FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Web Design Fundamentals / Basic Web Design

Cascading Style Sheets (CSS), DIV, SPAN – ITEWEBDES / ITWD113

CASCADING STYLE SHEETS (CSS)

A style language that lets you control the display characteristics of your Web site.

<font> VS CSS

<font> style element was introduced by browser developers to help Web designers bypass the design limitations of HTML but tailored only towards one type of display medium, the computer screen.

A **style sheet** can display information for different devices.

What are they?

- A set of style rules that tell the web browser how to present a web page or document.
- Styles are normally stored in style-sheets
- External style sheets are stored in .css files
- Multiple style sheets will cascade into one

Reasons for Using CSS

1. Build from the ground up to replace traditional Web design methods
2. Faster download times
3. Shorter development time
4. Greater control over the typography in a Web page

Reasons for Using CSS

5. It's easy to write
6. Improvements in accessibility
7. Print designs as well as Web page designs
8. Better control over the placement of elements in Web page

Reasons for Using CSS

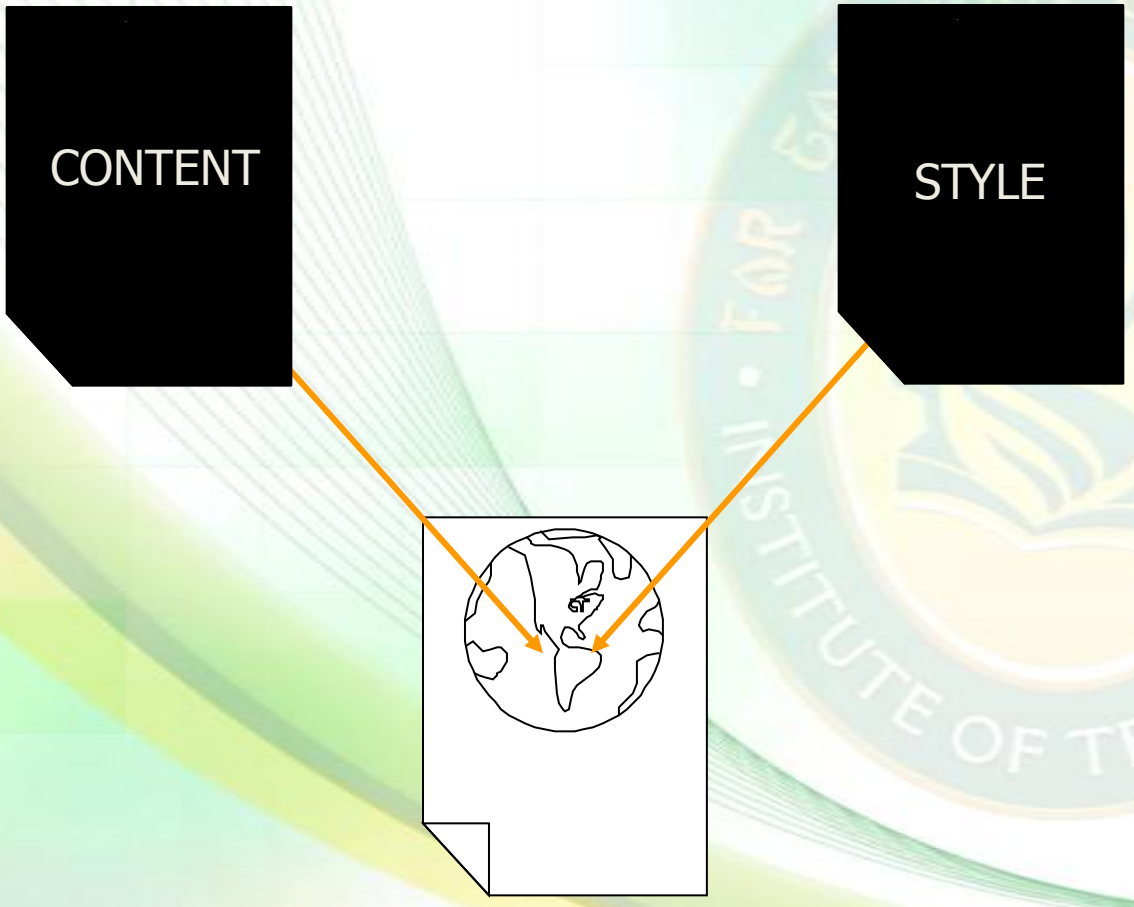
9. The design of Web pages is separated from the content
10. Better search engine rankings

Evolution of CSS

CSS Version	Description
CSS 1	The first version released in December 1997
CSS 2	Released in May 1998
CSS 2.1	All major browser supports CSS, released in 2005
CSS level 3 (CSS3)	Work started in 1998. This is the latest release of CSS

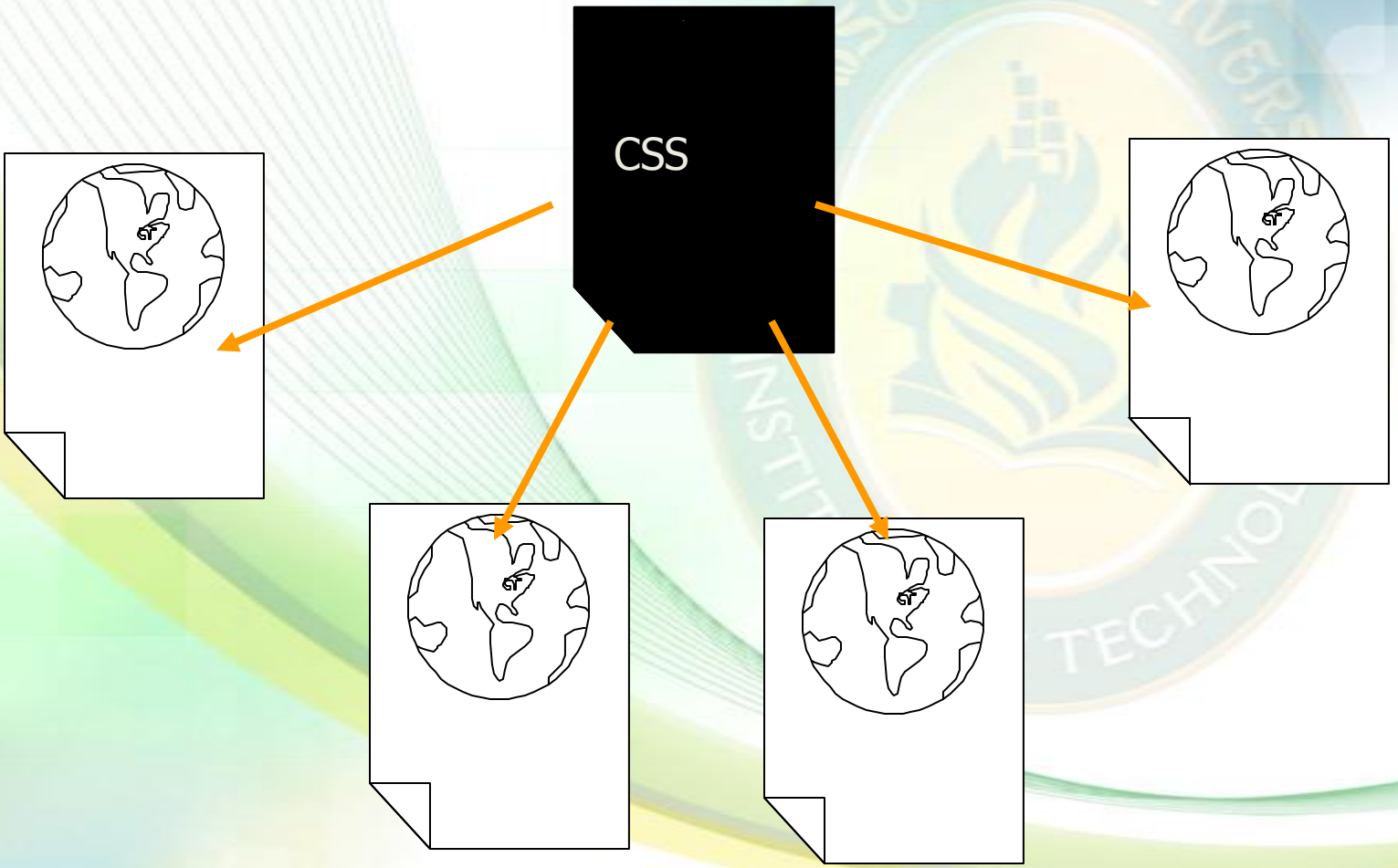
Note that in some cases minimal difference in browser view will be noticed.

What are they?

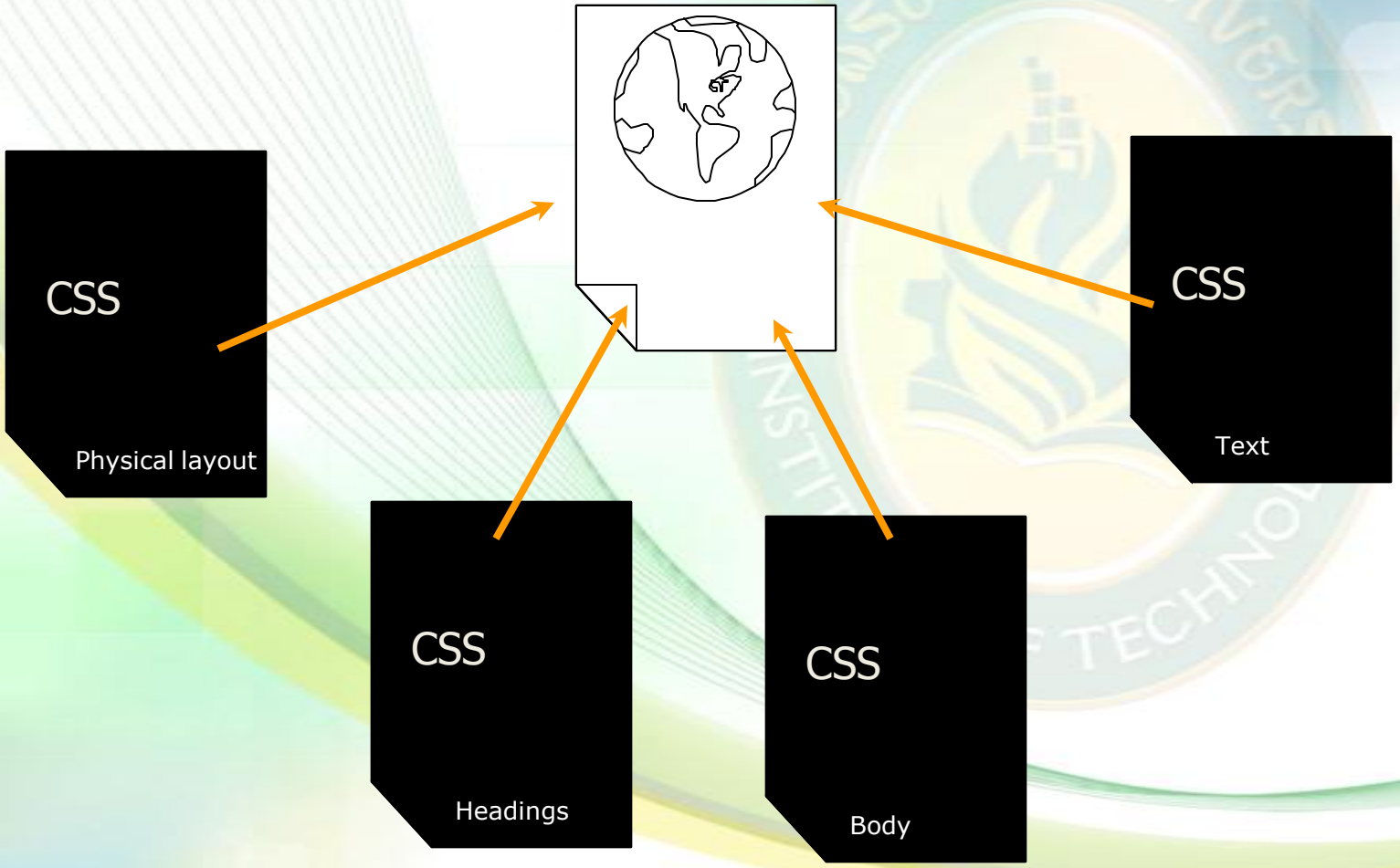


What are they?

What are they?



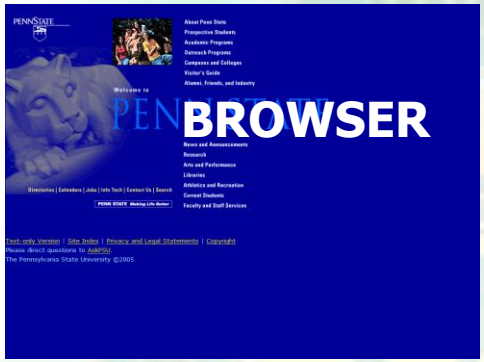
What are they?

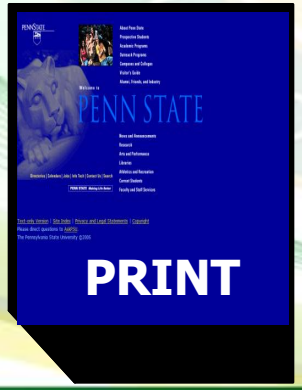


Devices

Content

CSS





Types of CSS

- Inline style
- Internal style sheet
- External style sheet and Imported

Inline Style

- Least flexible
- Requires each element to be tagged if you want them to appear differently
- Looses the advantage of using CSS

```
<h1 ALIGN="center" STYLE="background: #000080; font: 36pt/40pt courier; font-variant: small-caps; border: thick dashed blue">
Welcome to my home page!
</h1>
```

Internal / Embedded style sheet

- Style characteristics are embedded in the HEAD section of the webpage
- Perhaps best used when a single page requires a unique style sheet

```
<head>
    <style>
    hr {color:sienna;}
    p {margin-left:20px;}
    body {background-image:url("images/back40.gif");}
    </style>
</head>
```

External style sheet

- Connection made via the LINK tag
- Use the optional TYPE attribute to specify a media type
  - type/css

Index.htm

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

Filename.CSS

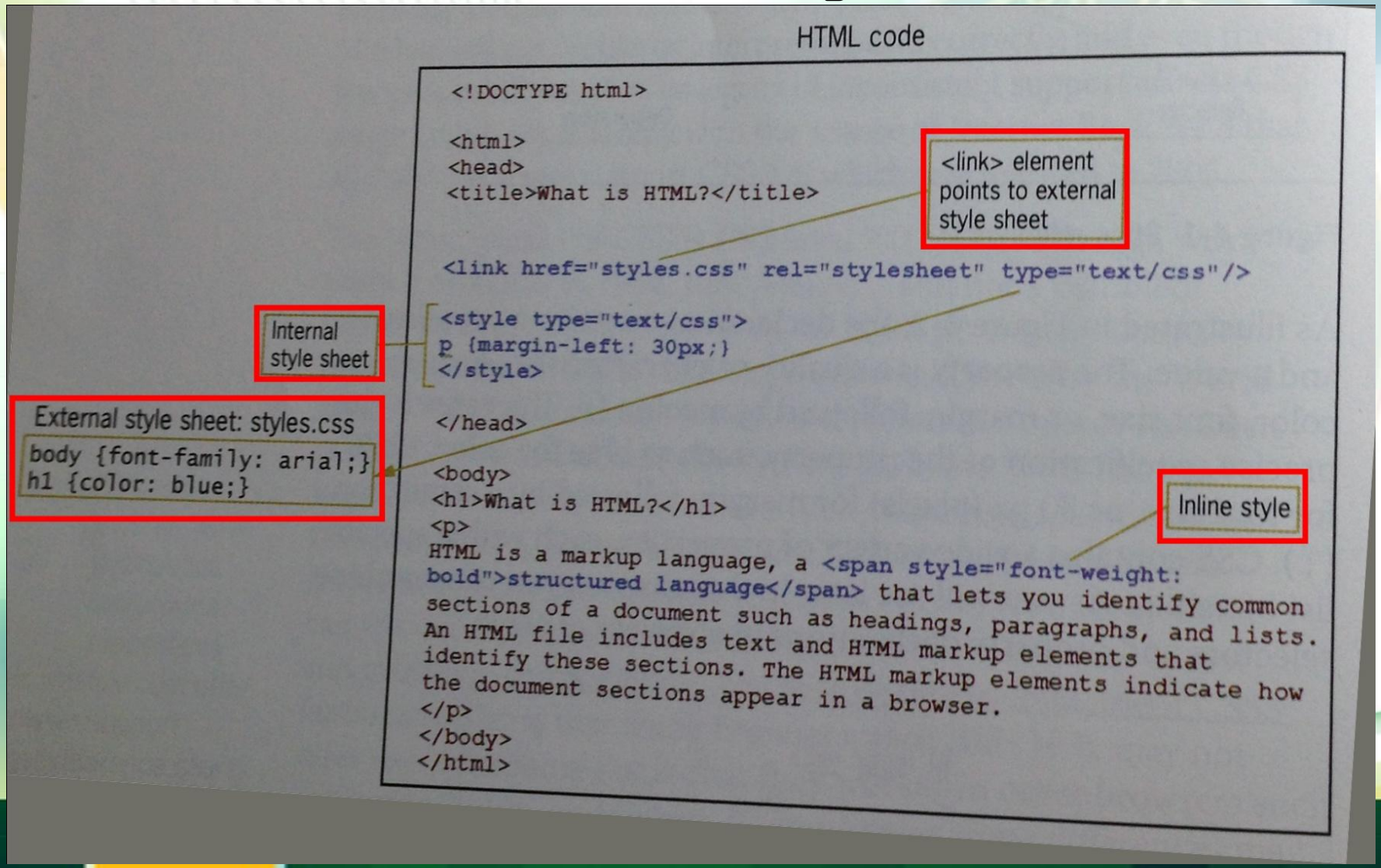
```
hr {color:sienna;        }
p {margin-left:20px;     }
body {background-
image:url("images/back40.gif");
}
```

Imported

- Allows for using style sheets from other sources
- Must be included at the beginning of the style sheet using the @import statement
- Other CSS rules can be included

```
<style type="text/css">
@import url("import3.css");    p { color : #f00; }
</style>
```

Methods of Combining CSS with HTML



Inserting a CSS

Inline

```
<p style="color: yellow; font-family:
  verdana">
This is a paragraph
</p>
```

Inserting a CSS

Internal/embedded sheet

```
<head>

<style type="text/css">

hr { color: navy;}
body {margin-left: 20px;}
</style>

</head>
```

Inserting a CSS

External sheet

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

Cascading multiple sheets

- You can use multiple sheets to define the style of your document
- Internal styles will override external styles, if they are duplicated

Cascading multiple sheets

h3 {color: red; text-align: right; font-size: 8pt} *(external CSS)*

h3 {text-align: center; font-size: 20pt} *(internal CSS)*

will yield

h3 {color: red; text-align: center; font-size: 20pt }

Sheet weight or Precedence

Understanding the Cascade

- Cascading
  - **Determining rule weight by specificity**
    - Rules with more specific selectors take precedence over rules with less specific selectors
  - **Determining rule weight by order**
    - Based on order of rule within style sheet
      - Those listed later take precedence over those listed earlier in the style sheet

Understanding the Cascade

- Inheritance
  - Based on hierarchical structure of documents
    - CSS rules inherit from parent elements to child elements:
      - Thus <LI> elements will inherit style rules from <UL> elements unless a style rule is specifically set for the <LI> element

FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Basic CSS Syntax

Basic CSS Syntax

- Three parts:
  - selector
  - property
  - value

} *declaration*

selector {property: value}

Basic CSS Syntax

selector {property: value}

selector: the basic HTML element tag you wish to define

body

property: the attribute of the selector that you wish to change

body {color

value: the particular markup value for that attribute

body {color : black}

Properties with multiple words

If the value has multiple words, put the value in quotes

p {font-family: “sans serif” }

Multiple properties

You can specify multiple properties to a single selector. Properties must be separated by a semicolon.

```
P { text-align: left; color: red }
```

Basic CSS Syntax

To make properties more readable, put each on a separate line.

```
p { text-align: center;
    color: navy;
    font-family: arial
    }
```

Grouping

Selectors can be grouped so that a common property can be specified

```
h1,h2,h3,h4,h5,h6
{ color: red}
```

```
<h1> This is a level 1 heading </h1>
<h2> This is a level 2 heading </h2>
```

Descendants

Selectors can be descendants

P B { color: yellow }

In this example, only those <B> elements within a <P> element would be yellow

 This would be yellow
 This would not be yellow

CSS Syntax - class

The class selector allows you to create different styles for the same HTML element.

```
p.right { text-align: right; }
p.center { text-align: center; }
```

CSS Syntax - class

```
p.right { text-align: right }
```

```
<p class="right">
        This paragraph will be right aligned.
</p>
```

Note: the class name must be in quotes inside the opening tag

CSS Syntax - class

This is an improper use of the class selector:

```
<p class="right" class="center">
        This paragraph will be right aligned.
</p>
```

Only one class selector can be included inside the tag

CSS Syntax - class

```
<p class="center bold">
        This is a paragraph.
</p>
```

Note: the paragraph will be styled by the class "center" AND the class "bold"

CSS Syntax - class

You can also create a class selector free of a tag name if you want all tags that have that class to be formatted the same.

```
.center { text-align: center }
```

Any tag with a “center” class will be aligned center

CSS Syntax - class

```
.center { text-align: center }

<h1 class="center">
        This heading will be centered
</h1>

<p class="center">
                So will this text
</p>
```

CSS Syntax - id

While the class selector can apply to several different elements, the id selector can only apply to one, unique element.

```
p#para1 { text-align: center;
          color: green }
```

Apply style to p element with id="para1"

CSS Syntax - id

```
p#para1 { text-align: center;
          color: green }
```

```
<p id="para1">
   This text would be centered and green
</p>
```

CSS Syntax - comment

You can insert comments to help you describe the particular style

Comments open with /* and are closed with */

```
/* This is a comment */
P { color: red;
/* This is another comment */
Font-family: verdana }
```

Background Properties

Background properties

- Define the background effects of an element
- Effects include color, using an image for a background, repeating an image and positioning an image

Background properties

- Basic syntax
  - background
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position

Background properties

- All attributes can be set in a single declaration:

```
background: #000000 url(asd.gif) no-repeat fixed center
```

Background properties

- Setting the body background (internal CSS)

```
body { background: #000000
   url(asd.gif) no-repeat fixed center }
```

Background properties

- Setting the body background (external CSS)

```
body: { background: #000000
   url(asd.gif) no-repeat fixed center }
```

Background properties

- Elements can also be set separately

```
body
{ background-image: url(asd.gif);
background-color: navy }
```

Text Properties

Text properties

- Controls the appearance of text in the web page

Text properties

- Commonly used attributes
  - color
  - direction
  - text-align
  - text-decoration
  - text-indent

Text properties

- color
  - sets the color of the text
  - color can be represented by the color name (red), an rgb value (rgb(255,0,0)), or by a hexadecimal number (#ff0000)
- Syntax
  - body {color: #ff0000}

Text properties

- direction
  - sets the direction of the text
  - can be set as left to right (ltr) or right to left (rtl)
- Syntax
  - body {direction: rtl}

Text properties

- text-align
  - aligns the text in an element
  - possible values are left, right, center and justify
- Syntax
  - p {text-align: center}

Text properties

- text-decoration
  - adds certain decoration elements to the text
  - possible values are none, underline, overline, line-through and blink
- Syntax
  - p {text-decoration: underline}

Text properties

- text-indent
  - indents the first line of text inside an element
  - possible values are length (defines a fixed value) and % (defines a % of the parent element)
- Syntax
  - p {text-indent: 20px}

Font Properties

Font properties

- Define the look of the font in text areas
- One of the broader sets of properties in CSS

Font properties

- font

- font-style
- font-variant
- font-weight
- font-size/line-height
- font-family

Font properties

- font-style
  - normal
  - italic
  - oblique

Syntax: body {font-style: italic}

Font properties

- font-variant
  - normal
    - font displays as is
  - small-caps
    - font displays in all capitals, with lower case letters in smaller size

Syntax: body {font-variant: small-caps}

Font properties

- font-weight
  - normal
  - bold
  - bolder
  - lighter
  - weighted values

Syntax: body {font-weight: bold}

Weighted values

- range from 100 – 900
- 400 is the same as normal weight
- 700 is the same as bold weight

Font properties

- font-size
  - xx-small to xx-large
  - smaller
    - smaller than parent
  - larger
    - larger than parent
  - %
    - % of the parent

Syntax: body {font-size: 20px}
{font-size: x-large}
{font-size: 125%}

Font properties

- font-family
  - family-name
    - "times", "arial", "courier", "verdana"
  - generic-family
    - "serif", "sans-serif", "monospace"

Syntax: body {font-family: verdana, sans-serif}

Border Properties

Border properties

- Allows you to specify the style, color and width of an element's border
- Many different properties can be applied

Border properties

- You can specify the width, style, color, thickness and on which sides the border appears

Margin Properties

Margin properties

- Define the space around elements
- You can use negative values to overlap content
- Margins can be set independently or collectively
- Can be set to auto, a fixed length or a % of the total height of the document

Margin properties

- Properties
  - margin
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left

Margin properties

- margin-bottom
  - auto
    - set by the browser
  - *length*
    - *fixed*
  - %

Syntax: h1 {margin-bottom: 20px}

Margin properties

- Can be set in one declaration
- Think clock face
  - top, right, bottom, left

```
h1 {margin: 10px 20px 30px 40px}
```

Margin properties

- All margins can be set the same

```
h1 {margin: 40px}
```

Margin properties

- Margin settings can be paired (left and right, top and bottom)

```
h1 {margin: 40px 5%}
```

In this example, the top and bottom margins would be 40 pixels, While the left and right margins would be 5% of the total height of the document.

Margin properties

- 0 size margins do not need to be specified. 0px, 0pt and 0 are all equivalent.

```
h1 {margin: 40px 0 5% 0}
```

In this example, the top margin would be 40 pixels, the left and right margins would be 0, and the bottom margin would be 5% of the total height of the document.

FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Using the and Elements

Using the and Elements

The (division) and (span of words) elements are designed to be used with CSS.

Division Tag

A tag is a container that divides the HTML document into sections. It is used for grouping HTML elements.

DIV TAG FOR LAYOUT

The CSS float property – float property tells elements how to arrange , float values are left and right

Webpage

```
<div class="header">This is the header</div>
<div class="menu">This is your menu</div>
<div class="content">This is the body content</div>
```

OUTPUT

Float Property

Left- The element floats to the left

Right - The element floats to the right

Inherit - Inherits this property from its parent element.

Ex:

```
span {
   color: blue;
}

.extra span {
   color: inherit;
}
```

DIV TAG FOR LAYOUT

```
.header{
    background-color: orange;
    width:800px;
}
.menu{
    background-color: red;
    width:300px;
    float:left;
    }
.content{
    background-color: yellow;
    width:500px;
    float:left;

}
```

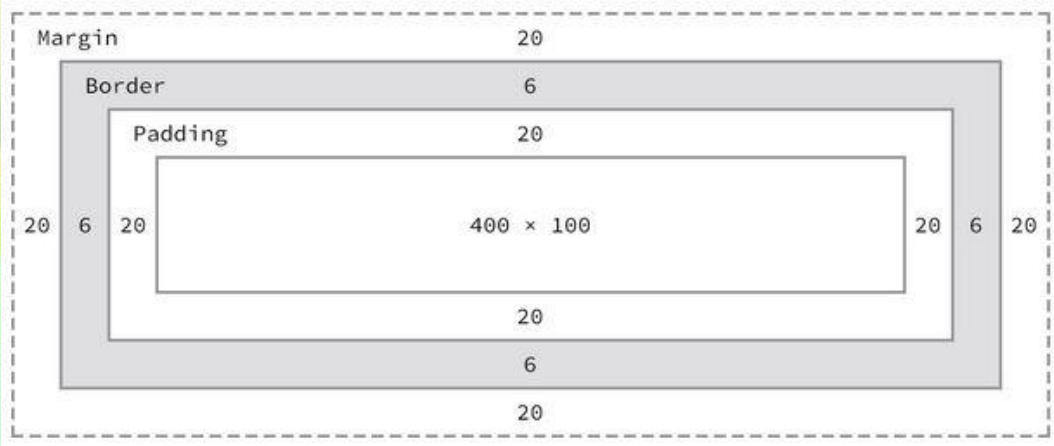
This is the header

This is your menu

This is the body content

CSS BOX MODEL using

Essentially a box that wraps around HTML elements that consists of: margin, borders, padding, and actual content



CSS syntax -

- <DIV> can be used with the CLASS attribute to create customized block-level elements
  - Declare it in the style rule:

```
DIV.introduction {font-size: 14pt; margin: 24 pt;}
```

  - Apply the style rule in the document:

```
<DIV CLASS="introduction"">This is the introduction
  to the document</DIV>
```

Sample

External css

```
#van{
   width: 200px;
   height: auto;
   padding:  15px;
   border: thin solid;
}
```

Html file

```
<html>
<head> <title> DIV </title>
<link rel="stylesheet"
     type="text/css"
     href="style.css" />
</head>

<div id=“van">
Sample text
</div>
</html>
```

Span of Words Tag

A elements lets you specify inline elements within a document that have their own name and style properties that reside within a line of text.

CSS syntax -

- <SPAN> can be used with the CLASS attribute to create customized inline elements
  - Declare it in the style rule:

    SPAN.logo {color: white; background-color: black;}
  - Apply the style rule in the document:

    <P>Welcome to the <SPAN CLASS="logo""> Wonder Software</SPAN>Web site</P>

Sample

External css

```
#logo {
    color:white;
    background-color: black;
}
```

Html file

```
<html>
<head> <title> SPAN </title>
<link rel="stylesheet"
    type="text/css"
    href="style.css" />
</head>

<h1>This is my sample text for
    <span id="logo">
    span.</span></h1>
</html>
```

Using Other CSS Selectors

Besides class and id, attribute selectors can also be used such as **PSEUDO-CLASS AND PSEUDO-ELEMENT SELECTORS, and CSS3 selectors.**

Using Attribute Selectors

Example:

 This is our sample text

ELEMENT HAS THREE ATTRIBUTES: **title**

IN CSS:

```
p[title] {background-color: blue;font-family: sans-serif;}
```

Using Pseudo-Class and Pseudo Element Selectors

Pseudo-class and pseudo-element selector let you express style declarations for characteristics of a document that are not signified with the standard HTML elements.

EXAMPLE:

Assume you want to change the color of a new or visited hypertext link. No HTML element directly lets you express these characteristic of the <a> element.

Using Pseudo-Class

The link pseudo-classes let you change the style characteristics for different hypertext link states.

Pseudo-Class	Description
:link	Selects any unvisited link that user has not clicked
:visited	Selects any link that your user has already visited
:hover	Selects any link that your user is pointing to with the mouse pointer
:active	Selects a link for the brief moment that your user is actually clicking the link

Application:

```
a:link {color:red;}
a:visited {color: green;}
```

Note: Always place your link pseudo-class in the following order Link, Visited, Hover, Active

Using the :hover Pseudo-Class

Lets you apply a style that appears when the user points to an element with a pointing device.

Example:

```
a:hover {background-color: yellow;}
```

Using the :first-letter Pseudo-Element

Apply style rules to the first letter of any element.

Example:

```
p:first-letter {
        font-weight: bold;
        font-size: 200%;
                }
```

End of Module