EAST WEST UNIVERSITY

CSE479

Web Programming

Nishat Tasnim Niloy

Lecturer

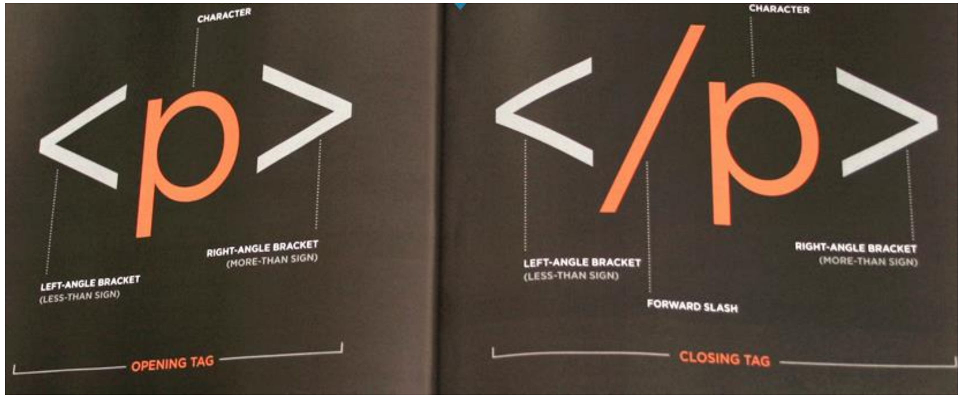
Department of Computer Science and Engineering

Faculty of Science and Engineering

Topic 2

CSS (Cascading Style Sheets)
(element styling, spacing, box model for borders)

By the end of this unit you should be able to...



❑ Explain how CSS works

❑ Write CSS rules

❑ Apply CSS rules to an HTML page

❑ Specify colors for various elements with CSS

❑ Use color terminology appropriately

❑ Use CSS to specify contrast and background colors for a page or part of a page

❑ Use CSS to specify style and typeface of text

❑ Transform text with CSS and affect other properties of text

❑ Specify spacing between lines, words, and letters

❑ Control the dimensions of boxes

❑ Explain the CSS box model for borders, margin, and padding

❑ Create borders around boxes

❑ Set margins, and padding for boxes

❑ Display and hide boxes

Understanding CSS: thinking inside the box

CSS allows you to write rules that specify how the content of an element should be presented.

The key to understanding CSS is to imagine that there is an invisible box around every HTML element.

CSS allows you to write rules that control the way that each individual box and its content is presented.

You can add border to any box, specify width and height, add background color, control appearance of text, etc.

CSS associates style rules with HTML elements

Rules govern how the content of the specified elements should display.

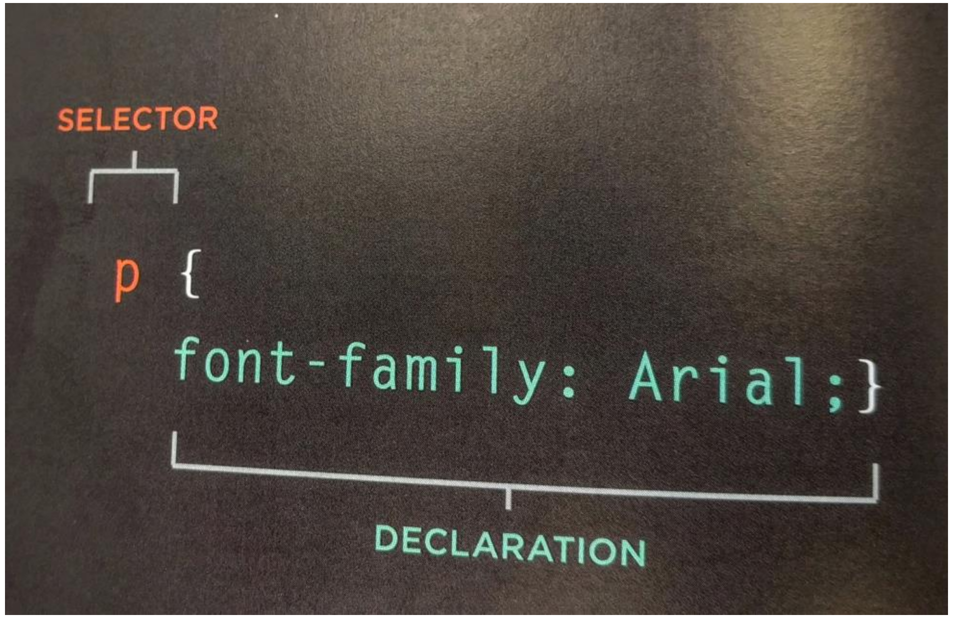
A rule has 2 parts:

- Selector indicates which elements the rule apply to
- Declaration block indicates how the element should be styled

Declarations are split into 2 parts

- Property
- Value

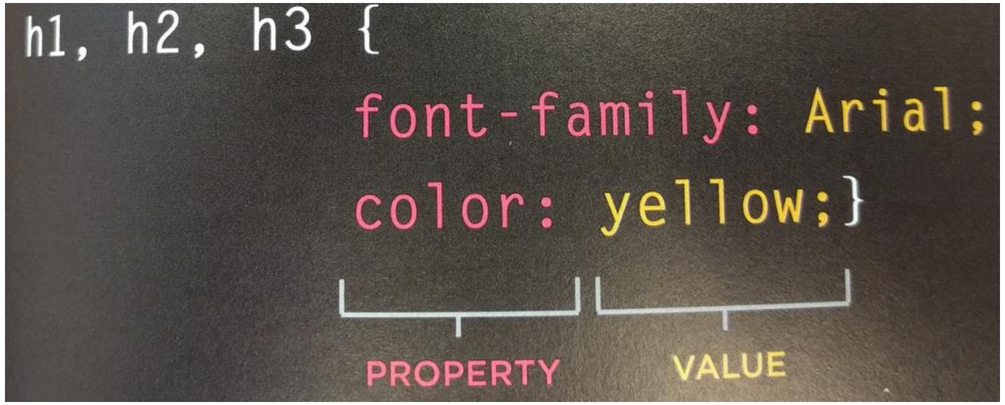
They are separated by a colon



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

CSS properties affect how elements are displayed

You can specify several declarations in one **declaration block**, each separated by a semicolon.



```
h1, h2, h3 {
        font-family: Arial;
        color: yellow;}
```

PROPERTY
VALUE

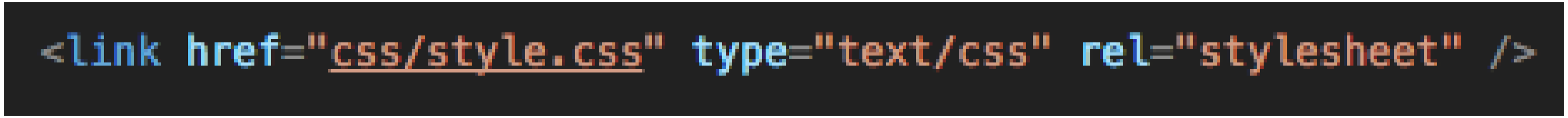
This rule indicates that all h1, h2, and h3 elements should be shown

- In Arial typeface
- In a yellow color

Properties and values

- A property indicates the aspect of an element you wish to change
- A value indicates the settings you want to use for the identified property

Using external CSS



```
<link href="css/style.css" type="text/css" rel="stylesheet" />
```

Use link element:

- Include in HTML document to tell browser where to find css file
- Is an empty element
- It lives inside the head element
- Can have more than one (for presentation, layout, etc)

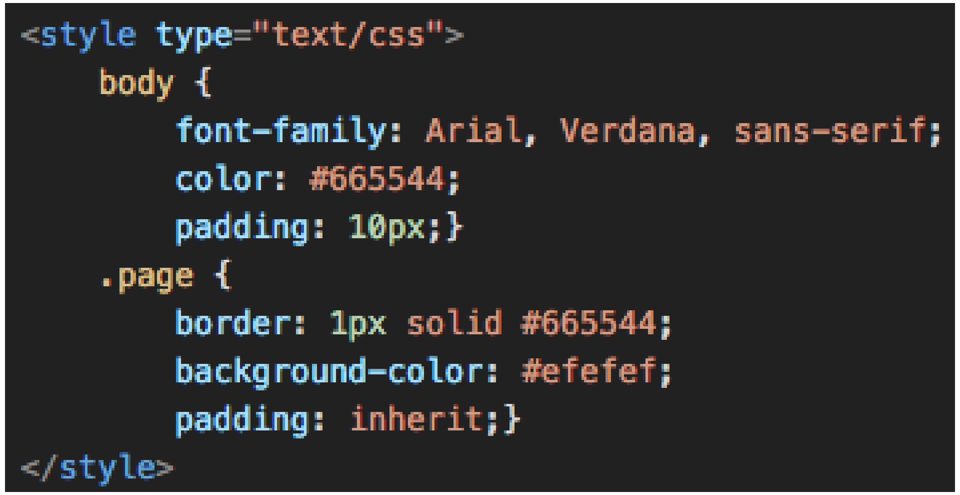
Attribute:

- href specifies the path to the css file (usually in css folder)
- type specifies the type of document being linked to
- rel specifies the relationship between the HTML page and the file being linked to

Use internal CSS sparingly

Use style element:

- Use type attribute to indicate that the styles are specified in CSS



```
<style type="text/css">
    body {
        font-family: Arial, Verdana, sans-serif;
        color: #665544;
        padding: 10px;}
    .page {
        border: 1px solid #665544;
        background-color: #efefef;
        padding: inherit;}
</style>
```

Use external stylesheets instead:

- When building a site with more than one page
    - All pages can use the same style rules
    - **Keep the content and structure separate from how the page looks**
    - Change styles once, see the effect on all pages

CSS Selectors

CSS selectors allow you to target rules to specific elements in an HTML document
Selectors are case sensitive.

Must match elements names and attribute values exactly!

http://www.w3schools.com/cssref/css_selectors.asp CSS selectors reference
http://www.w3schools.com/cssref/trysel.asp CSS selector tester (see a few examples)

Commonly used CSS Selectors

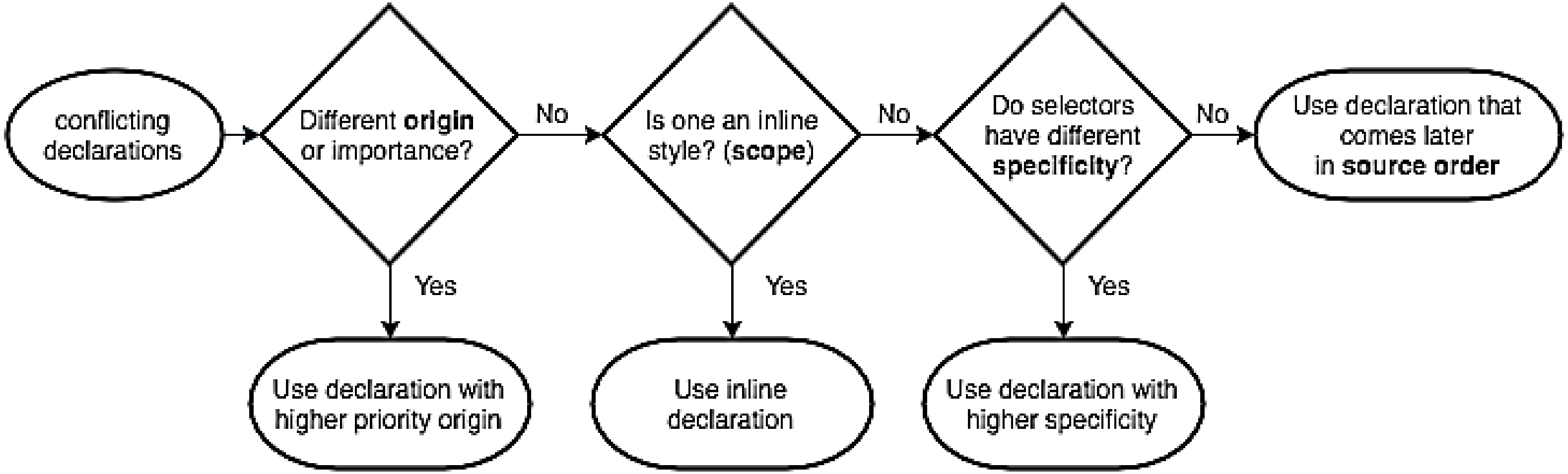
❑	Universal selector	* { }	
❑	Type selector	h1, h2, h4 { }	
❑	Class selector	.note { }	
❑	ID selector	#listfriends { }	
❑	Child selector	li>a { }	Selects <a> elements where ***direct parent*** is a <li> element
❑	Descendent selector	p a { }	Selects <a> elements ***anywhere inside*** <p> elements
❑	Adjacent sibling selector	h2+p { }	Selects <p> elements placed ***immediately after*** <h2> elements
❑	General sibling selector	h1~p { }	Selects <p> elements ***preceded anywhere*** by a <h1> sibling

Cascading of CSS rules / declarations

Two or more declarations apply to the same element. Which should take precedence?

- **Last declaration**
  - If all the selectors are identical, last one takes precedence
  - E.g., see ***color*** property of the content of the ***<i>*** element
- **Most specific declaration**
  - More specific one takes precedence over more general
  - E.g., ***h1*** over *******; ***p b*** over ***p***
- **Important** (use cautiously)
  - Adding ***!important*** after any property value indicates that it should be considered more important than other declarations that apply to same element

Cascading of CSS declarations



conflicting declarations

Different **origin** or importance?

No

Is one an inline style? (**scope**)

No

Do selectors have different **specificity**?

No

Use declaration that comes later in **source order**

Yes

Use declaration with higher priority origin

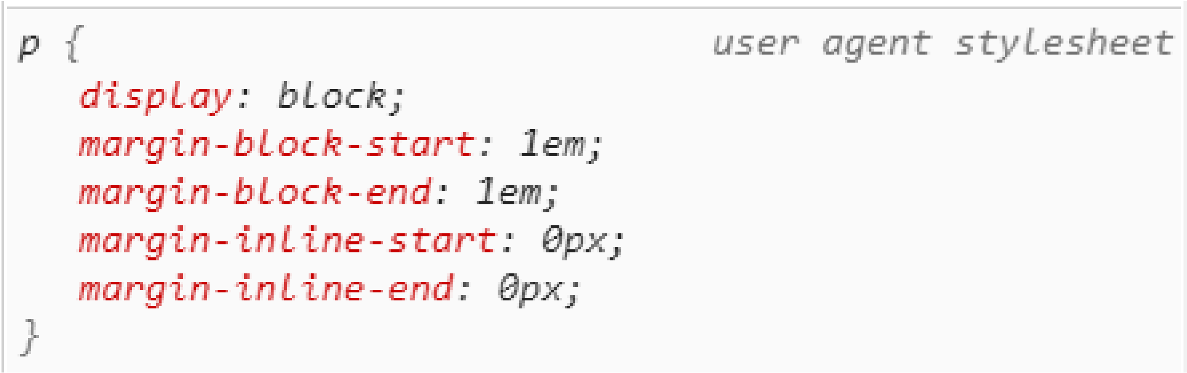
Yes

Use inline declaration

Yes

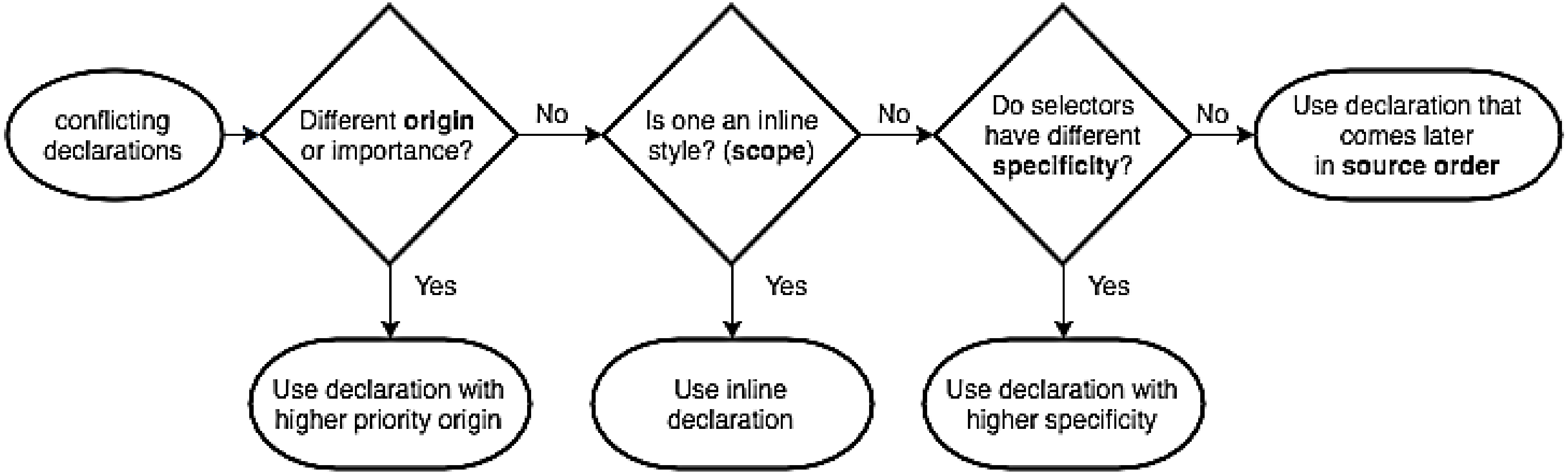
Use declaration with higher specificity

Origin => user agent declarations or author declarations



```
p {                                    user agent stylesheet
   display: block;
   margin-block-start: 1em;
   margin-block-end: 1em;
   margin-inline-start: 0px;
   margin-inline-end: 0px;
}
```

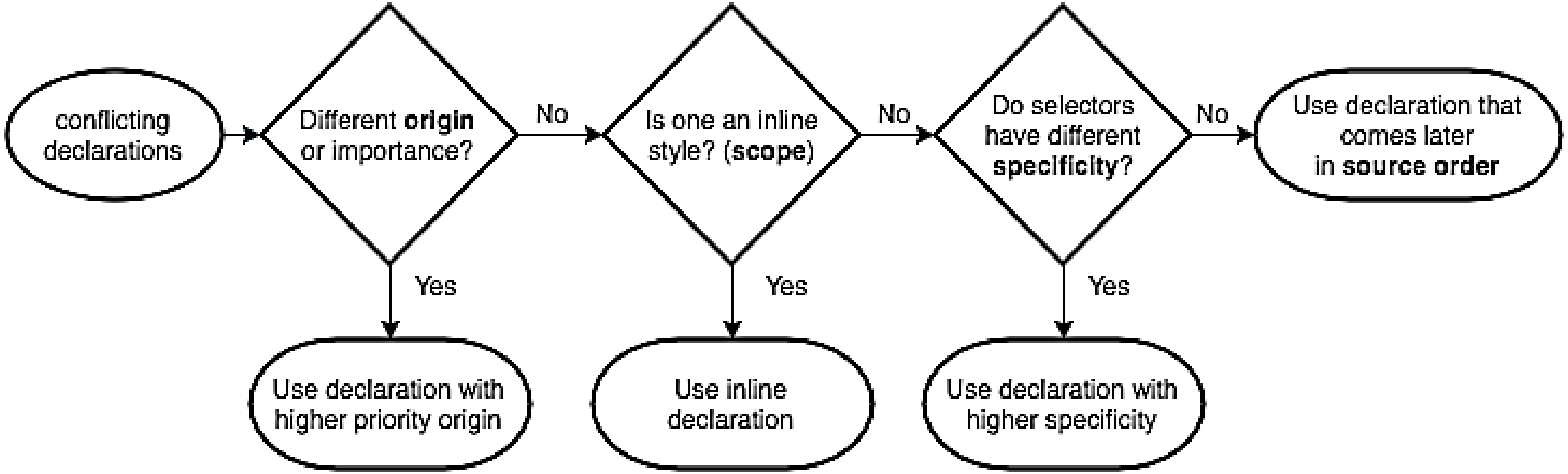
Cascading of CSS declarations


conflicting declarations
Different **origin** or importance?
No
Is one an inline style? (**scope**)
No
Do selectors have different **specificity**?
No
Use declaration that comes later in **source order**
Yes
Use declaration with higher priority origin
Yes
Use inline declaration
Yes
Use declaration with higher specificity

Scope => a set of styles may be scoped only to a particular element and its children. e.g.,

```
<a href="/specials" class="featured" style="background-color: orange;">
        Specials
</a>
```

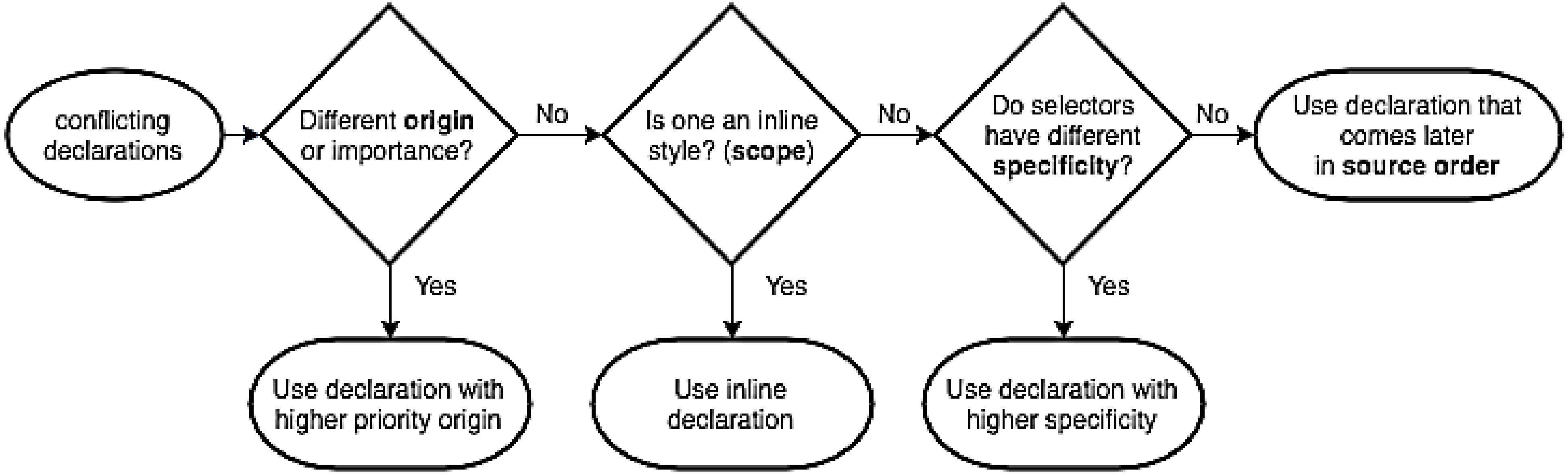
Cascading of CSS declarations


conflicting declarations
Different **origin** or importance?
No
Is one an inline style? (**scope**)
No
Do selectors have different **specificity**?
No
Use declaration that comes later in **source order**
Yes
Use declaration with higher priority origin
Yes
Use inline declaration
Yes
Use declaration with higher specificity

Specificity => how specific we can be in determining the selectors whose rules/declarations take precedence.

Rules: more ids win;
if tie, more classes win;
if tie, more tag names win

Cascading of CSS declarations


conflicting declarations
Different **origin** or importance?
No
Is one an inline style? (**scope**)
No
Do selectors have different **specificity**?
No
Use declaration that comes later in **source order**
Yes
Use declaration with higher priority origin
Yes
Use inline declaration
Yes
Use declaration with higher specificity

Source order => how early or late the declaration appears in stylesheet(s)

Rules: the declaration that appears later in the stylesheet—or appears in a stylesheet included later on the page—wins.

Inheritance

Specify ***font-family*** or ***color*** property on ***<body>*** element.

Effect: They will apply to most elements.

Reason: These properties are inherited by child elements.

Some properties are **not inherited** by child elements.
E.g.

background-color
border

Can force a lot of properties to inherit values from their parents
How? Use ***inherit*** as the **value** of the property

Some browser quirks and CSS versions

History:
CSS1 was released in 1996
CSS2 was released in 1998
CSS3 work in ongoing, since 1998

- **Note**
  - Browsers do not implement all CSS features at once
  - Some older browsers do not support every property
- **Recommendation**
  - Test your site in several browsers because different browsers may display your page differently
- **CSS Bug or browser quirk**
  - When a CSS property does not display as expected, it is referred to one of these.
  - Search for fix or solution

Color property

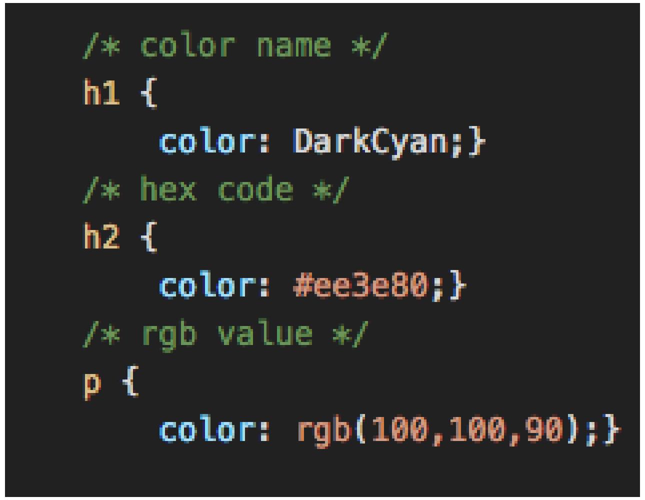
Allows you to specify the color of text inside an element - http://www.w3schools.com/cssref/css_colors.asp

Ways to specify any color in CSS

❑ **Color names:** - 147 predefined color names recognized by browsers. E.g. DarkCyan

❑ **Hex codes:** 6-digit hexadecimal codes that represent the amount of red, green, and blue in a color. E.g., #ee3e80

❑ **RGB values:** how much red, green, and blue are used to make up the color. E.g. **rgb(100, 100, 90)**



```
/* color name */
h1 {
    color: DarkCyan;}
/* hex code */
h2 {
    color: #ee3e80;}
/* rgb value */
p {
    color: rgb(100,100,90);}
```

Background color property

CSS treats each HTML element as if it appears in a box. The **background-color** property sets the color of the background of that box.

http://htmlandcssbook.com/code-samples/chapter-11/background-color.html

Can specify background-color property using RGB values, hex codes, or color names.

Background is ***transparent*** if background-color is not specified

By default, most browser windows have a white background, but you can specify differently on the body element

Understanding color

Color picking tool - http://paletton.com/

- ❑ Also available in image editing programs
- ❑ w3schools discussion on Colors HSL

CSS3 opacity and rgba properties

Opacity: describes the transparency level of an element.
http://htmlandcssbook.com/code-samples/chapter-11/example.html

- ❑ 1 - not transparent at all
- ❑ .5 is 50 % see through
- ❑ 0 is completely transparent

rgba: specify RGB color and add opacity value (called alpha)

E.g. ***rgba(0, 0, 0, 0.5)***

rgba fallback for older browsers:

Specify an element’s property color using name, hex code, or RGB value

Specify rgba color (same color) as last rule for same element

Properties that allow you to control the appearance of text

Two groups of properties:

Properties that affect the font and its appearance
(typeface, size, regular, bold, italics)

Properties that have same effect on text regardless of font
(color, spacing between words/letters)

Text formatting can significantly affect readability

TYPEFACE TERMINOLOGY

SERIF

Serif fonts have extra details on the ends of the main strokes of the letters. These details are known as serifs.

im

In print, serif fonts were traditionally used for long passages of text because they were considered easier to read.

SANS-SERIF

Sans-serif fonts have straight ends to letters, and therefore have a much cleaner design.

im

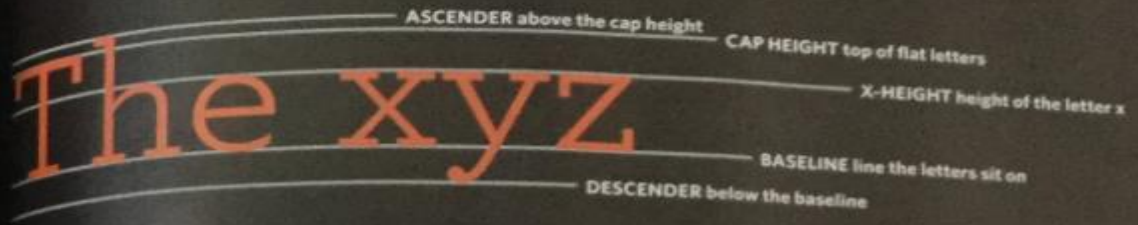
Screens have a lower resolution than print. So, if the text is small, sans-serif fonts can be clearer to read.

MONOSPACE

Every letter in a monospace (or fixed-width) font is the same width. (Non-monospace fonts have different widths.)

im

Monospace fonts are commonly used for code because they align nicely, making the text easier to follow.


The xyz
ASCENDER above the cap height
CAP HEIGHT top of flat letters
X-HEIGHT height of the letter x
BASELINE line the letters sit on
DESCENDER below the baseline

WEIGHT

Light

Medium

Bold

Black

The font weight not only adds emphasis but can also affect the amount of white space and contrast on a page.

STYLE

Normal

Italic

Oblique

Italic fonts have a cursive aspect to some of the lettering. Oblique font styles take the normal style and put it on an angle.

STRETCH

Condensed

Regular

Extended

In condensed (or narrow) versions of the font, letters are thinner and closer together. In expanded versions they are thicker and further apart.

Choosing a typeface for your website

Note that browsers will display your chosen typeface if it is installed on the user's computer

Serif: have extra details on the end of the main strokes of letters *(Georgia, Times, Times New Roman)*

Sans-Serif: have straight ends to letters and have cleaner design *(Arial, Verdana, Helvetica)*

Monospace: every letter has the same width *(Courier, Courier New)*

Cursive: have joining strokes, handwriting styles *(Comic Sans MS, Monotype Corsiva)*

Fantasy: decorative fonts, usually used for titles, not long text

(Impact, Haettenschweiler)

Choosing a typeface for your website

Use a font stack: Use more than one typeface to express an order of preference. Good when a user does not have the first typeface installed on their computer - **Font Family**

Browsers are supposed to support at least one font from each of the groups identified on the previous slide. It is a good reason to specify the generic font name from the stack last, after your preferred choice. E.g.,

font-family: Georgia, Times, Serif;

Use this property to specify font family for the text
inside the element to which the CSS rule applies

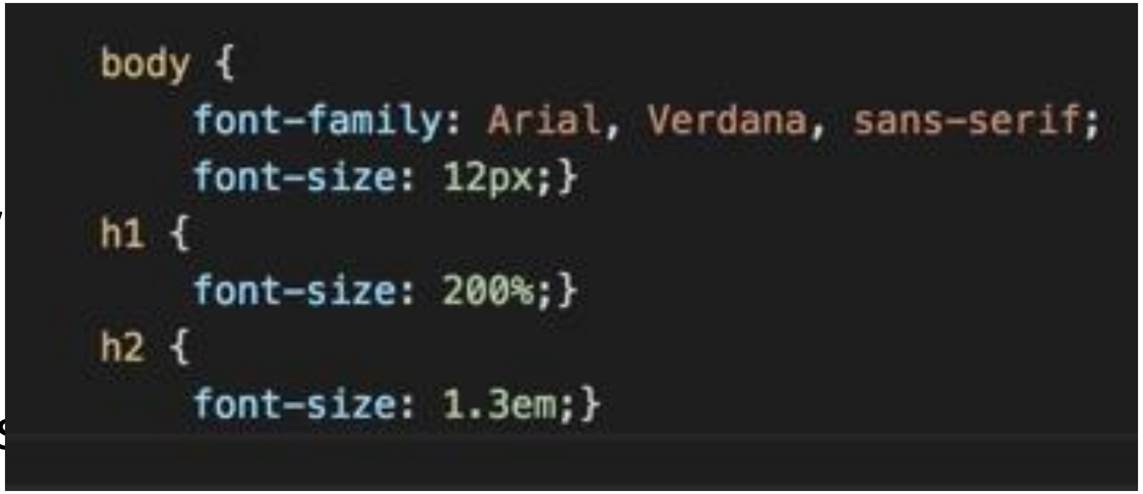
Size of type -- font-size property

Specify the size of the font.

Ways to specify font size:
Pixels: offers designers control over exactly how much space their text takes up.

Percentages: The default size of font in browsers 16px. 75% == 12px; 200% == 32px.
What is the result of nesting percentages?

Ems: An em is equivalent to the width of the letter m



```
body {
    font-family: Arial, Verdana, sans-serif;
    font-size: 12px;}
h1 {
    font-size: 200%;}
h2 {
    font-size: 1.3em;}
```

Open source font

Use ***@font-face*** rule to use a font even it is not installed on the computer of the person viewing your site

Need to specify a path (in ***src*** property) to copy of the font, which will be downloaded on the user’s computer if not yet downloaded

Need to specify the name of the font and download multiple formats (eot, woff, ttf/otf, svg)

Free font lists:
http://www.fontex.org
https://www.fontsquirrel.com
http://openfontlibrary.com

Font weight and style

Use ***font-weight*** property to allow you to create **bold** text

Values of this property are ***bold*** and ***normal***. Why does a normal value exist? Hint: consider body text having ***font-weight: bold;***

Use ***font-style*** property to create *italic text*

This property takes on **3 possible values**:

normal: causes text to appear in a normal font
italic: causes text to appear italic
oblique: causes text to appear oblique

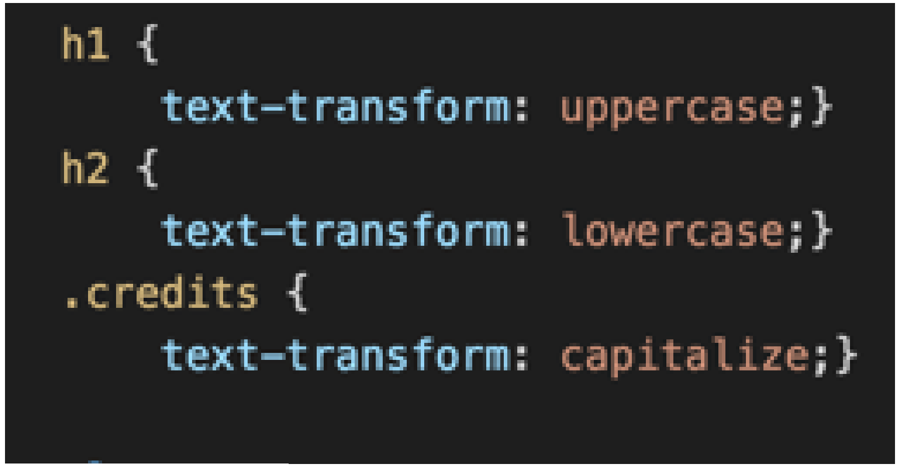
Text transformation

The ***text-transform*** property is used to change the case of text.

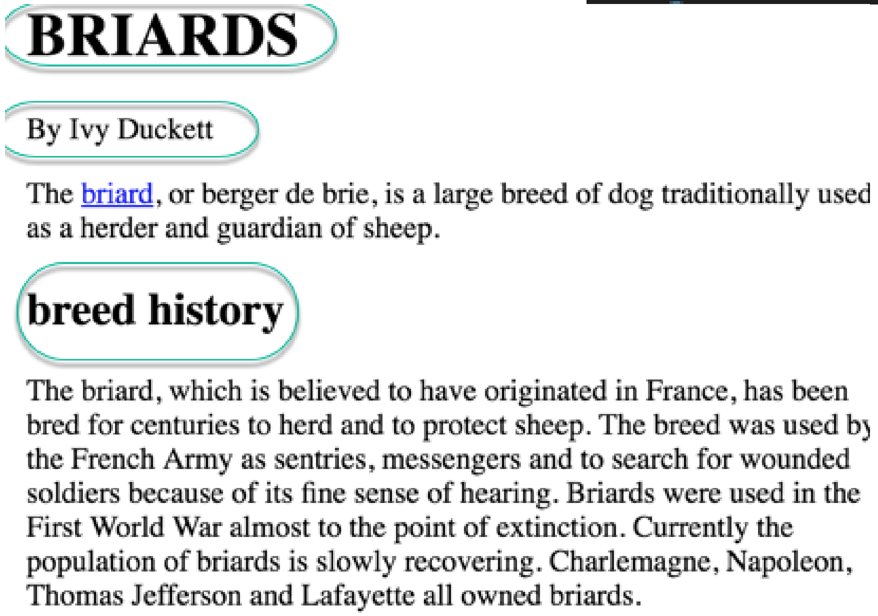
Possible values:
uppercase: all characters appear in uppercase

capitalize: the first letter in each word appears capitalized

lowercase: all characters appear in lowercase



```
h1 {
    text-transform: uppercase;}
h2 {
    text-transform: lowercase;}
.credits {
    text-transform: capitalize;}
```



BRIARDS

By Ivy Duckett

The briard, or berger de brie, is a large breed of dog traditionally used as a herder and guardian of sheep.

breed history

The briard, which is believed to have originated in France, has been bred for centuries to herd and to protect sheep. The breed was used by the French Army as sentries, messengers and to search for wounded soldiers because of its fine sense of hearing. Briards were used in the First World War almost to the point of extinction. Currently the population of briards is slowly recovering. Charlemagne, Napoleon, Thomas Jefferson and Lafayette all owned briards.

Text decoration

Use the ***text-decoration*** property to add decoration to text content

Property values:
none: removes any decoration already applied to the text

underline: adds a line underneath the text

line-through: adds a line through words

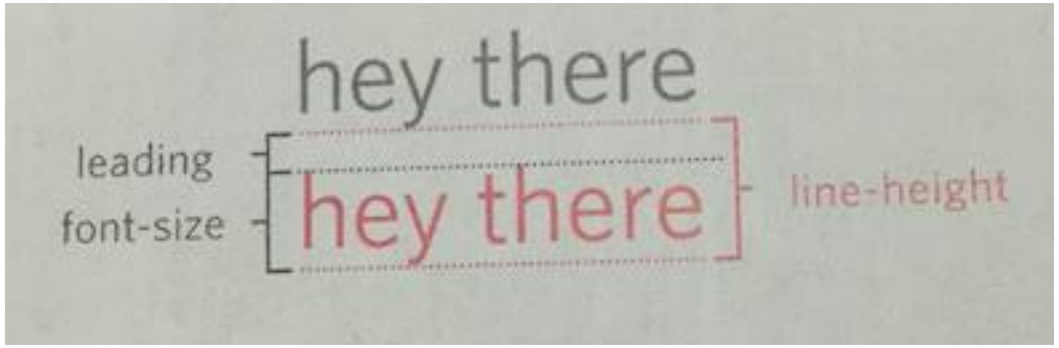
overline: adds a line over the top of text

blink: animates the text to make it flash on and off (annoying)

Leading, line-height,

p {

line-height: 1.4em; }



Leading (pronounced ledding) is the space between **descender** and **ascender**.

In CSS the ***line-height*** property sets the height of an entire line of text, so the difference between the ***font-size*** and the ***line-height*** is equivalent to the leading.

Best to measure ***line-height*** in ***ems****,* not **pixels** because the gap between lines is relative to the size of the text the user has selected

Letter and word spacing

Use the ***letter-spacing*** property to increase or decrease ***kerning*** (the space between each pair of letters)

Should increase for uppercase headings or sentences

Use the ***word-spacing*** property to control the gap between word

The values for these should be given in **ems**, not **pixels**

The default gap between words is set by the typeface (~ 0.25em)

Text alignment (horizontal)

Use the ***text-align*** property to control the *horizontal* alignment of text

Property values:
left: indicates that the text should be left-aligned

right: indicates that the text should be right-aligned

center: indicates that the text should be centered

justify: indicates that the text should take up the full width of the containing box

Left-aligned works well when you have several
paragraphs of text

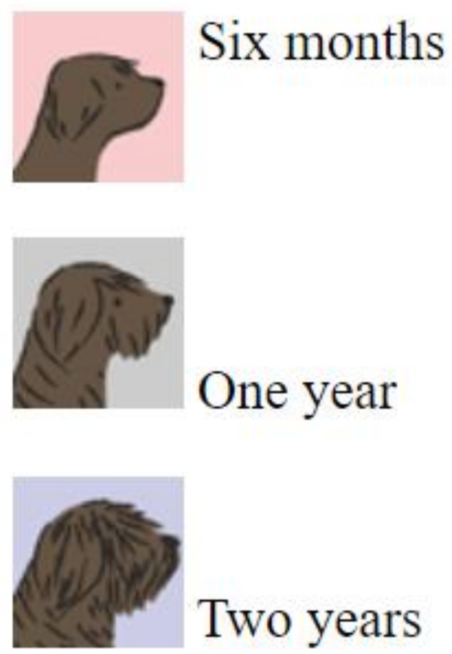
Text alignment (vertical)

Use the ***vertical-align*** property to control the *vertical* align of text relative to neighboring **inline elements**. Should apply to the neighboring **inline element**. This is a common source of confusion.

http://htmlandcssbook.com/code-samples/chapter-12/vertical-align.html

Property values:

- **baseline**
- **sub**
- **super**
- **top**
- **text-top**
- **middle**
- **bottom**
- **text-bottom**



Text indentation and CSS3 drop shadow

Use the ***text-indent*** property to indent the **first line of text** within an element.
http://htmlandcssbook.com/code-samples/chapter-12/text-indent.html
Specify amount of indentation in **pixels** or **ems**. Can take negative values.

http://htmlandcssbook.com/code-samples/chapter-12/text-shadow.html
Use the **text-shadow** property to create a drop shadow.

```
p {
        color: #cccccc;
        text-shadow: -2px 2px 7px #111111;}
```

First value: how far to the left or right the shadow should fall
Second value: the distance to the top or bottom shadow should fall
Third value: optional, but specifies amount of blur
Fourth value: color of drop shadow

First letter, first line (pseudo-elements)

::first-letter OR ***::first-line*** can be used to specify different values for the first letter or line inside an element.

USE   :: as a convention to make it clear to web developers

These are not properties, but **pseudo-elements**. Acts like an extra element in the code

Specify these **pseudo-elements** at the end of the selector, then specify the declarations how you would normally for the element

http://htmlandcssbook.com/code-samples/chaper-12/first-letter-and-line.html

Styling links (pseudo-classes)

Browsers tend to show links in blue and underlined by default and change the appearance of visited links to help users know which links they have visited

http://htmlandcssbook.com/code-samples/chapter-12/link-visited.html

In CSS use ***:link*** OR ***:visited*** to change the appearance of links

:link allow you to set style for links that have not been visited

:visited allow you to set style for links that have been clicked on

These are not classes, but **pseudo-classes**. Acts like an extra value for a class attribute

Responding to users (pseudo-classes)

Use the following **pseudo-classes** to change the appearance of an element when a user interacts with them

http://htmlandcssbook.com/code-samples/chapter-12/hover-active-focus.html

:hover this is applied when a user hovers over an element (e.g., link, button) with a pointing device such as a mouse

:active this is applied when an element is activated by a user (e.g., when a button is being pressed or when a link is being clicked)

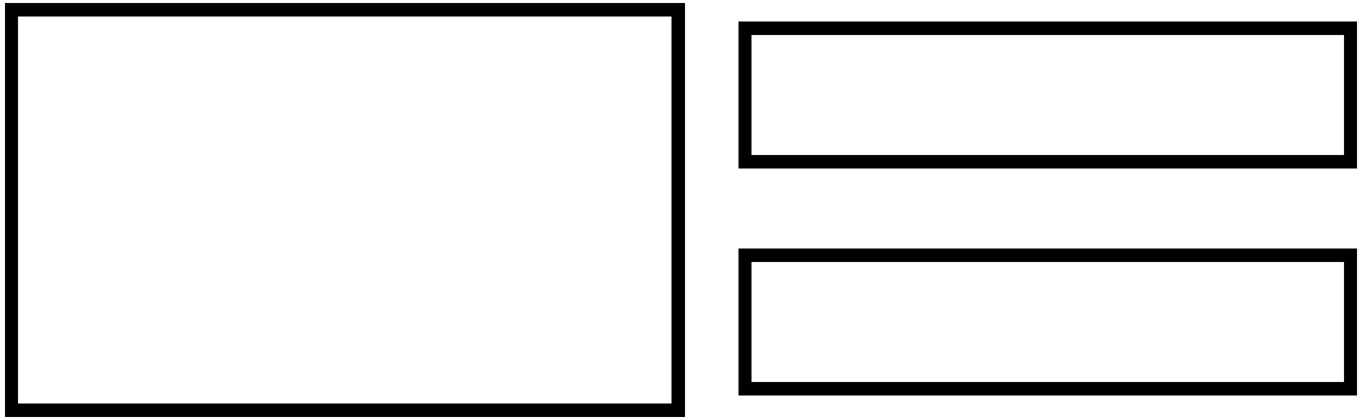
:focus this is applied when an element has focus
(e.g., link, form control)

Stop and think for a moment... Why should we care?

- ❑ Why is it significant to know how to adjust everything?

- ❑ Why is the web important in general?

- ❑ Why make some design like this below? (or a better one)



Box Dimensions

By default a box is just big enough to hold its content.

You can use the ***width*** and ***height*** properties to change box dimensions. This is **only** the width / height of the **content** of the box.

Can specify dimensions in **pixels**, **percentages**, or **ems** -- pixels most commonly used

Limiting dimensions:

Shrink or stretch size of boxes to fit user’s screen

- ❑ ***min-width*** / ***max-width*** properties shrink or stretch the width
- ❑ ***min-height*** / ***max-height*** properties shrink or stretch the height

What if the box is too small for its content?

If the size of a box is too small for its content, the content can expand outside the box and it can look messy

http://htmlandcssbook.com/code-samples/chapter-13/min-height-max-height.html

To control what happens when there is not enough space inside a box for its content, use the ***overflow*** property.

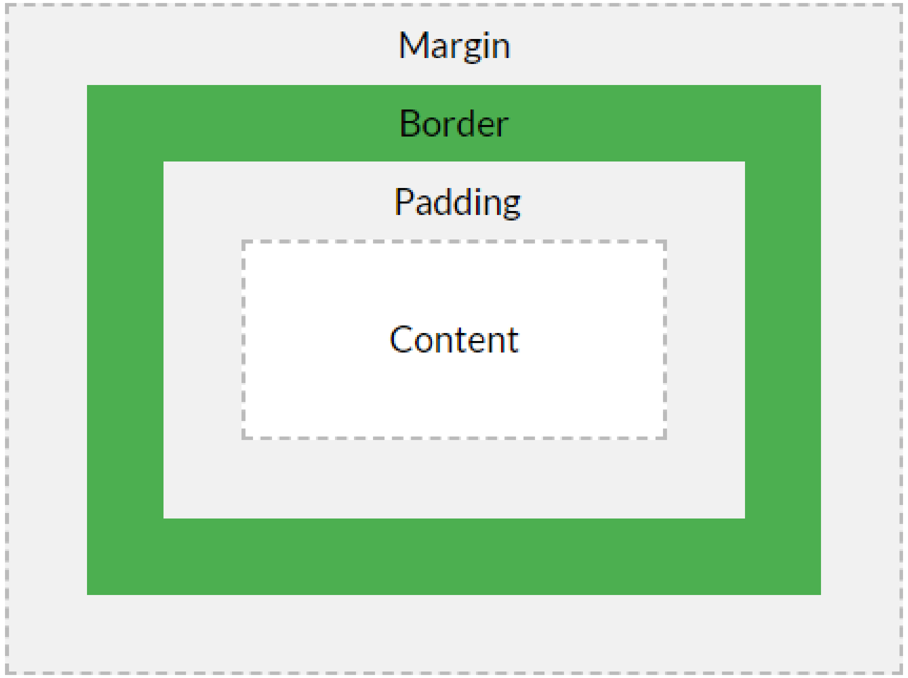
overflow property values:

- ❑ ***hidden***: hides any extra content that does not fit inside the box
- ❑ ***scroll***: adds a scrollbar to the box so users can scroll to see missing content

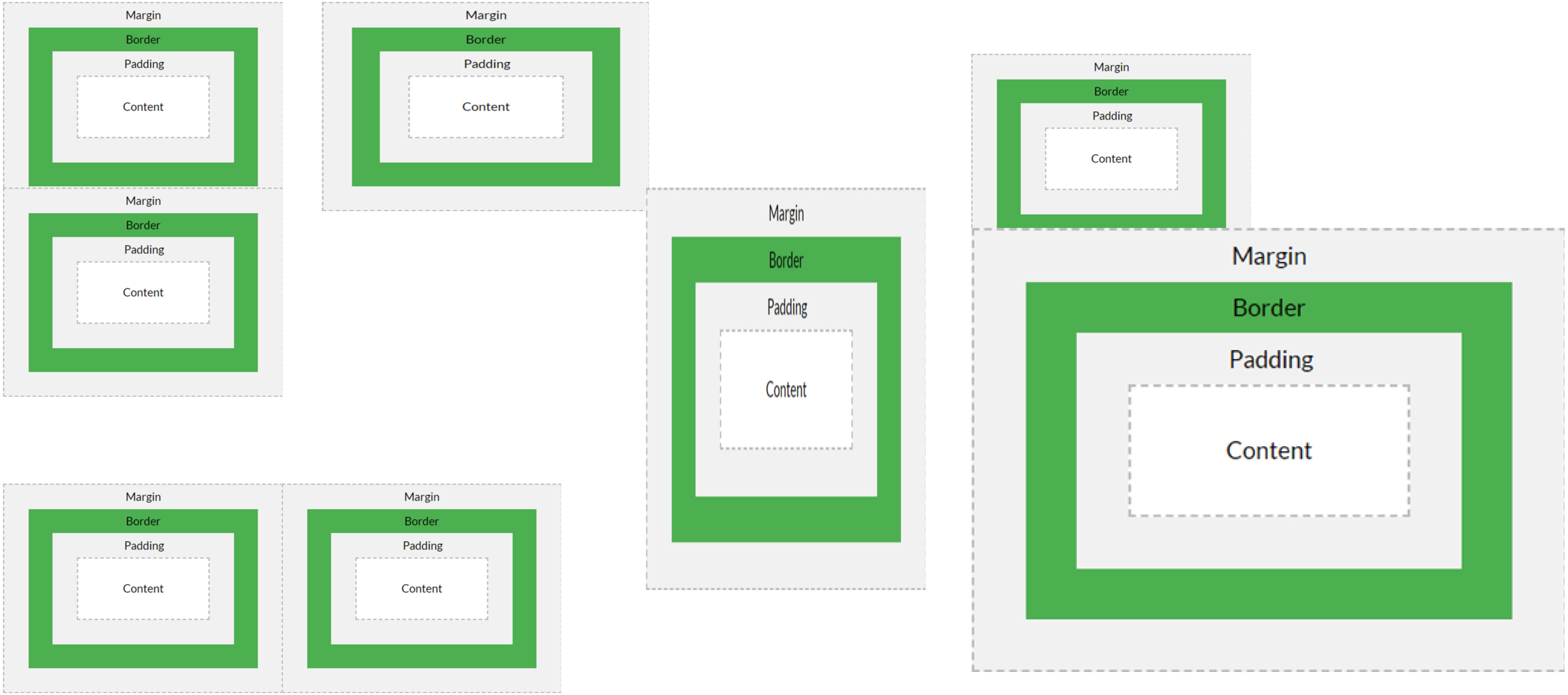
Box model for border, margins, and padding

Visual of box model - http://www.w3schools.com/css/css_boxmodel.asp

- ***border***:
  - Every box has a ***border***, even if it is invisible or specified to be 0 ***pixels*** wide.
  - It separates the edge of one box from another
- ***margin***:
  - Sits outside the edge of the border.
  - Can set ***margin*** width to create a gap between the ***border*** of two adjacent boxes
- ***padding***:
  - The space between the ***border*** of a box and its content.
  - Adding ***padding*** increases readability of the content



For overlapping margins, largest margin wins



Controlling border: width

border-width: controls the width of the border. Value given in pixels or using one of these values: ***thin***, ***medium***, ***thick***

p.one {   border-width: 5px;   }

Can control individual size of borders using one of these properties:

border-top-width, ***border-right-width***, ***border-bottom-width***, ***border-left-width***

Can also specify different border widths for the four border values in one property

p.three {   border-width: 2px 4px 12px 4px;   }

(top right bottom left) TRBL   (remember "trouble" and you won't have trouble!)

(or think of a clock that starts at 12 and goes around...)

Controlling border: style

border-style: controls the style of the border. See possible values for the border-style property at http://www.w3schools.com/css/css_border.asp.

p.one {   border-style: dotted;   }

Can individually change the style of different borders using one of these properties:

border-top-style, ***border-right-style***, ***border-bottom-style***, ***border-left-style***

Controlling border: color, and shorthand for properties

border-color: specifies the color of the border using RGB values, hex codes, or CSS color names. http://www.w3schools.com/css/css_border.asp
p.one { border-color: red; }

Can individually control the color of the borders on different sides of a box using one of these properties:
border-top-color, ***border-right-color***, ***border-bottom-color***, ***border-left-color***

Use a shorthand to control all four border colors using one property (***TRBL***)
p.three { border-color: red #bbbbaa blue #ee3e80; }

Use a shorthand to control the width, style, and color of a border
p.three { border: 3px solid blue; }

Padding

padding: specifies how much space should appear between the content of an element and its border. Values specified in **pixels**. Can use percentages or ems. http://www.w3schools.com/css/css_padding.asp

p.one { padding: 10px; }

If a ***width*** is specified for a box, ***padding*** is added onto the ***width*** of the box. **What other possibilities could there be and why?** (IE6 was different)

padding value **not** inherited by children elements. **Why do you think this is?**

Can specify different values for each side of a box using (***TRBL***):
padding-top, ***padding-right***, ***padding-bottom***, ***padding-left***

Use a shorthand to specify values for all four sides using one property (***TRBL***):

p.three { padding: 10px 5px 15px 10px; }

Margin

margin: specifies how much space should appear between boxes. Values specified in **pixels**. Can use percentages or ems. http://www.w3schools.com/css/css_margin.asp

p.one { margin: 10px; }

If a ***width*** is specified for a box, ***margin*** is added onto the ***width*** of the box. ***margin*** value **not** inherited by children elements. ***Larger of two margins is used when one box sits on top of the other--margins are collapsed.***

Can specify different values for each side of a box using (***TRBL***):
margin-top, ***margin-right***, ***margin-bottom***, ***margin-left***

Use a shorthand to specify values for all four sides using one property (***TRBL***):
p.three { margin: 10px 5px 15px 10px; }

Change inline/block display

display: allows you to display an inline element as if it were a block-level element and vise versa. Can also be used to hide an element.
http://www.w3schools.com/css/css_display_visibility.asp

https://www.w3schools.com/css/css_inline-block.asp

```
li {
        display: inline;                    // ideal for creating navigation
        margin-right: 10px;}
li.future {
            display: none;}
```

display values:

- ***inline***: causes a block-level element to act as an inline element
- ***block***: causes an inline element to act as a block-level element
- ***inline-block***: causes a block-level element to flow like an inline element while preserving other features of a block-level element (e.g., dimensions).
- ***none***: hides the element from the page

Hiding boxes

visibility: allows you to hide boxes from users, but it leaves a space where the element would have been.

```
li {
    display: inline;                // ideal for creating navigation
    margin-right: 10px;
}
li.future {
        visibility: hidden;
}
```

visibility values:

- ***hidden***: hides the element (but leaves a blank space in its place)
- ***visible***: shows the element

If you do not wish to show the blank space, use ***display*** instead of visibility

Flexbox and Grid layout

Flexbox layout: ideal for one dimensional layout (row OR column)

Explanation with examples:
https://www.w3schools.com/css/css3_flexbox.asp

Grid layout: ideal for two dimensional layout (row and column)

Explanation with examples:
https://scrimba.com/g/gR8PTE