CSS

Part I

Introduction

- CSS: Cascading Style Sheets
- Created by W3C
- Describes how HTML elements are to be displayed
- 3 ways:
  - Internal
  - External
  - Inline

Advantages of using CSS

- The presentation of the website can be centralized
- Users can compose style sheet of their own for the website
- It is possible for users to select the CSS that suit their look and feel
- Style sheets allow content to be optimized for more than 1 type of device
- Using external CSS make the document size smaller

Inline CSS

- Style attribute

content

Internal css

- Defined inside <script> element

```
<head>
        <style>
                css
        </style>
</head>
```

External CSS

- File with .css extension

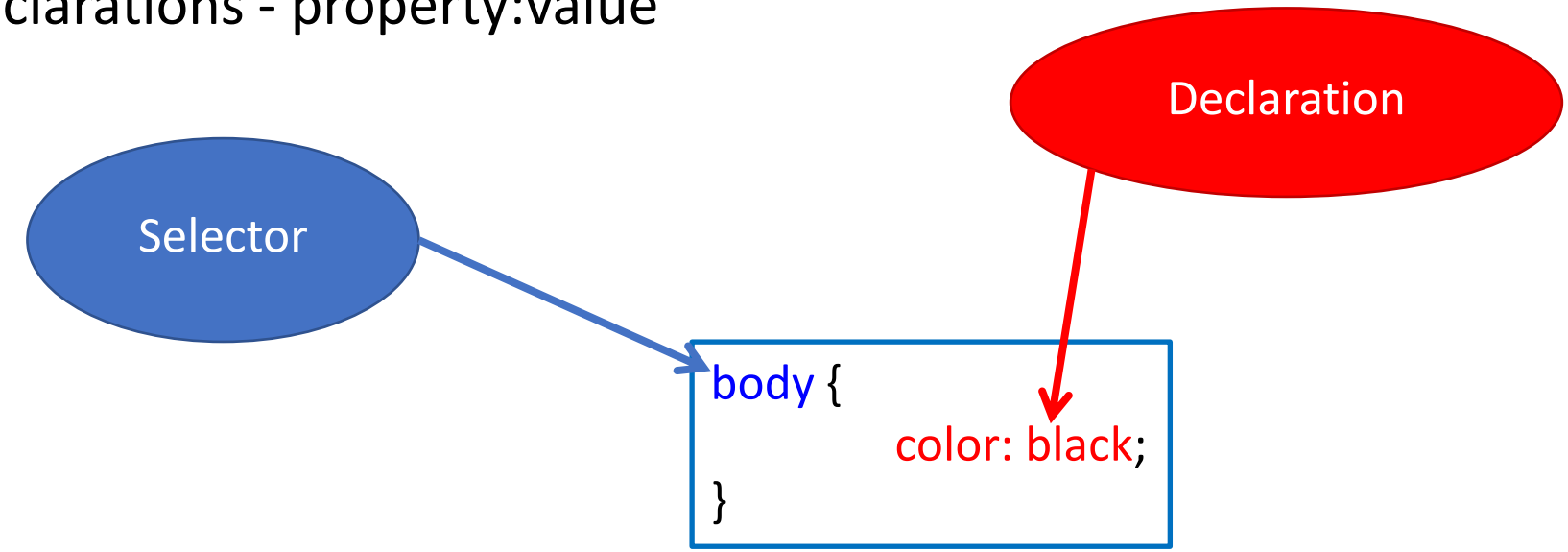
```
<head>
        <link rel="stylesheet" type="text/css" href="xstyle.css">
        or
        <link rel="stylesheet" href="xstyle.css">
        or
        @import url(xstyle.css);
</head>
```

CSS order

- Inline
- Internal and external
- Browser default

CSS Rules and syntax

- CSS can use white space and line break for purposes of readability
- Comment text is put inside /* */
- CSS is composed of 2 parts
  - Type selector or Selector in short
  - Declarations - property:value
- Ex:



Keywords

- Used as property values
- Example: red, green, blue, auto, …
- Some keywords (e.g. auto) can have different meaning depending on the element to which it is applied

Data types

- Strings
- Integers and real numbers

Length and measurement options

- 3 kinds:
  - Absolute
    - in = inches
    - cm = centimeters
    - mm = millimeters
    - Pt = points => 1 point = 1/72 inch
    - Pc = picas => 1 picas = 12 points
  - Relative
    - em = length relates to font-size
      - ex: font-size:10px, 2em = 20px
    - ex = length relates to font-height
    - px = pixels (length relative to viewing device)
  - Percentage
    - Ex: width:100%

Colors

- CSS supports a number of options for specifying color:

Colors

- Color keywords

indianRed	orange	mediumPurple	forestGreen	cadetBlue	wheat	whiteSmoke
lightCoral	gold	blueViolet	green	steelBlue	burlyWood	seashell
salmon	yellow	darkViolet	darkgreen	lightSteelBlue	tan	beige
darkSalmon	lightYellow	darkOrchid	yellowGreen	powderBlue	rosyBrown	oldLace
lightSalmon	lemonChiffon	darkMagenta	oliveDrab	lightBlue	sandyBrown	floralWhite
red	lightGoldenRodYellow	purple	olive	skyBlue	goldenRod	ivory
crimson	papayaWhip	indigo	darkOliveGreen	lightSkyBlue	darkGoldenrod	antiqueWhite
firebrick	moccasin	darkSlateBlue	mediumAquamarine	deepSkyBlue	peru	linen
darkRed	peachPuff	slateBlue	darkSeaGreen	dodgerBlue	chocolate	lavenderBlush
pink	paleGoldenRod	mediumSlateBlue	lightSeaGreen	cornflowerBlue	saddleBrown	mistyRose
lightPink	khaki	greenYellow	darkCyan	royalBlue	sienna	gainsboro
hotPink	darkKhaki	chartreuse	teal	blue	brown	lightGray
deepPink	lavender	lawnGreen	aqua	mediumBlue	maroon	silver
mediumVioletRed	thistle	lime	cyan	darkBlue	white	darkGray
paleVioletRed	plum	limeGreen	lightCyan	navy	snow	gray
lightSalmon	violet	paleGreen	paleTurquoise	midnightBlue	honeydew	dimGray
coral	orchid	lightGreen	aquamarine	cornsilk	mintCream	lightSlateGray
tomato	fuchsia	springGreen	turquoise	blanchedAlmond	azure	slateGray
orangeRed	magenta	mediumSeaGreen	mediumTurquoise	bisque	aliceBlue	darkSlateGray
darkOrange	mediumOrchid	seaGreen	darkTurquoise	navajoWhite	ghostWhite	black

Colors

- RGB values
- RGB percentage
- RGBA (RGB with Alpha channel: CSS 3)
- Hexadecimal – 3 pairs of hex number to represent RGB
- Shorthand Hexadecimal (limited to 216 colors) – defines method to simplify Hex number from 6 hex numbers to 3 hex numbers by using only pair of the same number

Color examples

Color Keywords

```
div {       color: black;
            background-color: red;
            border: thin solid orange;
}
```

RGB Colors

Body {background-color: rgb(128, 128, 128);}

Equal amounts of 3 channels form variation of gray
0,0,0 is black and 255,255,255 is white

RGB values can also be represented using percentage

body {background-color: rgb(50%, 50%, 50%);}

Hexadecimal Colors

```
div {       color: #000000;
            background-color: #FF0000;
            border: thin solid #FFA500;
}
```

Short Hexadecimal Colors

```
div {       color: #000;
            background-color: #F00;
            border: thin solid #FA5;
}
```

Selector

- Address the target elements to be CSS-formatted
- May be HTML tag
- May be user-specified via class or id attributes
- Can be grouping
  - Ex:

```
h1, h2, h3, h4, h5
{
        font-family: Arial;
        color: black;
}
```

CSS Selector

- The universal (wild card) selector
- Contextual/descendant selectors
- Child selectors
- Direct/indirect adjacent sibling combinators
- Attribute selectors
- User-defined class and id selector

The universal selector

- CSS 3
- The universal selector is an asterisk (*)
- When use alone, it tell CSS interpreter to apply the CSS rule to the entire document
- Ex:

```
* {font-family: Arial; color: black; }
```

Contextual/descendant selectors

- CSS 3
- In CSS 1 descendant selectors are referred to as contextual
- Apply style based on whether one element is a descendant of another

Example:

```
<body>
        <h1>this header1 is outside div</h1>
        <div>
                <h1>this h1 is inside div</h1>
                <h1>this h1 is also inside div</h1>
                <p>this is not h1</p>
        </div>
        <h1>outside div</h1>
        <div>
                <table>
                        <tr>
                                <td>
                                        <h1> Some header text </h1>
                                </td>
                        </tr>
                </table>
        </div>
</body>
```

```
div h1 {color: darkolivegreen;}
```

Example:

```
<body>
        <h1>this header1 is outside div</h1>
        <div>
                <h1>this h1 is inside div</h1>
                <h1>this h1 is also inside div</h1>
                <p>this is not h1</p>
        </div>
        <h1>outside div</h1>
        <div>
                <table>
                        <tr>
                                <td>
                                        <h1> Some header text </h1>
                                </td>
                        </tr>
                </table>
        </div>
</body>
```

```
div td h1 {color: darkolivegreen;}
```

```
div table h1 {color: darkolivegreen;}
```

```
div table td h1 {color: darkolivegreen;}
```

Universal – Descendant Combination

- Universal selector can be combined with other selectors
- Ex:

```
div * {color:lavender;}
```

Direct child selectors

- CSS 3
- Quite similar to descendant selector
- Apply style only to immediate children of the element
- Ex:

h2>em {color: blue;}

Direct Adjacent Sibling Combinator

- CSS 3
- Select based on whether two element appear side by side in a document as a sibling
- Ex:

```
h2 + p {color: red;}
```

Example

```
h2 + p {color: red;}
```

```
<body>
  <div>
    <h2>Welcome to CSS widgets.</h2>
    <p>This paragraph of text is indented 20 pixels.</p>
  </div>
  <p>This paragraph of text is not indented; it does not have the same
      parent as an h2 element.
  </p>
</body>
```

Indirect Adjacent Sibling Combinator

- CSS 3
- Select based on sibling relationship like direct adjacent sibling combinator
- Do not require that the element appear side by side
- The element must share the same parent element
- Use a tilde (~)

Example

```
h2 ~ h3 {color:lightgreen;}
```

```
<body>
        <div>
                <h2>Welcome to CSS widgets.</h2>
                <p>This paragraph of text is indented 20 pixels.</p>
                <h3>Some underlined text.</h3>
        </div>
</body>
```

Example

```
h2 ~ h3 {color:lightgreen;}
```

```
<body>
        <div>
                <h2>Indirect Adjacent</h2>
                <h3>Welcome to CSS widgets.</h3>
                <p>This paragraph of text is indented 20 pixels.</p>
                <h3>Some underlined text.</h3>
        </div>
</body>
```

Attribute Selector

- CSS 3
- Apply style declaration based on the presence of attributes or attribute values that appear in the tag
- Ex:

```
input[type] {background: green;}
```

Attribute Selector

- Selection based on the value of attribute
  - input[type=“text”] { …. }
  - input[type=“text”][name=“some name”] { … }
  - Input[type~=“text” “file” “password”] { … }
- Attribute substring selectors
  - Select attribute values that begin with a string
    - a[href^=“http://”] { … }
  - Select attribute values that end with a string
    - a[href$=“.html”] { … }
  - Select attribute values the contain a string
    - a[href*=“.ac”] { … }

Class and ID Selectors

- CSS 3
- Select based on ‘class’ or ‘id’ attribute
- class selector
  - html tag: some content
  - class selector: .aclass { … }
- id selector
  - html tag: some content
  - id selector: #someid { … }
- Class and ID selector can be made more specific
  - div.aclass { … }
  - div#someid { … }