Study Unit 4: LAYOUT TECHNIQUES

Outline

Students will learn More CSS HTML attributes: CSS id, class multiple classes, Pseudo-classes Styling Page Sections

- div, span
- CSS Cascading & Inheritance
- Introduction to Layout Box Model, document flow
- Properties for borders, paddings, margins
- Properties for dimensions

- More CSS(id & class)
- Styling Page Sections
- Introduction to Layout

Learning Outcomes of Study Unit 4

Upon completion of this study unit, you should be able to learn more basic HTML with div and CSS with id and classes in practice.

1.1: More CSS
- HTML attributes: id, class
- Multiple classes • Pseudo-classes

1.2: Styling Page Sections
- div , span
- CSS context selector
- CSS Cascading & Inheritance

1.3: Introduction to Layout
- Box Model, document flow
- Properties for borders, paddings, margins
- Properties for dimensions

1.1: More CSS

Outline

- **More CSS(id & class)**
- Styling Page Sections
- Introduction to Layout

The HTML id Attribute

```
<p>Spatula City! Spatula City!</p> <p id="mission">Our mission is to provide the most spectacular spatulas and splurge on our specials until our customers <q>esplode</q> with splendor!</p>
```

HTML

Spatula City! Spatula City!
Our mission is to provide the most spectacular spatulas and splurge on our specials until our customers esplode with splendor!

output

- Allows you to give a unique ID to any element on a page
- Each ID must be unique; can only be used once in the page

Linking to Sections of a Web Page

```
<p>Visit <a href=
“http://www.textpad.com/download/index.html#downloads">
textpad.com</a> to get the TextPad editor.</p> <p><a
href="#mission">View our Mission Statement</a></p>
```

HTML

Visit textpad.com to get the TextPad editor.

View our Mission Statement

output

- A link target can include an ID at the end, precede by a #
- Browser will load that page and scroll to element with given ID

Box Model **CSS**

ID Selectors

```
#mission {
    font-style: italic;
    font-family: "Garamond", "Century Gothic", serif;
} HTML
```

Spatula City! Spatula City!

Our mission is to provide the most spectacular spatulas and splurge on our specials until our customers esplode with splendor! output

- Applies style only to the paragraph that has the ID of mission
- Element can be specified explicitly p#mission { … }

The HTML class Attribute

```
<p class="shout">Spatula City! Spatula City!</p>
<p class="special">See our spectacular spatula specials!</p> <p
class="special">Today only: satisfaction guaranteed.</p>
```

HTML

Spatula City! Spatula City!
See our spectacular spatula specials!
Today only: satisfaction guaranteed.

output

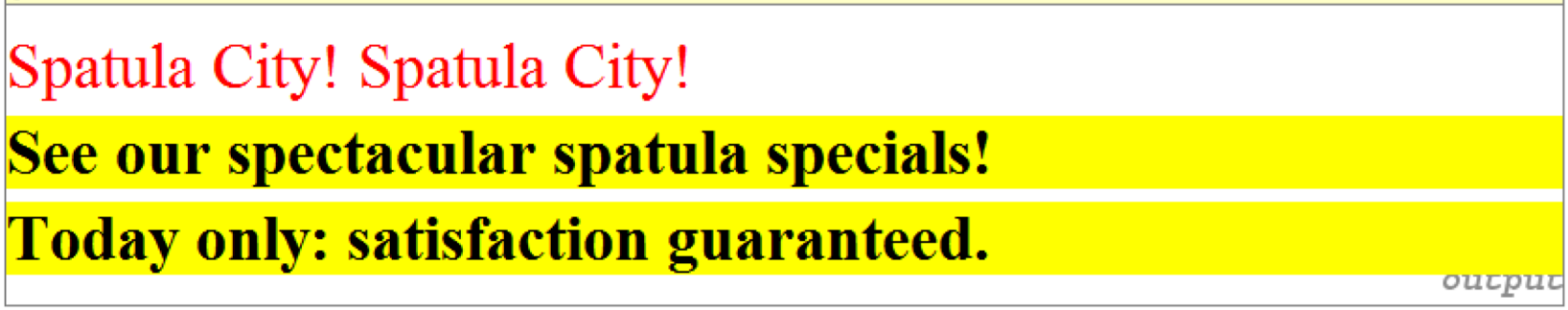
- Classes are a way to group some elements and give a style to only that group
- Unlike an id, a class can be reused as much as you like on the page

Box Model **CSS**

class Selectors

```css
.special {
  background-color: yellow;
  font-weight: bold;
}
p.shout {
  color: red;
  font-family: cursive;
}
```

CSS



output

- Applies corresponding rule to any element with class special or a p with class shout

Multiple Classes

```html
<h2 class="shout">Spatula City!  Spatula City!</h2>
<p class="special">See our spectacular spatula specials!</p>
<p class="special shout">Satisfaction guaranteed.</p>
<p class="shout">We'll beat any advertised price!</p>
```

HTML

Spatula City! Spatula City!

See our spectacular spatula specials!

Satisfaction guaranteed.

We'll beat any advertised price!

output

- An element can be a member of multiple classes (separated by spaces)

CSS pseudo -classes

```
a:link    { color: #FF0000; }    /* unvisited link */
a:visited { color: #00FF00; }    /* visited link */
a:hover   { color: #FF00FF; }    /* mouse over link */
```
css

Buy early, buy often!

output

Class	Description
:active	an activated or selected element
:focus	an element that has the keyboard focus
:hover	an element that has the mouse over it
:link	a link that has not been visited
:visited	a link that has already been visited
:first-letter	the first letter of text inside an element
:first-line	the first line of text inside an element
:first-child	an element that is the first one to appear inside another

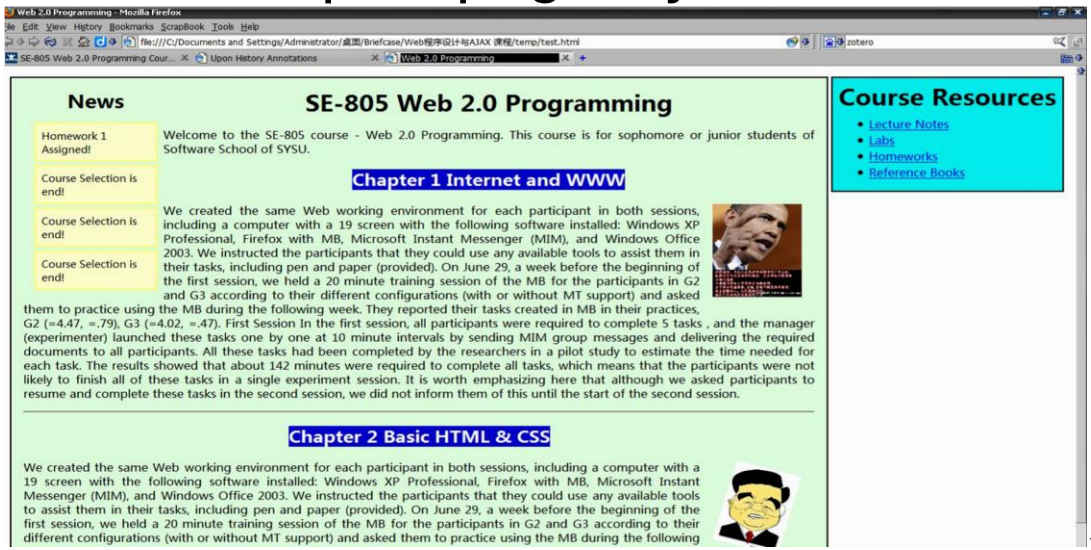
1.2: Styling Page Section

Outline

- More CSS
- **Styling Page Sections**
- Introduction to Layout

Page Segmentation

- style **individual elements, groups of elements,** and **sections of text** of the page differently
- create complex page layout



s

a section or division of your HTML page (block)

```
<div class="shout">
  <h2>Spatula City!  Spatula City!</h2>
  <p class="special">See our spectacular spatula specials!</p>
  <p>We'll beat any advertised price!</p>
</div>
```

HTML

output

Sections of a Page: < div >

- A tag used to indicate a logical section or area of a page
- Has no appearance by default, but you can apply styles to it

Inline Section: < span >

an inline element used purely as a range for applying styles

```
<h2>Spatula City!  Spatula City!</h2>
<p>See our <span class="special">spectacular</span> spatula specials!</p>
<p>We'll beat <span class="shout">any advertised price</span>!</p>
```

HTML

Spatula City! Spatula City!

See our **spectacular** spatula specials!

We'll beat any advertised price!

output

- Has no onscreen appearance, but you can apply a style or ID to it, which will be applied to the text inside the span
- So, when should we use , , and when <p>, < h1>, etc.?

CSS Context Selectors

```
selector1 selector2 {
    properties
}
```

CSS

Applies the given properties to *selector2* only if it is inside a *selector1* on the page

```
selector1 > selector2 {
    properties
}
```

CSS

Applies the given properties to *selector2* only if it is **directly** inside a *selector1* on the page (*selector2* tag is immediately inside *selector1* with no tags in between)

```html
<p>Shop at <strong>Hardwick's Hardware</strong>...</p>
<ul>
  <li>The <strong>best</strong> prices in town!</li>
  <li>Act while supplies last!</li>
</ul>
```

HTML

```css
li strong { text-decoration: underline; }
```

CSS

Shop at **Hardwick's Hardware**...

- The **best** prices in town!
- Act while supplies last!

output

More Complex Example

```html
<div id="ad">
  <p>Shop at <strong>Hardwick's Hardware</strong>...</p>
  <ul>
    <li class="important">The <strong>best</strong>
    prices in town!</li>
    <li>Act <strong>while supplies last!</strong></li>
  </ul>
</div>
```

HTML

```css
#ad li.important strong { text-decoration: underline; }
```

CSS

Shop at **Hardwick's Hardware**...

- The **best** prices in town!
- Act **while supplies last!**

output

CSS Cascade

- The browser's style sheet is the weakest.
- The user's style sheet takes precedence over the browser's style sheet.
- The author's style sheet is the strongest and takes precedence over the user's and the browser's style sheets.
- The (X)HTML style attribute is more important than styles defined in any style sheet.
- Within a style sheet, when conflict occurs, the most specific rule wins.

Specificity of Selector

- When two conflict rules has same specificity, the one occurs later is style sheet file wins

  At last, a rule with **!important** overrides precedence!

Specificity of Selector

Selector	Selector Type
Li	Element Name
ul li	Element Name
div h1 + p	Element Name
input[type='text']	Element Name + Attribute
.someclass	Class Name
div.someclass	Element Name + Class Name
div.someclass.someother	Element Name + Class Name + Class Name
#someid	ID Name
div#someid	Element Name + ID Name

```
body {
    font-size: 24px;
}
p {
    background: lightblue !important;
}
p {
    background: none;
}
```

The !important rule takes precedence.

CSS Inheritance

- Many properties in a CSS rule are inheritable by children elements of the rule specified, but some are not.
- Types of properties are inheritable: text, color, and font
- Types of properties are not: border, margin, padding
- If you can't remember whether a property is inheritable, better to figure it out by examining, other than via Googling or W3-Schooling

1.3: Introduction to Layout

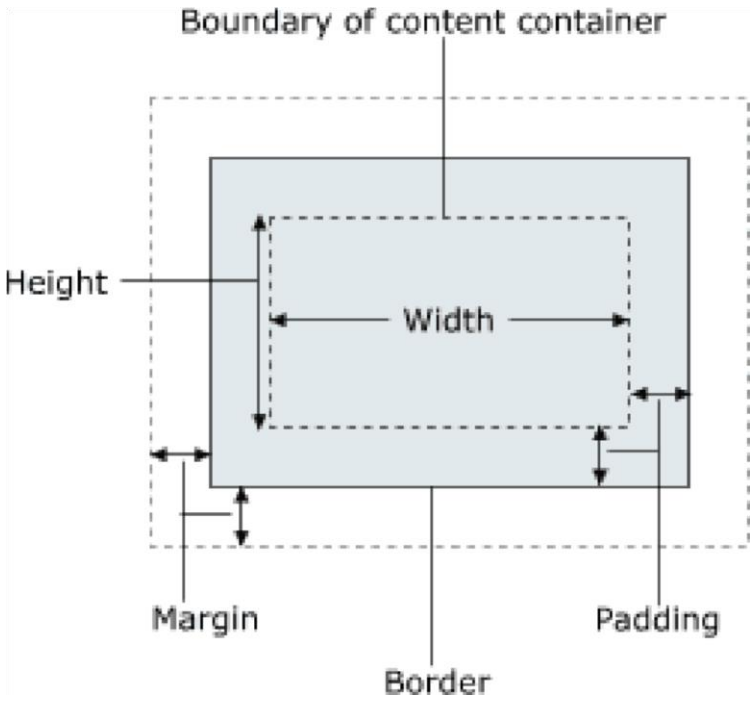
Outline

- More CSS
- Styling Page Sections
- **Introduction to Layout**

The CSS Box Model

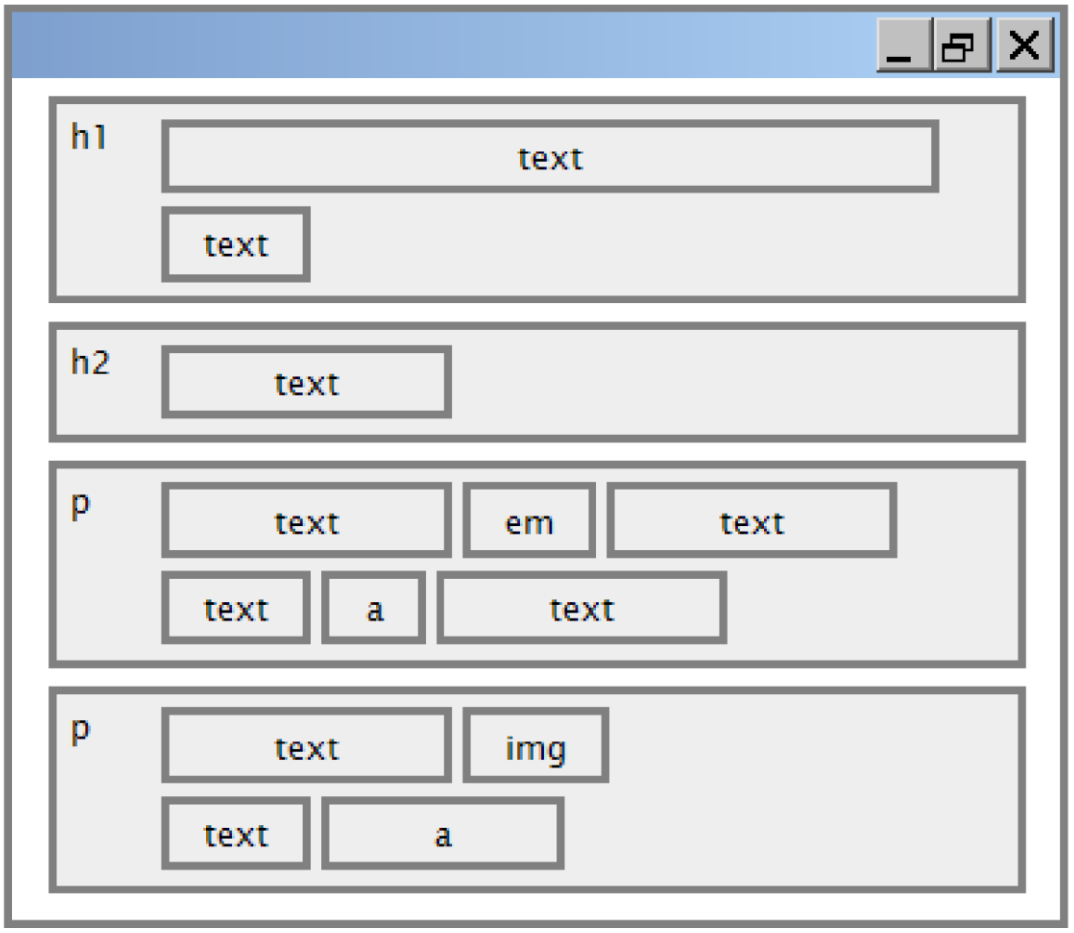
- For layout purpose, every element is composed of:
  - The actual element's **content**
  - A **border** around the element
  - A **padding** between the content and the border (*inside*)
  - A **margin** between the border and other content (outside)
- width = content width + L/R padding + L/R margin

  height = content height + T/B padding + T/B margin



Document Flow –

Example



CSS Properties for BordersMore Border Properties

Property	Description
border-color, border-width, border-style	specific properties of border on all 4 sides
border-bottom, border-left, border-right, border-top	all properties of border on a particular side
border-bottom-color, border-bottomstyle, border-bottom-width, border-left-color, border-left-style, border-left-width, border-right-color, border-right-style, border-right-width, border-top-color, border-top-style, border-top-width	properties of border on a particular side

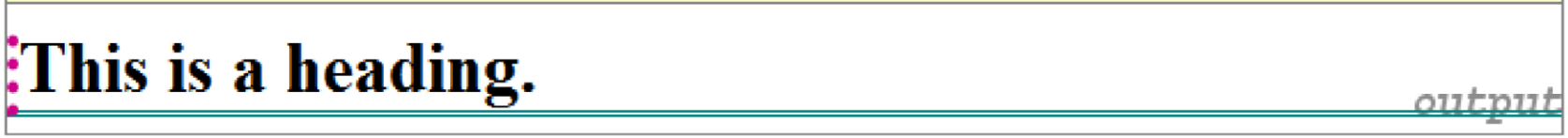
Complete list of border properties

Border Example 2

- Each side's border properties can be set individually
- If you omit some properties, they receive default values (e.g. border-bottom-width above)

```css
h2 {
  border-left: thick dotted #CC0088;
  border-bottom-color: rgb(0, 128, 128);
  border-bottom-style: double;
}
```

CSS



output

CSS Properties for Padding *Web 2.0 Programming –*

Property Description padding padding
on all 4 sides
padding-bottom padding on bottom side only
padding-left padding on left side only padding-right padding on right side only padding-top padding on top side only

Complete list of padding properties

Padding Example

Padding Example 2

```css
p { padding: 20px; border: 3px solid black; }
h2 { padding: 0px; background-color: yellow; }
```

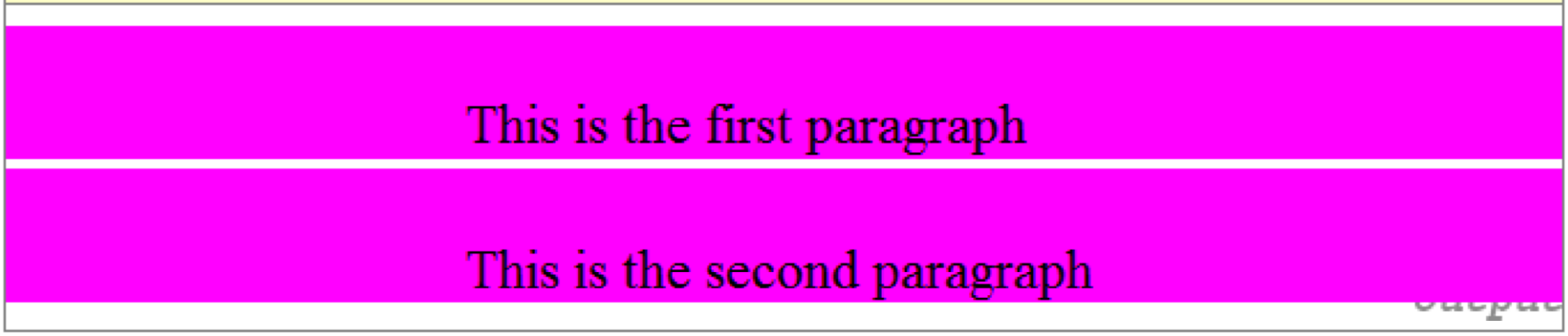
This is the first paragraph

This is the second paragraph

This is a heading

```css
p {
  padding-left: 200px; padding-top: 30px;
  background-color: fuchsia;
}
```

CSS



- Each side's padding can be set individually
- Notice that padding shares the background color of the element

CSS Properties for Margins

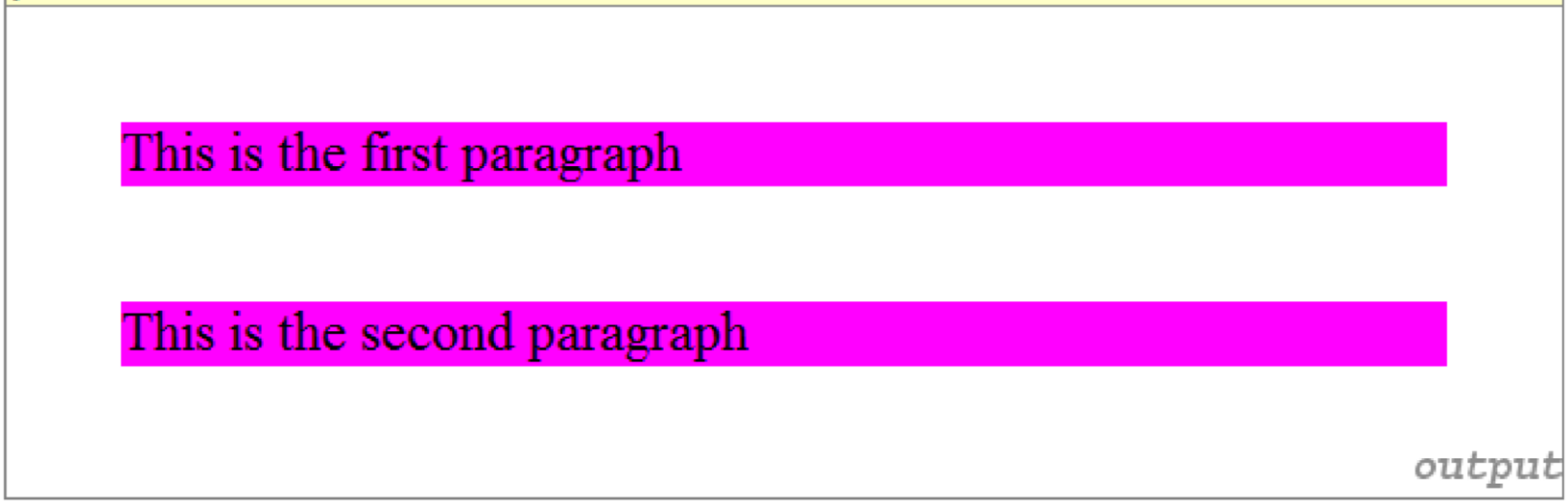
property	description
margin	margin on all 4 sides
margin-bottom	margin on bottom side only
margin-left	margin on left side only
margin-right	margin on right side only
margin-top	margin on top side only

Complete list of margin properties

Margin Example 1

```css
p {
  margin: 50px;
  background-color: fuchsia;
}
```

CSS



output

- Notice that margins are always transparent
  (they don't contain the element's background color, etc.)

Margin Example 2

```css
p {
  margin-left: 8em;
  background-color: fuchsia;
}
```

CSS

This is the first paragraph

This is the second paragraph

- Each side's margin can be set individually

CSS Properties for Dimensions

```css
p { width: 350px; background-color: yellow; }
h2 { width: 50%; background-color: aqua; }
```

Property	Description
width, height	how wide or tall to make this element (block elements only)

max-width, max-height, min-width, min-height	max/min size of this element in given dimension

Centering a Block Element: auto Margins

```css
p {
  margin-left: auto;
  margin-right: auto;
  width: 750px;
}
```

CSS

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

output

- Works best if width is set (otherwise, may occupy entire width of page)
- To center inline elements within a block element, use text-align: center

Self-Review Questions (SRQ) For Study Session 4

Now that you have completed this study unit, you can assess how well you have achieved its Learning Outcomes by answering these questions. Write your answers in your Study Diary and discuss them with your Tutor at the next Study Support Meeting or Online interactive sessions. You can also check your answers at the Self-Review Answers section which is located at the end of this Module.

1: Difference between id and class in HTML [ID vs CLASS]
2: what's the difference between table and div layout in html.
3: Create a web page looks like this:

Self-Review Answers (SRA) for Study Unit 4

1: Difference between id and class in HTML [ID vs CLASS].

HTML element (like div, input, nav, body, etc) can have both “id” and “class” together and at the same time. The only difference here is that “id” can have only one unique value and “class” can have more than one. For instance see the example below.

2: what's the difference between table and div layout in html?

Very simply, table based layouts were the old way of organizing the position of content and divs are the newer way. Think of them as being invisible rectangles (unless you choose to make them visible) that hold content and both have their pros and cons in how you are able to position them.

You can position divs into nearly every situation a table can be positioned into, whereas there's a lot of positioning you can do with divs that you can't do with tables. This is one reasons why using tables to layout your page is discouraged.

Tables haven't completely disappeared. They're still very good at (funnily enough) table-based display of information or statistics. Think excel-type layouts.

3: Create a web page looks like this:

For practice purpose use following link

https://www.w3schools.com/html/html_layout.asp

Exercises

- What are the most popular Web page fonts, and why?
- What are common layout elements of a contemporary Web page?
- Why “css + div” style layout is better than “table” style?

References and Additional Reading Materials

- W3C CSS2 Specification: http://www.w3.org/TR/REC-CSS2/
- W3Schools CSS Tutorial: http://www.w3schools.com/css/default.asp
- http://www.barelyfitz.com/screencast/html-training/css/positioning/
- http://www.quirksmode.org/css/display.html