Cascading Style Sheets (CSS)

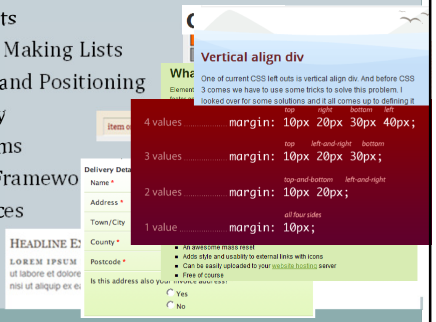
1

DEFINING OUR VIEW

What Web Designers Need to Know About CSS

2

- Page Layouts
- Styling and Making Lists
- Box model and Positioning
- Typography
- Styling Forms
- Using CSS Framewo
- Best Practices
- CSS Hacks



Vertical align div

One of current CSS left outs is vertical align div. And before CSS 3 comes we have to use some tricks to solve this problem. I looked over for some solutions and it all comes up to defining it

```
                    top   right bottom left
4 values ......margin: 10px 20px 30px 40px;
                    top  left-and-right bottom
3 values ......margin: 10px 20px 30px;
                    top-and-bottom left-and-right
2 values ......margin: 10px 20px;
                    all four sides
1 value .......margin: 10px;
```

- An awesome mass reset
- Adds style and usablity to external links with icons
- Can be easily uploaded to your website hosting server
- Free of course

Delivery Deta

Name *

Address *

Town/City

County *

Postcode *

Is this address also your invoice address?

Yes

No

HEADLINE E

LOREM IPSUM

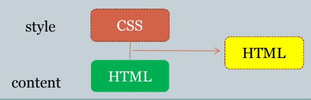
ut labore et dolore

nisi ut aliquip ex e

Style Sheets

3

- With HTML v4 (published 1998) a new approach to web page definition was developed using style sheets
- A style sheet is a set of defined presentation instructions that is separate from the content
- The concept of style sheets had been around since SGML days (1980s)



Style and Content Together

4

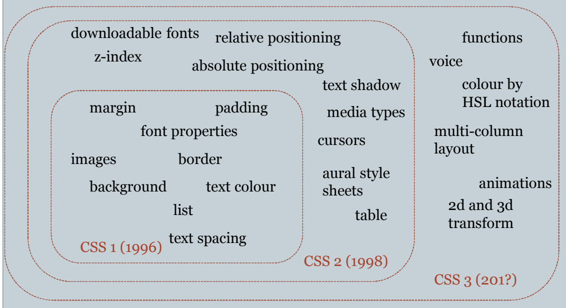
Origin of CSS

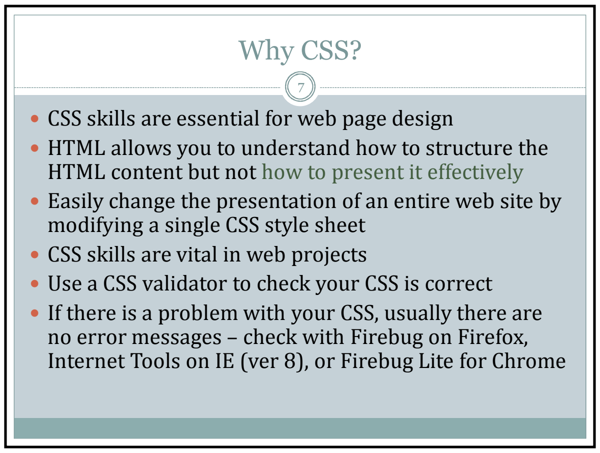
5

- W3C has produced three style recommendations CSS1, CSS2, and CSS3 – each level builds on the previous version - http://www.w3.org/TR/CSS21/
- Not all browsers implement these changes consistently
  - http://tools.css3.info/selectors-test/test.html
- Early browsers (MS Internet Explorer 3 and 4) did not fully support CSS
- As of July 2010, no browser has yet fully implemented CSS3 – some more than others
- http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(CSS)

Evolution of CSS Features from W3C

6

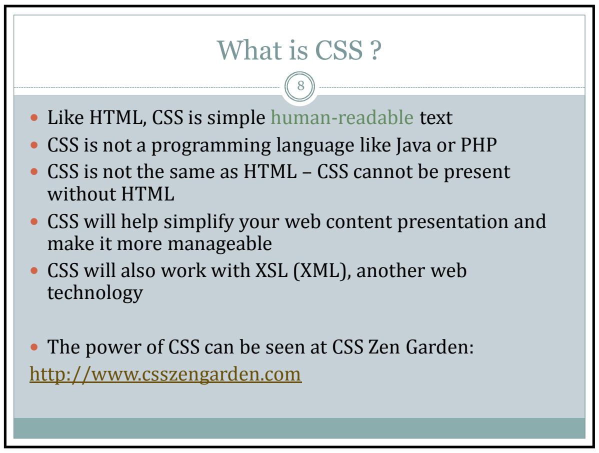




Why CSS?

7

- CSS skills are essential for web page design
- HTML allows you to understand how to structure the HTML content but not how to present it effectively
- Easily change the presentation of an entire web site by modifying a single CSS style sheet
- CSS skills are vital in web projects
- Use a CSS validator to check your CSS is correct
- If there is a problem with your CSS, usually there are no error messages – check with Firebug on Firefox, Internet Tools on IE (ver 8), or Firebug Lite for Chrome



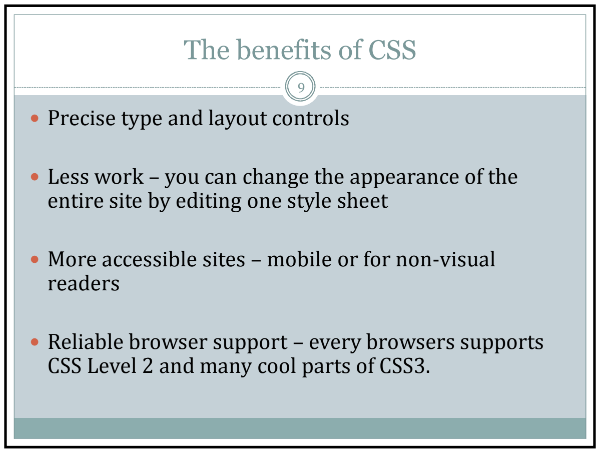
What is CSS ?

8

- Like HTML, CSS is simple human-readable text
- CSS is not a programming language like Java or PHP
- CSS is not the same as HTML – CSS cannot be present without HTML
- CSS will help simplify your web content presentation and make it more manageable
- CSS will also work with XSL (XML), another web technology

- The power of CSS can be seen at CSS Zen Garden:

http://www.csszengarden.com



The benefits of CSS

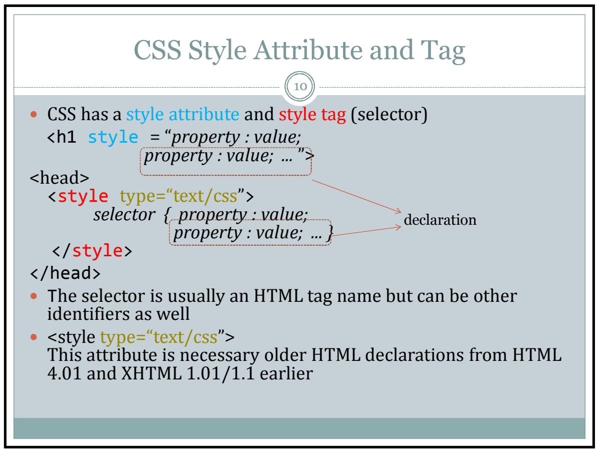
9

- Precise type and layout controls

- Less work – you can change the appearance of the entire site by editing one style sheet

- More accessible sites – mobile or for non-visual readers

- Reliable browser support – every browsers supports CSS Level 2 and many cool parts of CSS3.



CSS Style Attribute and Tag

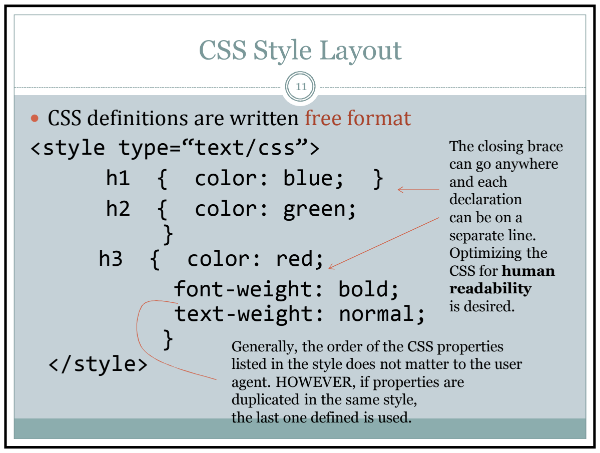
10

- CSS has a style attribute and style tag (selector)

```
<h1 style = “property : value;
              property : value; … ”>
<head>
  <style type=“text/css”>
        selector { property : value;
                   property : value; … }
  </style>
</head>
```

declaration

- The selector is usually an HTML tag name but can be other identifiers as well
- <style type=“text/css”>
  This attribute is necessary older HTML declarations from HTML 4.01 and XHTML 1.01/1.1 earlier



CSS Style Layout

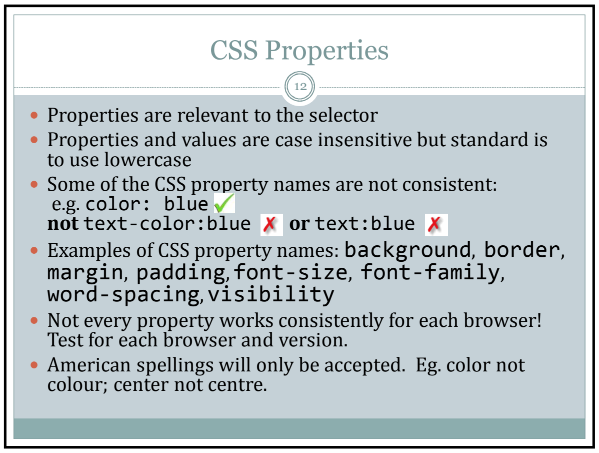
11

- CSS definitions are written free format

```
<style type=“text/css”>
      h1  {  color: blue;  }
      h2  {  color: green;
           }
     h3  {  color: red;
            font-weight: bold;
            text-weight: normal;
          }
  </style>
```

The closing brace can go anywhere and each declaration can be on a separate line. Optimizing the CSS for **human readability** is desired.

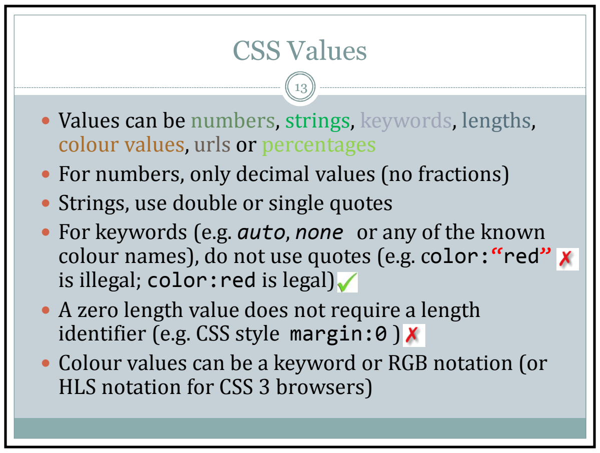
Generally, the order of the CSS properties listed in the style does not matter to the user agent. HOWEVER, if properties are duplicated in the same style, the last one defined is used.



CSS Properties

12

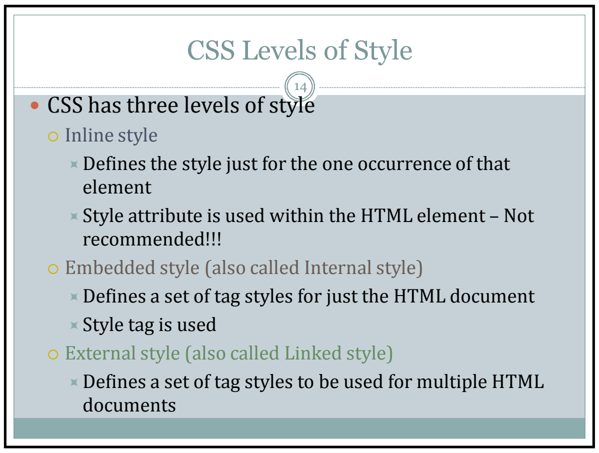
- Properties are relevant to the selector
- Properties and values are case insensitive but standard is to use lowercase
- Some of the CSS property names are not consistent:
  e.g. `color: blue` ✓
  not `text-color:blue` ✗ **or** `text:blue` ✗
- Examples of CSS property names: `background`, `border`, `margin`, `padding`, `font-size`, `font-family`, `word-spacing`, `visibility`
- Not every property works consistently for each browser! Test for each browser and version.
- American spellings will only be accepted. Eg. color not colour; center not centre.



CSS Values

13

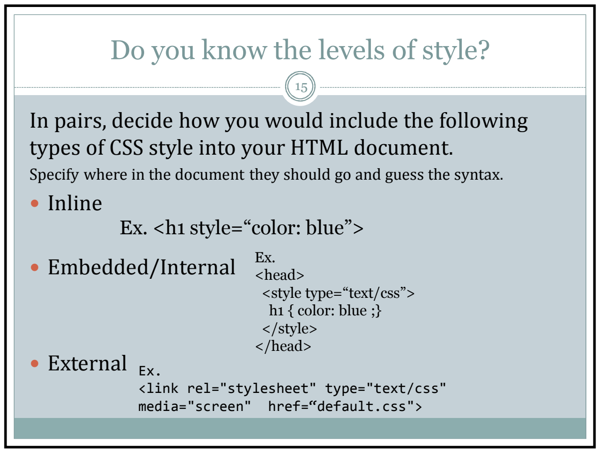
- Values can be numbers, strings, keywords, lengths, colour values, urls or percentages
- For numbers, only decimal values (no fractions)
- Strings, use double or single quotes
- For keywords (e.g. *`auto`*, *`none`* or any of the known colour names), do not use quotes (e.g. `color:“red”` ✗ is illegal; `color:red` is legal) ✓
- A zero length value does not require a length identifier (e.g. CSS style `margin:0`) ✗
- Colour values can be a keyword or RGB notation (or HLS notation for CSS 3 browsers)



CSS Levels of Style

14

- CSS has three levels of style
  - Inline style
    - Defines the style just for the one occurrence of that element
    - Style attribute is used within the HTML element – Not recommended!!!
  - Embedded style (also called Internal style)
    - Defines a set of tag styles for just the HTML document
    - Style tag is used
  - External style (also called Linked style)
    - Defines a set of tag styles to be used for multiple HTML documents



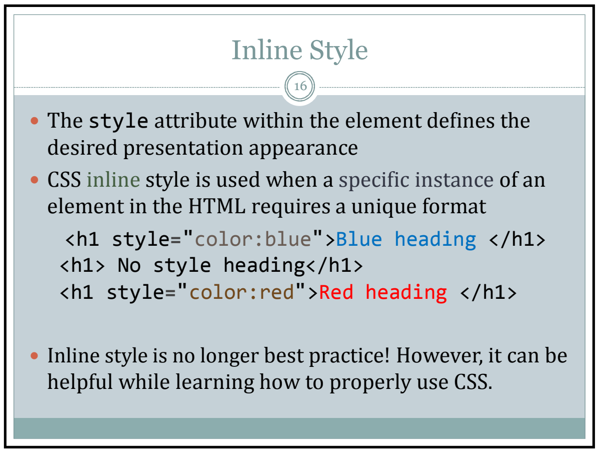
Do you know the levels of style?

15

In pairs, decide how you would include the following types of CSS style into your HTML document.

Specify where in the document they should go and guess the syntax.

- Inline
  Ex. <h1 style=“color: blue”>
- Embedded/Internal
  ```
  Ex.
  <head>
    <style type=“text/css”>
      h1 { color: blue ;}
    </style>
  </head>
  ```
- External
  ```
  Ex.
  <link rel="stylesheet" type="text/css"
  media="screen"  href=“default.css">
  ```



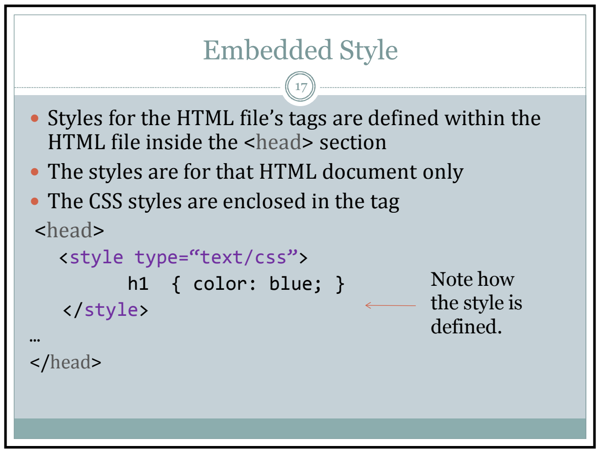
Inline Style

16

- The `style` attribute within the element defines the desired presentation appearance
- CSS inline style is used when a specific instance of an element in the HTML requires a unique format

  ```
  <h1 style="color:blue">Blue heading </h1>
  <h1> No style heading</h1>
  <h1 style="color:red">Red heading </h1>
  ```

- Inline style is no longer best practice! However, it can be helpful while learning how to properly use CSS.



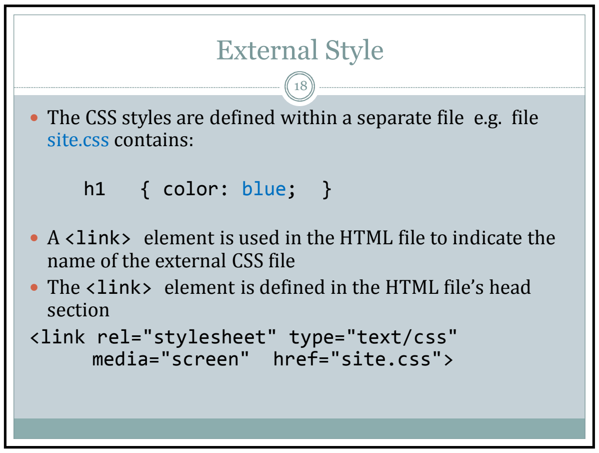
Embedded Style

17

- Styles for the HTML file’s tags are defined within the HTML file inside the <head> section
- The styles are for that HTML document only
- The CSS styles are enclosed in the tag

```
<head>
  <style type=“text/css”>
        h1  { color: blue; }
  </style>
…
</head>
```

Note how the style is defined.



External Style

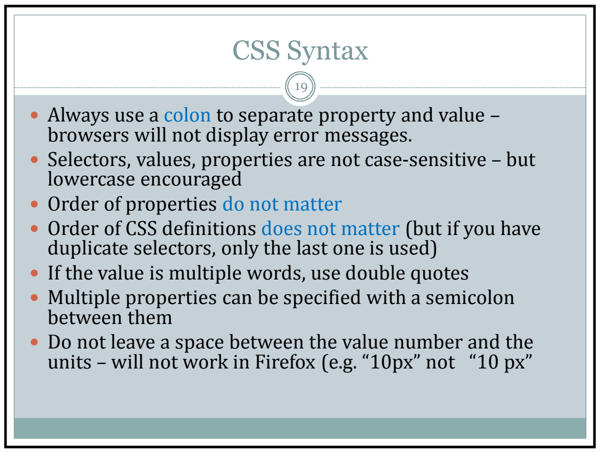
18

- The CSS styles are defined within a separate file e.g. file site.css contains:

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

- A `<link>` element is used in the HTML file to indicate the name of the external CSS file
- The `<link>` element is defined in the HTML file’s head section

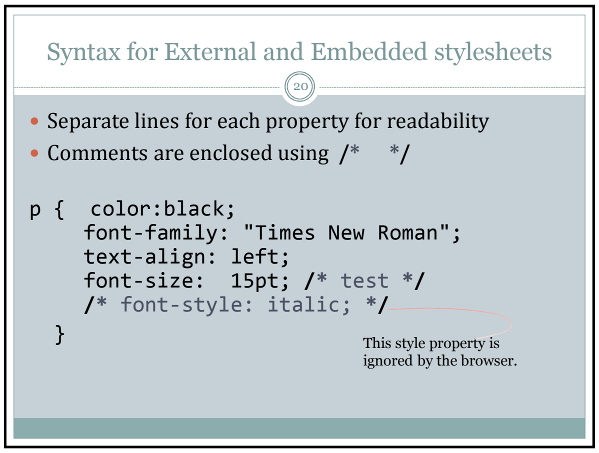
```
<link rel="stylesheet" type="text/css"
     media="screen"  href="site.css">
```



CSS Syntax

19

- Always use a colon to separate property and value – browsers will not display error messages.
- Selectors, values, properties are not case-sensitive – but lowercase encouraged
- Order of properties do not matter
- Order of CSS definitions does not matter (but if you have duplicate selectors, only the last one is used)
- If the value is multiple words, use double quotes
- Multiple properties can be specified with a semicolon between them
- Do not leave a space between the value number and the units – will not work in Firefox (e.g. “10px” not “10 px”



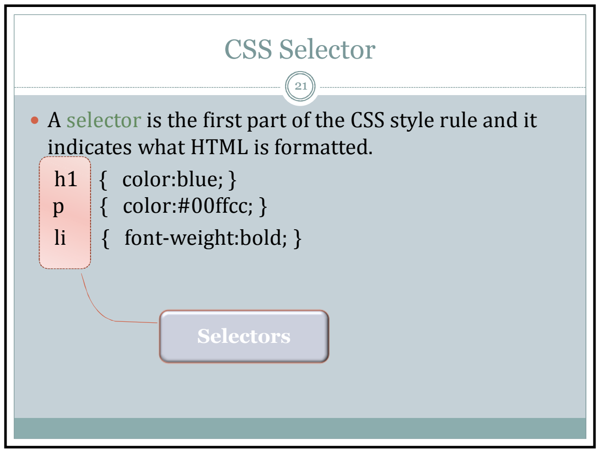
Syntax for External and Embedded stylesheets

20

- Separate lines for each property for readability
- Comments are enclosed using `/*   */`

```
p {  color:black;
    font-family: "Times New Roman";
    text-align: left;
    font-size:  15pt; /* test */
    /* font-style: italic; */
  }
```

This style property is ignored by the browser.



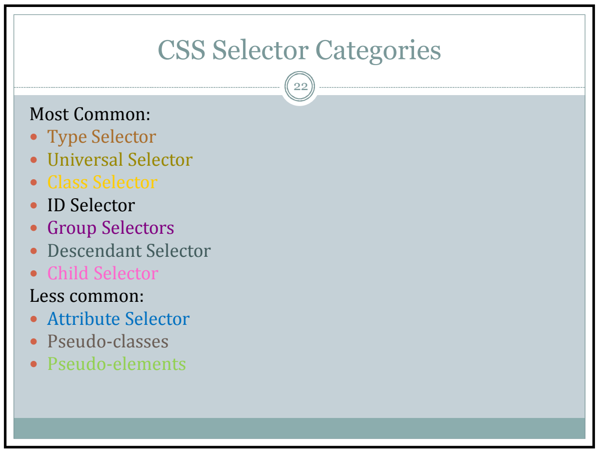
CSS Selector

21

- A selector is the first part of the CSS style rule and it indicates what HTML is formatted.

```
h1  {  color:blue; }
p   {  color:#00ffcc; }
li  {  font-weight:bold; }
```

Selectors

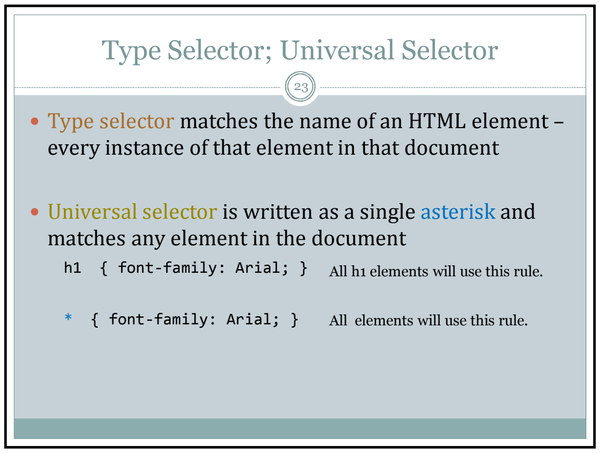


CSS Selector Categories

22

Most Common:
- Type Selector
- Universal Selector
- Class Selector
- ID Selector
- Group Selectors
- Descendant Selector
- Child Selector

Less common:
- Attribute Selector
- Pseudo-classes
- Pseudo-elements



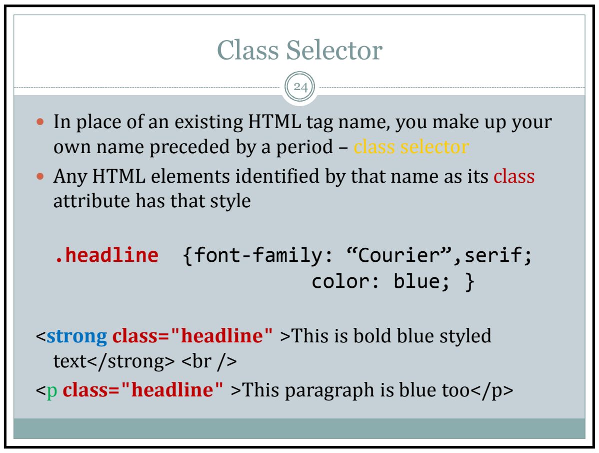
Type Selector; Universal Selector

23

- Type selector matches the name of an HTML element – every instance of that element in that document
- Universal selector is written as a single asterisk and matches any element in the document

```
h1  { font-family: Arial; }    All h1 elements will use this rule.

*  { font-family: Arial; }     All  elements will use this rule.
```



Class Selector

24

- In place of an existing HTML tag name, you make up your own name preceded by a period – class selector
- Any HTML elements identified by that name as its class attribute has that style

```
.headline   {font-family: “Courier”,serif;
                         color: blue; }
```

<strong class="headline" >This is bold blue styled text</strong> <br />
<p class="headline" >This paragraph is blue too</p>

ID Selector

25

- Identifies the style for a unique instance of the element defined by the ID name – ID selector

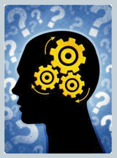
```
#menu { text-transform: uppercase; }

<div id="menu"> Text will show as uppercase.
</div>
```

Classes vs IDs

26

- Use classes for multiple occurrences of that style in a web page
  - Helpful mnemonic:
    - class has many students
    - ID – the letter I = one
- Use ID when there is only one occurrence of that style in a web page



Group Selectors

27

- A group of different selectors can share the same style definition – selectors are separated by a **comma**

```
h1, h2, h3 {  color : blue;  }

p, h1, h2 {  text-align: left; }

ol, ul {  margin-right: 20px; }
```

Descendant Selectors

28

- When a style rule needs to be applied to an element contained anywhere within another element, CSS descendant selectors are used

```
p strong { color: red; }
```

```
1 <p> Lorem ipsum
2 <strong> dolor sit amet, </strong> </p>
3 <div> < p > consectetur
4 <strong> adipiscing </strong>
5 </ p > </div>
6 <div><h1>Curabitur
7 <strong> ac. </strong>
8 </div></h1>
```

Which lines will be formatted in red?

Any <strong> elements defined within an p element will use this rule.
The <strong> element is the descendant of p element.

Dummy text:
http://www.lipsum.com/

Child Selectors

29

- When a style rule is applied to an element's child, a child selector is used

```
p > strong { color: blue; }
```

```
1 <p>This is
2 <strong>important</strong>
3 </p>
4 <div> <p> This is
5 <strong>important</strong>
6 </p>
7 </div>
8 <h1>This header is
9 <strong>not important</strong>
10 </h1>
```

Which lines will be blue?

Any strong elements defined within an p element *as child elements* (not simply a descendent) will use this rule.

Here the strong element is the child of p element.

Attribute Selector

30

- Attribute selectors match when the element sets an attribute in some way

img[title] { color: blue }	Matches elements <img title="*text*" ... />
img[title=start] { color: blue }	Matches elements <img title="start" ... />
img[title~="blue"] { color: blue }	Matches elements <img title="The blue hill" ... /> and <img title="blue rodeo" ... />

*[lang|="en"] { color: blue }

Vertical bar or "pipe"

Matches any elements having the attribute lang="en", or lang="en-US", or lang="en-CA"

Specifying class and ID selectors

31

- Class and ID style rules can be further specified using a type selector as prefix

```
h1.headline   { color: blue;     }
h2.headline   { color: green;    }
h3.headline   { color: red;      }
div#mainpar   { font: Arial;     }
p#logo        { font-size: 8pt;  }
```

Since the ID styles are unique anyway, the type selector prefix usage for them is superfluous and can marginally degrade rendering performance in the browser.

Pseudo-classes

32

- Pseudo-classes are similar to classes except that they refer to CSS specific types of text

e.g. :first-child matches the *first child* of an element

```
div > p:first-child { text-indent:20px; }
```

```
<div class="note">
  <p> Important </p>
  <p> Not important </p>
</div>
```

This style rule would apply only to this paragraph and not to this one.

Pseudo-class Links

33

a:link	{ color: blue; }	a link not yet clicked
a:visited	{ color: red; }	a link you clicked
a:hover	{ color: green; }	as you hover over
a:active	{ color: black; }	as you click link

LVHA -- definition of CSS style order required

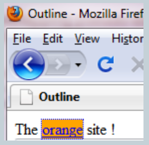
- The style `font-weight: 700;` makes the text thicker (bold) on hypertext link text

Focus Pseudo-class

34

- The :focus pseudo-class matches any element having keyboard input focus (e.g. form input or a link)
- Supported in IE 8, all versions of Firefox, Chrome

```
a:focus  { background: orange; }
```



Focus occurs only when user tabs to this link, not when the cursor is moved there.

This is for accessibility and for some javascript validations.

Pseudo-element Selectors

35

- The :first-letter pseudo-element selects the first *letter (or digit)* of the first line of a text block.

-
  ```
  p:first-letter { font-size: 3em; float:left;
                   font-weight:bold; }
  ```

<p>Mary had a little lamb...etc</p>

What do you think this would look like?

M ary had a little lamb, little lamb, little lamb. Mary had a little lamb and

Selector Family Tree

36

- Descendant Selector (the child of the parent)
  `div p` (**p** is the child of parent **div**)
- Grandchildren Selector
  - Use the universal selector *

  `div * p` (match all **paragraphs** that are grandchildren or greater of **div** elements)
- Child Selector >
  `body > p` (match only children **p** elements of **body**)
- Adjacent Sibling Selector +
  `em + p` (match any **p** tags that follow after **em** element)
- Attribute Selector [ ]
  `a[title]` (match any **a** tags having a defined **title** attribute)

CSS Rule of Proximity

37

- The closer style rule will be applied to text
  - **Inline** style **overrides embedded** style
  - In the case of **nested** tags (like `div`), the style rules of the innermost tags override the style rules of the outer tags

```
<div style="color:yellow;"> This is yellow
  <div style="color:green;"> This is
green.
    <div style="color:blue;"> This is blue.
   </div>  </div>  </div>
```

CSS Rule of Inheritance

38

- Many CSS styles are automatically passed into any child elements from the parent element.
  - font and color styles defined for a parent element will be 'inherited' by any contained child elements by default
  - When inheritance occurs, elements inherit computed values based on the parent and child

```
body  { font-size: 10pt  }
h1    { font-size: 120% }  What is going to be the font-size ?

      <body>
        <h1> Large Heading </h1>
        ..
      </body>
```

The computed value of this text will be 10pt x 120% = 12pt.

Cascading

39

Cascading order

- HTML elements may have different styles applied to them – which one is used by the browser?
- All styles "cascade" into a single style sheet *in part* by the following **proximity** rule
  - Browser default style -- lowest priority
  - External style sheet (a .css file)
  - Embedded style sheet (inside <head> section)
  - Inline style (inside an HTML tag) – highest priority

!important rules

40

- You can override any style rule with the !important declaration

```
p  { text-index: 1em !important;
     font-style: italic !important;
     font-size: 16pt  !important   }
```

If these style rules are defined in the user's style sheet, they will override similar style rules in the author's style sheet – even if the author used !important as well.

CSS Values - Examples

41

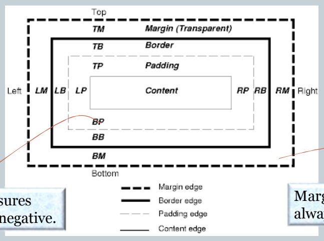
```
<style type="text/css">

  p  {  background-color: gray;             ← colour keyword
        background-position: 40% 50%;       ← percentages
        background-size:  10em 10em;        ← lengths
        background-image: url(blue.png);    ← URL
        background-repeat: repeat-x;        ← keywords
        border-style: solid;                ← keywords
        border-color: #FF30A0;              ← Colour in RGB format
        width: 50px;                        ← length
        font-family: 'Courier New Times' ;  ← CSS string
     }
</style>
```

CSS Box properties

42

- CSS box model defines properties such as margin, padding, border, background, position.



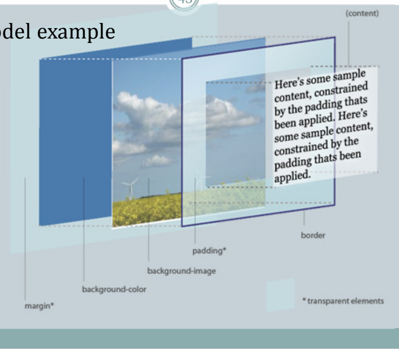
Padding measures must be non-negative.

Margins are always transparent.

CSS Box model in 3D

43

- Box model example

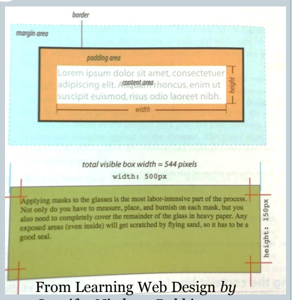

(content)
Here's some sample content, constrained by the padding thats been applied. Here's some sample content, constrained by the padding thats been applied.
border
padding*
background-image
background-color
margin*
* transparent elements

Box Examples

44

p {
 background: #c2f670;
 width: 500px;
 height: 150px;
 padding: 20px;
 border: 2px solid gray;
 margin: 20px; }

What's the total visible box width?


border
margin area
padding area
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam rhoncus, enim ut suscipit euismod, risus odio laoreet nibh.
content area
height
width
total visible box width = 544 pixels
width: 500px
Applying masks to the glasses is the most labor-intensive part of the process. Not only do you have to measure, place, and burnish on each mask, but you also need to completely cover the remainder of the glass in heavy paper. Any exposed areas (even inside) will get scratched by flying sand, so it has to be a good seal.
height: 150px

From Learning Web Design *by Jennifer Niederst Robbins*:

IE Quirks Mode

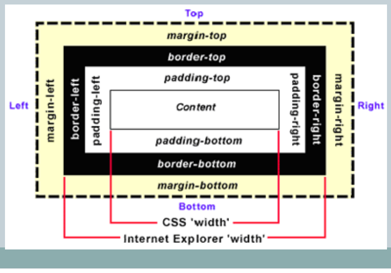
45

- Prior to version 6, IE browsers used a different method for determining the width of an element's box than the method used by the W3C CSS
- Many web sites had already used the non-standard Microsoft implementation of width
- Quirks mode refers to a technique used by a browser to make an earlier, non-standard W3C web page compatible for viewing
- Any later version of IE can be "flipped" into quirks mode if the HTML has a missing Document Type Declaration (the DOCTYPE)

IE box model – quirks mode

46

- In quirks mode IE browsers understand the box model width property their own way:



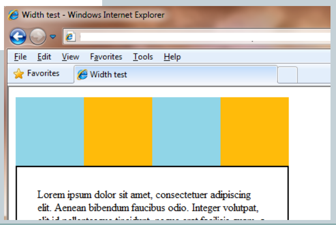
Quirks Mode example

47

- In IE an incorrect or missing DOCTYPE triggers quirks mode

```
<html>
<!-- Missing a fully qualified DOCTYPE so this HTML
     will be rendered in quirks mode in IE browsers.
 -->
<head>
<title>Width test</title>
<style type="text/css">
```

```
#w1 {
      width:400px;
      padding:30px;
      margin:0px;
      border:solid black thin;
      float:left;
    }
```



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean bibendum faucibus odio. Integer volutpat,

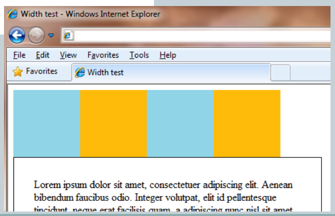
Standards Mode example

48

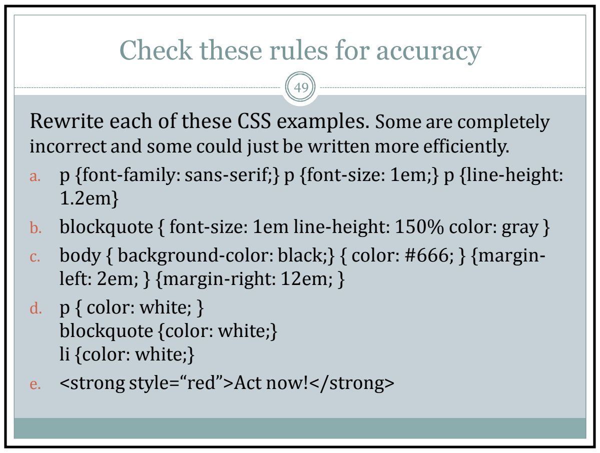
- IE v8 uses standard box model width if a valid DOCTYPE is defined in the HTML

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<head>
<title>Width test</title>
<style type="text/css">
```

```
#w1 {
      width:400px;
      padding:30px;
      margin:0px;
      border:solid black thin;
      float:left;
    }
```



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean bibendum faucibus odio. Integer volutpat, elit id pellentesque

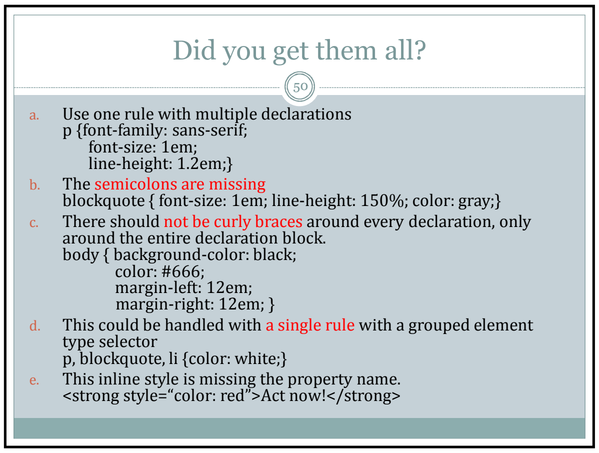


Check these rules for accuracy

49

Rewrite each of these CSS examples. Some are completely incorrect and some could just be written more efficiently.

a. p {font-family: sans-serif;} p {font-size: 1em;} p {line-height: 1.2em}
b. blockquote { font-size: 1em line-height: 150% color: gray }
c. body { background-color: black;} { color: #666; } {margin-left: 2em; } {margin-right: 12em; }
d. p { color: white; }
   blockquote {color: white;}
   li {color: white;}
e. <strong style="red">Act now!</strong>



Did you get them all?

50

a. Use one rule with multiple declarations
   p {font-family: sans-serif;
   font-size: 1em;
   line-height: 1.2em;}
b. The semicolons are missing
   blockquote { font-size: 1em; line-height: 150%; color: gray;}
c. There should not be curly braces around every declaration, only around the entire declaration block.
   body { background-color: black;
   color: #666;
   margin-left: 12em;
   margin-right: 12em; }
d. This could be handled with a single rule with a grouped element type selector
   p, blockquote, li {color: white;}
e. This inline style is missing the property name.
   <strong style="color: red">Act now!</strong>