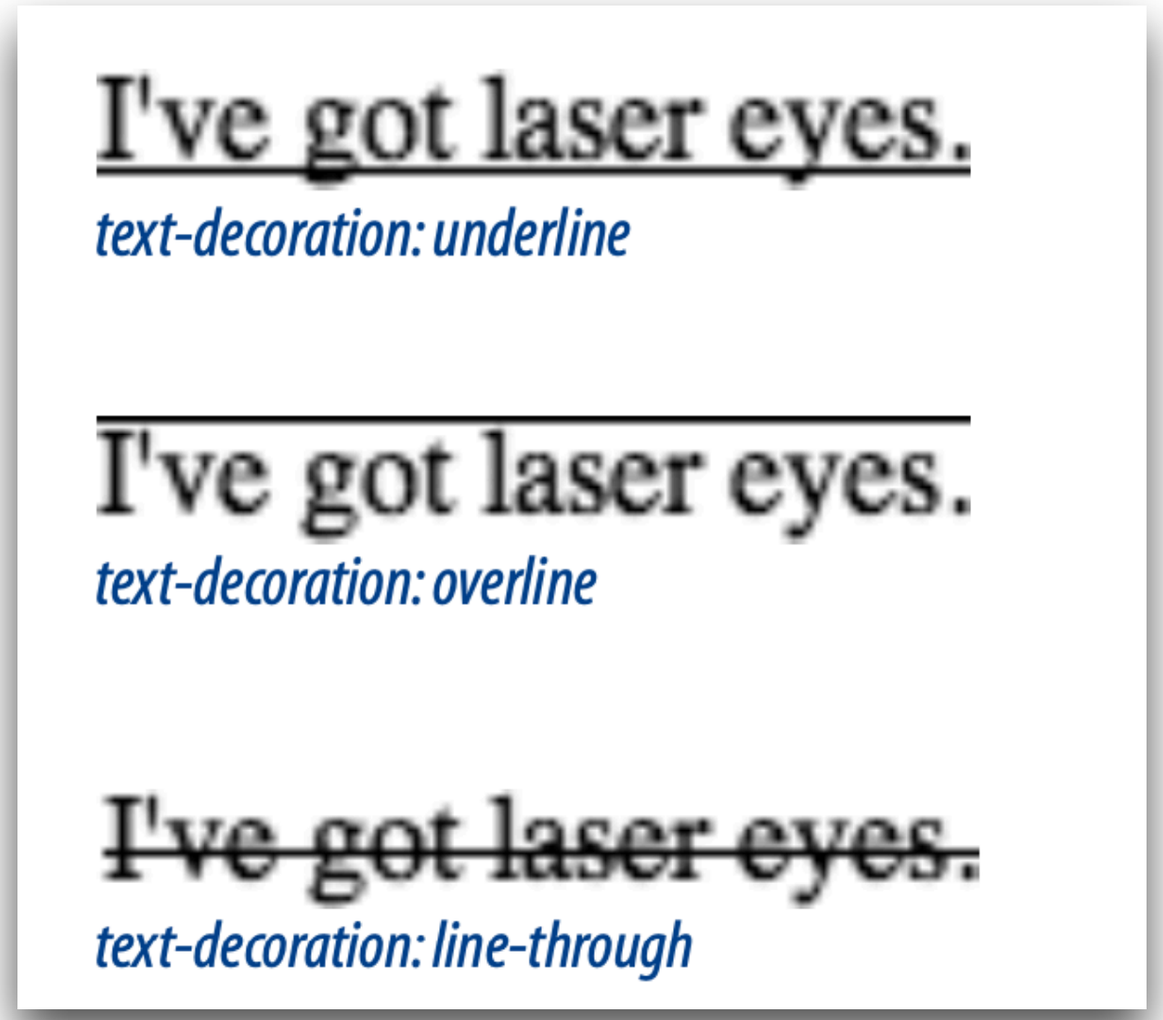
Paragraph 4. The text-align property controls the horizontal alignment of the text within an element. It does not affect the alignment of the element on the page.The resulting text behavior of the various values should be fairly intuitive.

text-decoration

Values:	none | underline | overline | line-through | blink
Default:	none
Applies to:	all elements
Inherits:	no, but since lines are drawn across child elements; they may look like they are "decorated" too

```
a  { text-decoration: underline; }
```

text-decoration



text-transform

Values:	none | capitalize | lowercase | uppercase | inherite
Default:	none
Applies to:	all elements
Inherits:	yes

```
p    { text-transform: none; }
p    { text-transform: capitalize; }
p    { text-transform: lowercase; }
p    { text-transform: uppercase; }
```

text-transform

And I know what you're thinking.

text-transform: none (as was typed in)

And I Know What You'Re Thinking.

text-transform: capitalize

and i know what you're thinking.

text-transform: lowercase

AND I KNOW WHAT YOU'RE THINKING.

text-transform: uppercase

letter-spacing, word-spacing

Values:	length measurement | normal | inherite
Default:	normal
Applies to:	all elements
Inherits:	yes

```
p    { lettter-spacing: 8px; }

p    { word-spacing: 1.5em; }
```

letter-spacing, word-spacing

letter-spacing: 8px;

B l a c k G o o s e B i s t r o S u m m e r M e n u

word-spacing: 1.5em;

Black Goose Bistro Summer Menu

text-shadow

Values:	'horizontal offset' 'vertical offset' 'blur radius' 'color' | none
Default:	none
Applies to:	all elements
Inherits:	yes

```
h1    { text-shadow: .2em .2em silver; }

h1    { text-shadow: -.3em -.3em silver; }

h1    { text-shadow: .2em .2em .05em silver; }
```

text-shadow

The Jenville Show text-shadow: .2em .2em silver

The Jenville Show text-shadow: -.3em -.3em silver;

The Jenville Show ext-shadow: .2em .2em **.05 em** silver

The Jenville Show ext-shadow: .2em .2em **.15 em** silver

The Jenville Show ext-shadow: .2em .2em **.3 em** silver