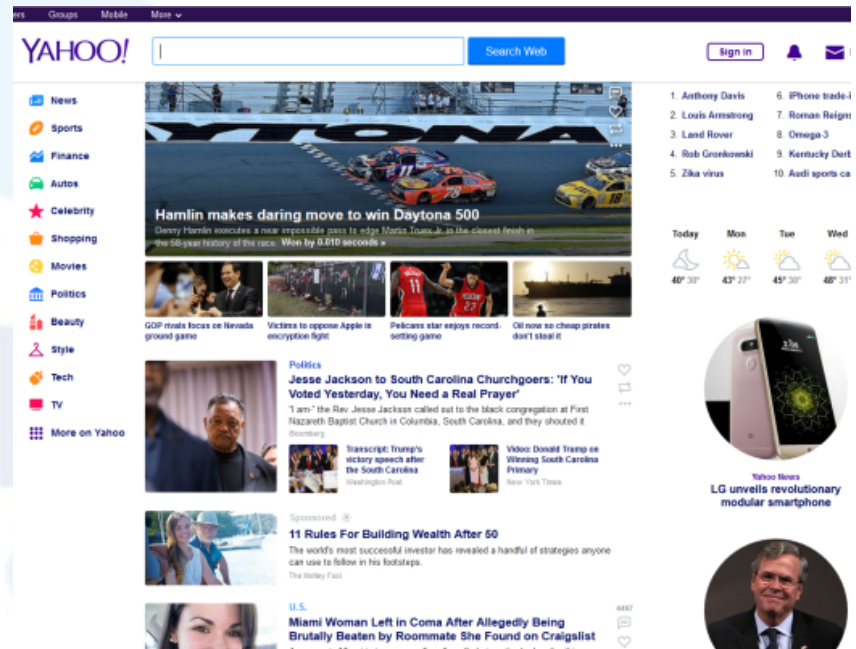# Chapter 2
CSS Cascading Style Sheets

Lectured by:

Nguyễn Hữu Hiếu

Introduction

- HTML has evolved a lot over the years - as computers and networks have gotten faster.

1995

2016

Introduction

<u>Sample text</u>

```
<strong>
    <i>
        <u>
            Sample text
        </u>
    </i>
</strong>
```
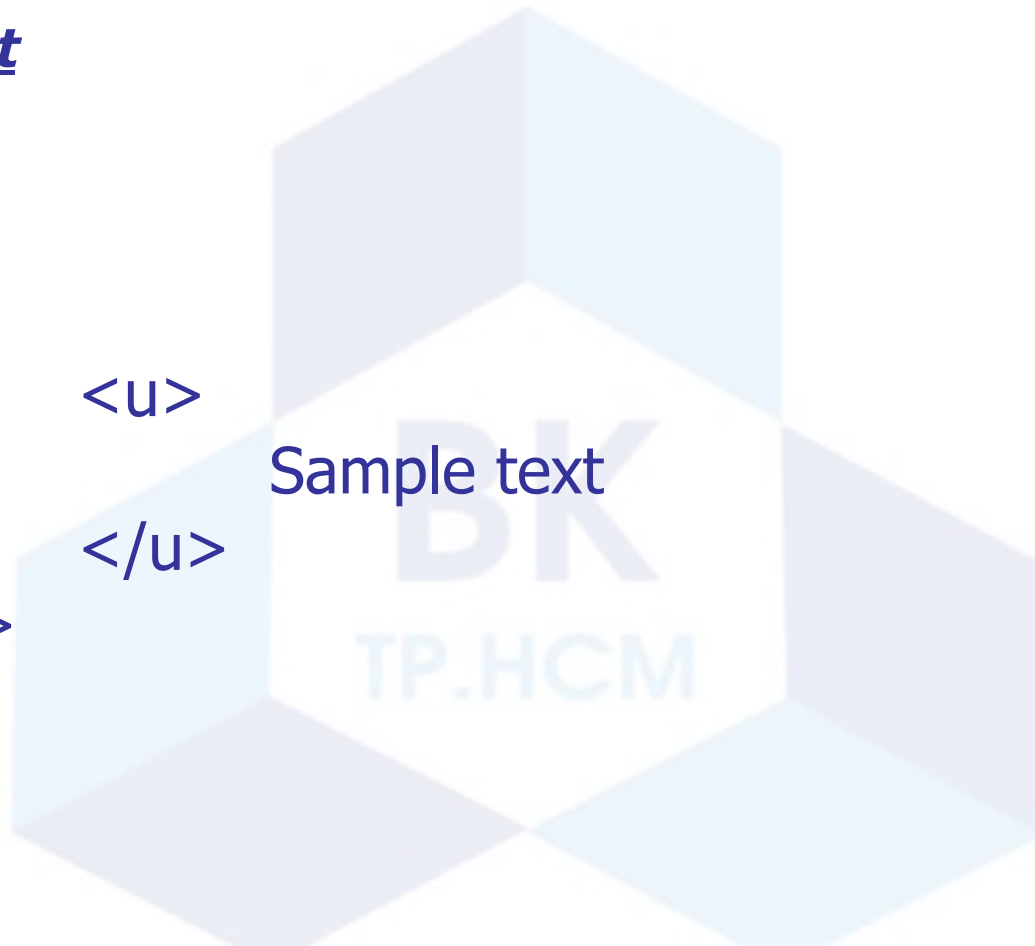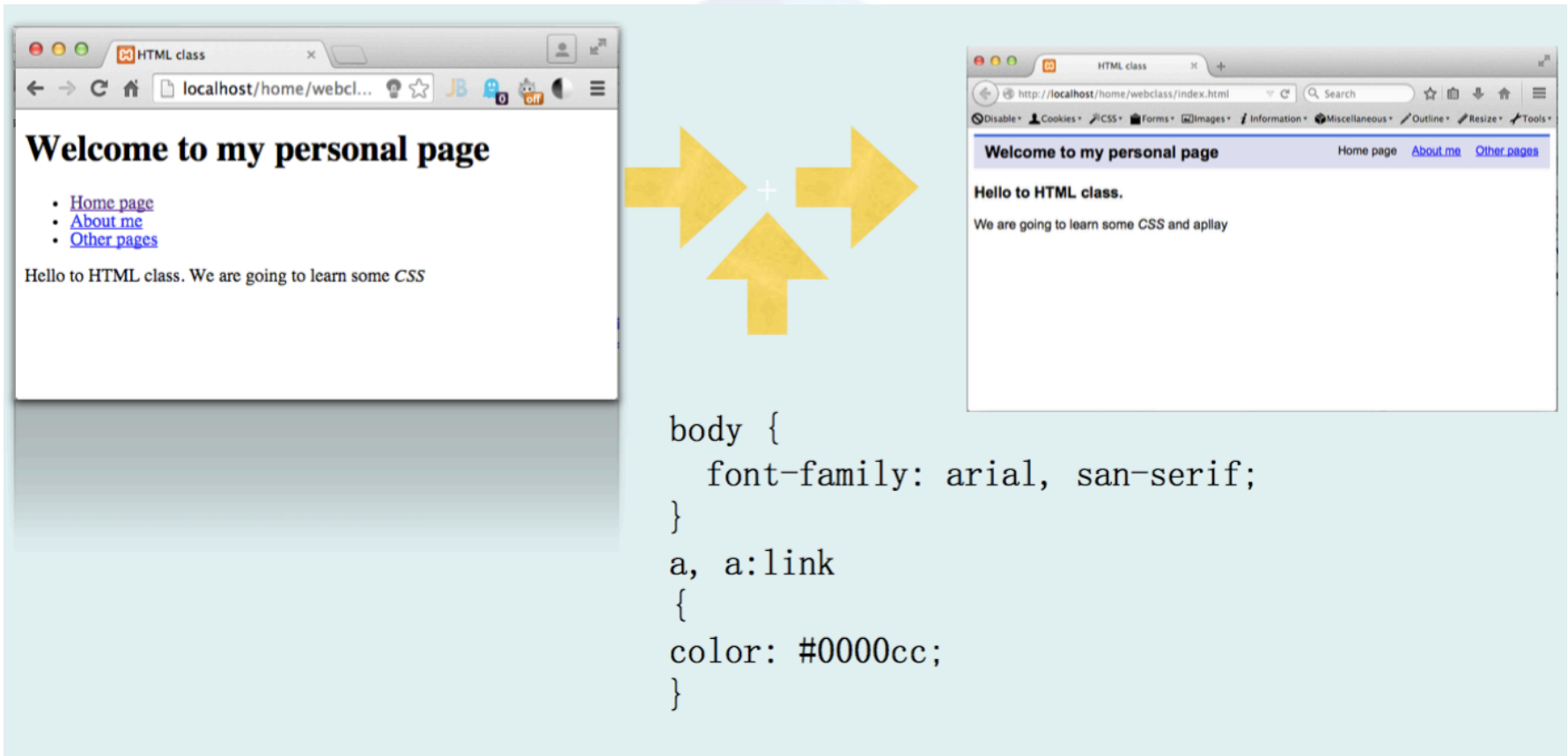
Introduction

- Tranforming the look and feel of a page using a CSS (Cascading Style Sheets) style sheet.


```
body {
    font-family: arial, san-serif;
}
a, a:link
{
color: #0000cc;
}
```

Introduction

- The Browser has "default styling" for all tags

```
<!DOCTYPE html>
 <html lang="en">
  <head>
   <meta charset="utf-8" />
   <title>HTML class</title>
  </head>
  <body>
   <h1>Welcome to my personal page </h1>
   <ul>
    <li><a href="index.html">Home page</a></li>
    <li><a href="about.html">About me</a>
    </li> <li><a href="#">Other pages</a></li>
   </ul>
   <p>Hello to HTML class. We are going to learn
some <em>CSS</em></p>
  </body>
 </html>
```
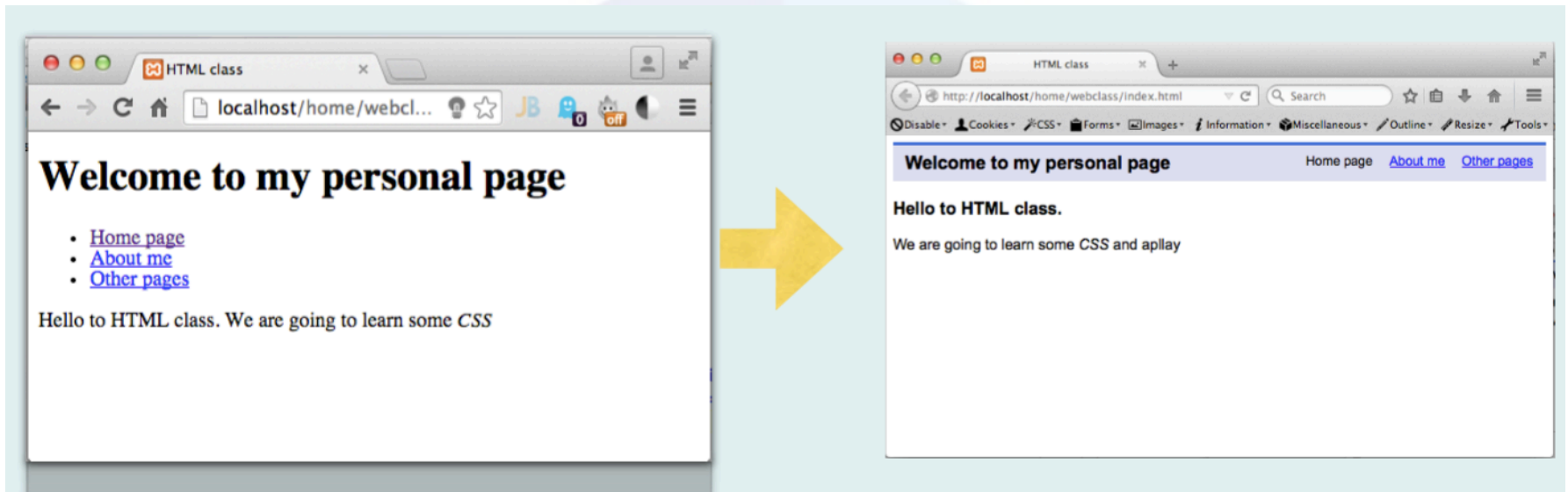
Welcome to my personal page

- Home page
- About me
- Other pages

Hello to HTML class. We are going to learn some *CSS*

Introduction

- We will apply CSS to the tags in the document. With no changes to the HTML.

Introduction

- Lots of CSS properties to play with

- background-color, border-width, border-color, margin-top, padding, font-family, top, left, right, float, font-size, background-image, text-align, text-decoration, font-style, font-weight, vertical-align, visibility, overflow,…

- We can set these properties on any HTML tag in a document.
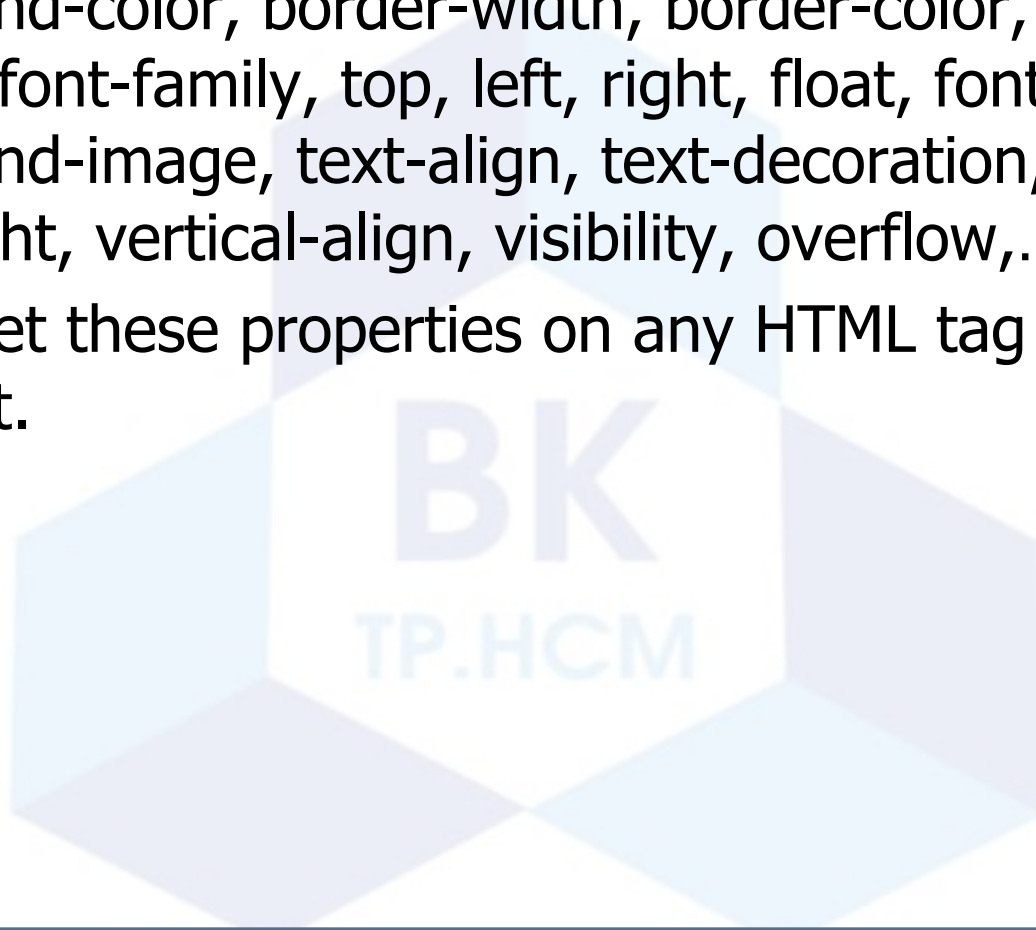
CSS Rules

- Anatomy of a CSS Rule

- selector - which part of the document does this rule apply

- property - which aspect of CSS are we changing

- value - What are we setting the property to.

```
selector {
  property: value;
  ...
}


Example:
p {
  font-family: times;
}
```

CSS Rules

- ## CSS Selector
  - https://www.w3schools.com/cssref/css_selectors.asp

| Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all \<p\> elements	1
element,element	div, p	Selects all \<div\> elements and all \<p\> elements	1
element element	div p	Selects all \<p\> elements inside \<div\> elements	1
element>element	div > p	Selects all \<p\> elements where the parent is a \<div\> element	2
element+element	div + p	Selects all \<p\> elements that are placed immediately after \<div\> elements	2
element1~element2	p ~ ul	Selects every \<ul\> element that are preceded by a \<p\> element	3

CSS Rules

- Multiple tags with same styling

h1, h2, h3 {
 **color: yellow;**
  background-color: black;
}

- Making a noticeable background color is a fun way to debug / identify blocks.
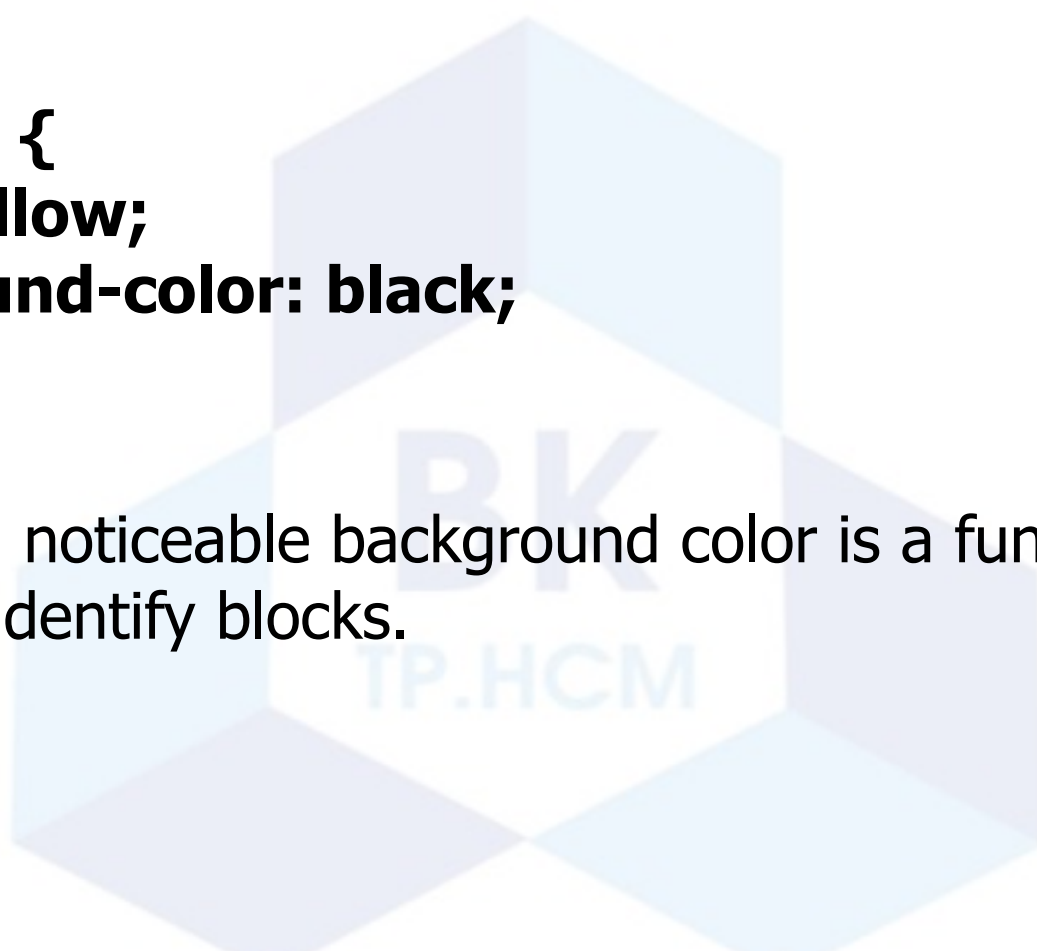
CSS Rules

- Three ways to add style rules
  - Inline Style - Add style information to a tag
  - Embedded Style - Add style information to the document at the beginning
  - External Style Sheet - Put all of your style in an external file
- Preferred - because two people can work independently
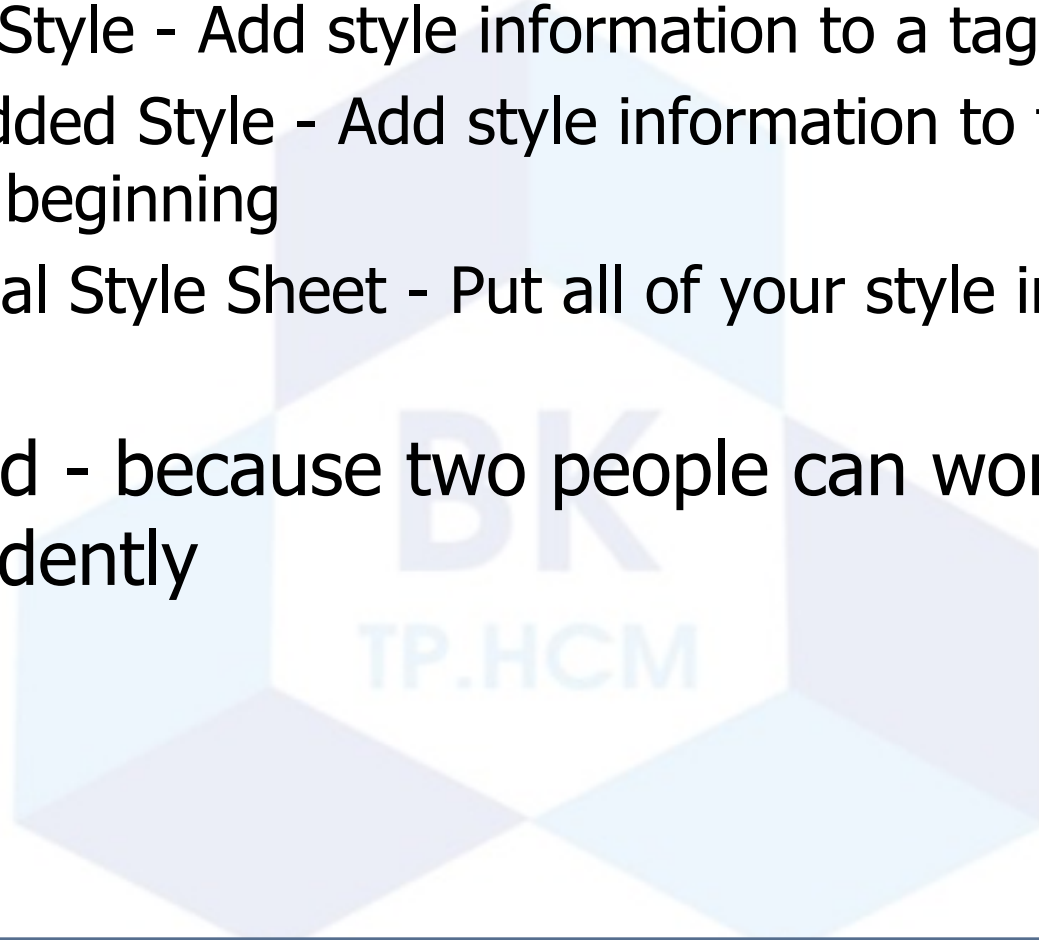
Inline Styles

```html
<!DOCTYPE html>
<html lang="en">
  <head>
   Inline Styles
   <meta charset="utf-8" />
   <title>HTML class</title>
  </head>
  <body style="font-family: arial, sans-serif;">
   <h1>Welcome to my personal page </h1>
   <ul>
     <li><a href="index.html">Home page</a></li>
     <li><a href="about.html">About me</a></li>
     <li><a href="#">Other pages</a></li>
   </ul>
   <p>Hello to HTML class. We are going to learn some <em>CSS</em></p>
  </body>
</html>
```

Welcome to my personal page

- Home page
- About me
- Other pages

Hello to HTML class. We are going to learn some *CSS*

Embedded Style (Internal CSS)

```
<!DOCTYPE html>
<html lang="en">
  <head>
   Embedded Style
   <meta charset="utf-8" />
   <title>HTML class</title>
   <style type="text/css">
     body {
       font-family: arial, sans-serif;
     }
   </style>
  </head>
  <body>
   <h1>Welcome to my personal page </h1>
   <ul>
    <li><a href="index.html">Home page</a></li>
    <li><a href="about.html">About me</a></li>
    <li><a href="#">Other pages</a></li>
   </ul>
   <p>Hello to HTML class. We are going to learn some <em>CSS</em> </p>
  </body>
</html>
```

Welcome to my personal page

- Home page
- About me
- Other pages

Hello to HTML class. We are going to learn some *CSS*

External Style Sheets (External CSS)

index.html
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>HTML class</title>
    <link type="text/css" rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Welcome to my personal page </h1>
    <ul>
      <li><a href="index.html">Home page</a></li>
      <li><a href="about.html">About me</a></li>
      <li><a href="#">Other pages</a></li>
    </ul>
    <p>Hello to HTML class. We are going to learn some <em>CSS</em></p>
  </body>
</html>
```

style.css
```
body {
  font-family: arial, sans-serif;
}
```

CSS file

- We put the CSS file in the same directory so the link works.

```
<head>
  <meta charset="utf-8" />
  <title>HTML class</title>
  <link type="text/css" rel="stylesheet" href="style.css">
</head>
```

Fonts

- Default fonts are ugly and they have Serifs - which make them harder to read on a screen

- So the first thing I usually want to do is override the font in my document

- And I want to do this everywhere.

Header One

Paragraph about ugly default fonts.

Figure 3.5. Highlighting the serifs of a serif font (Georgia)

serif

Fonts

```
body {
  font-family: "Trebuchet MS, Helvetica, Arial, sans-serif;
  font-size: x-large;
}
```

https://www.w3schools.com/cssref/css_websafe_fonts.asp

Font Factors

- **font-size**: xx-small x-small small medium large x-large xx-large

- **font-weight**: bold or normal

- **font-style**: normal or italic

- **text-decoration**: none, underline, overline, or line-through

Color Names

- W3C has listed 16 color names that will validate with an HTML validator.

- The color names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.
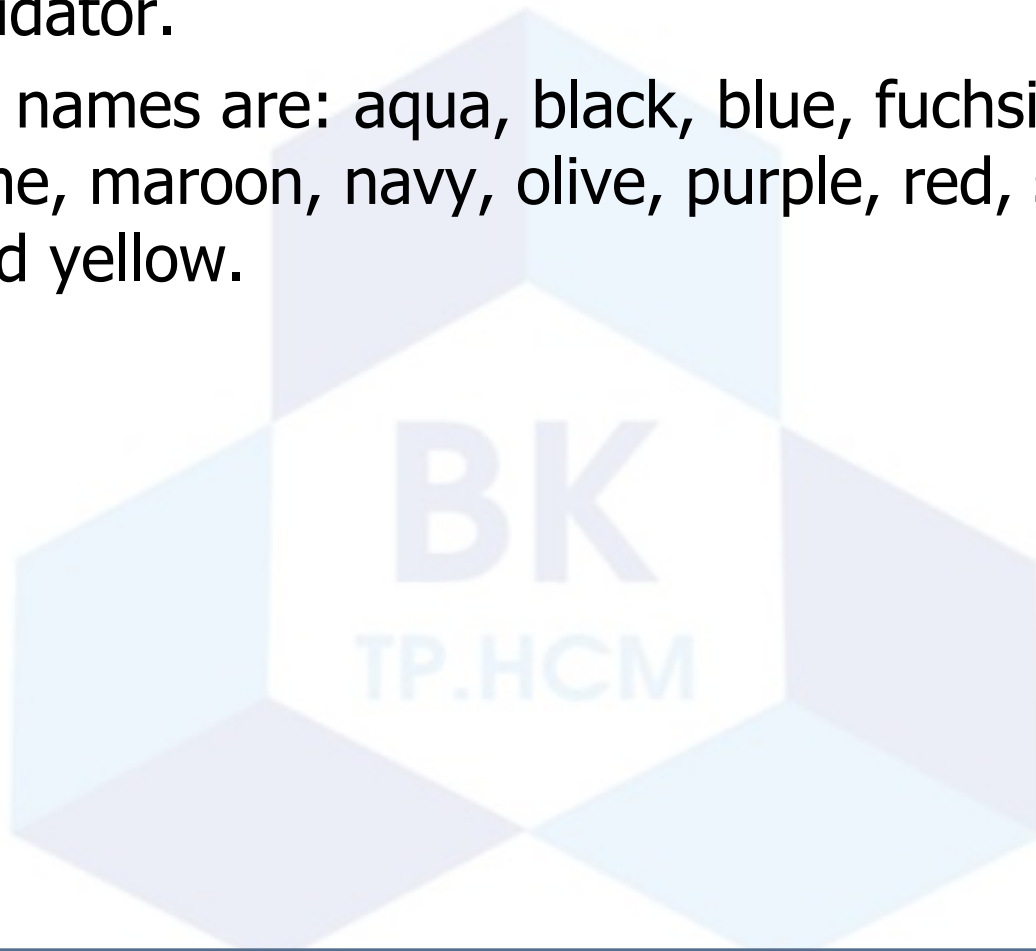
Color Names

Color Values

HTML colors can be defined as a hexadecimal notation for the combination of Red, Green, and Blue color values (RGB).

The lowest value that can be given to one light source is 0 (hex #00) and the highest value is 255 (hex #FF).

The table below shows the result of combining Red, Green, and Blue light sources:.

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

W3C Standard Color Names

W3C has listed 16 color names that will validate with an HTML validator.

The color names are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

Colors by the number...

- #e2edff
- #edf = #eeddff
- #ffffff = white
- #000000 = black
- #ff0000 = red
- #00ff00 = green
- #0000ff = blue

CSS properties:
color
background-color
border-color

Color Name	Color HEX	Color
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	

Styling Links

```
a {
  font-weight: bold;
}
a:link {
  color: black;
}
a:visited {
  color: gray;
}
a:hover {
  text-decoration: none;
  color: white;
  background-color: navy;
}
a:active {
  color: aqua;
  background-color: navy;
}
```

link - before a visit
visited - after it has been visited
hover - when your mouse is over it but you have not clicked
active - you have clicked it and you have not yet seen the new page

Tags and Attributes

- As CSS was introduced, they introduced two new tags that are pretty much there to serve as handles for styling
  - `<div>` - A block tag (breaks justification)
  - `<span>` - An inline tag that does not break justification
- There are two attributes with special meaning to CSS
  - id= - Marks a unique block within the document for styling (use only once)
  - class= - Marks a non-unique tag within the document for styling (multi-use)

div as Container

- The id attribute on the tag allows us to uniquely mark a div in a document. The id tag is also useful for screen readers.
  - "div" stands for "division" as it allows us to divide our page into parts or sections and then do something different with each "section".

```
<div id="header">
  <h1>Welcome to my personal page </h1>
  <ul>
   <li><a href="index.html">Home
    page</a>
   </li>
   <li><a href="about.html">About me</a></li>
   <li><a href="#">Other pages</a></li>
  </ul>
</div>
```

Styling a block with "id"

Everything within block
```css
#footer {
  font-style: italic;
  font-family: Times, serif;
}
```

Paragraphs within block
```css
#footer p {
  font-style: italic;
  font-family: Times, serif;
}
```

```html
<div id="footer">
  <p>Please send any comments to asomari@uqu.edu.sa</p>
</div>
```

id= identifies a particular block - only one in a document

Nested divs

- Adding divs give us a "handle" to apply styling (CSS) to a block of text.

```
<div id="outer">
  <div id="nested1">
    <p>A paragraph inside the first nested div.</p>
  </div>
  <div id="nested2">
    <p>A paragraph inside the second nested div.</p>
  </div>
</div>
```

Paragraphs and Divs

```
<p>This is a paragraph.</p>
<div>This looks like a paragraph, but it's actually a div.</div>
<p>This is another paragraph.</p>
<div>This is another div.</div>
```

Styling with class

```
.fun {
  color: #339999;
  font-family: Georgia, Times, serif;
  letter-spacing: 0.05em;
}
```

class can be used many times in a document.

```
<p class="fun">A man walks into a bar; you would've thought he'd see it coming!
</p>
<p>Have a nice day.</p>
<p class="fun">More fun stuff</p>
```

Span

```
<p>
 <span class="fun">Bubble Under</span>
 is a group of diving enthusiasts based in the south-west UK who meet up for
diving trips in the summer months when the weather is good and the bacon rolls
are flowing. We arrange weekends away as small groups to cut the costs of
accommodation and travel and to ensure that everyone gets a trustworthy dive
buddy.
</p>
```

- Sometimes you want to style something smaller than a whole block - then use span. Do not use span if you are applying something to a whole block - just put your styling on the enclosing block tag.

CSS Box Model

- **height** and **width** properties size the block element

- **margin** properties define the space around the block element

- **border** properties define the borders around a a block element

- **padding** properties define the space between the element border and the element content

- **background** properties allow you to control the background color of an element, set an image as the background, repeat a background image vertically or horizontally, and position an image on a page

CSS Box Model

```
<p class="trapped">
  I am trapped in a glass case of emotion which is 100px high and 200px wide.
</p>
```

```
.trapped {
  height: 100px;
  width: 200px;
  margin: 20px;
  border: 5px solid yellow;
  background:red;
  padding: 20px;
  font-family:Arial;
  color:orange;
  font-size:20px;
}
```


I am trapped in a glass case of emotion which is 100px high and 200px wide.
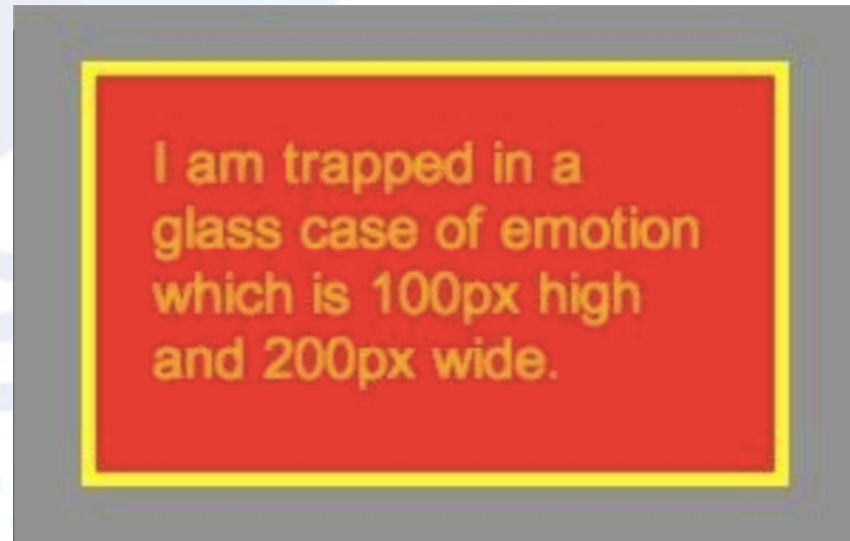
CSS Box Model

```
<p class="trapped">One</p>
<p class="trapped2">Two</p>
```

```
.trapped {
  height: 50px;
  width: 50px;
}
.trapped2 {
  height: 50px;
  width: 50px;
  border: 5px solid yellow;
  padding: 10px;
}
```
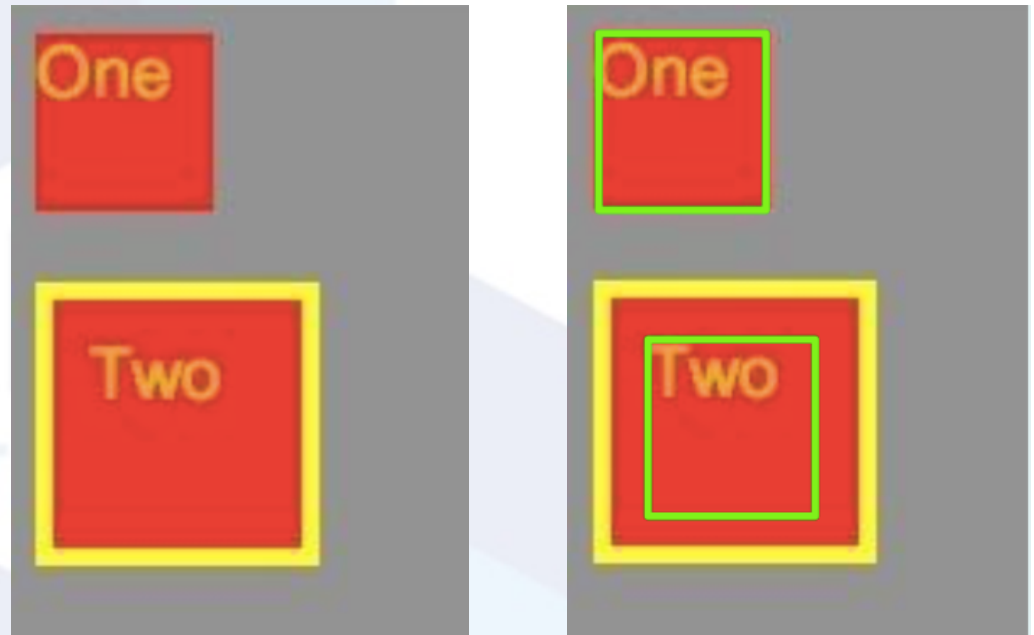
CSS Validation

- You can validate your CSS to make sure it has no syntax errors

- Browsers will generally quietly ignore bad CSS syntax

- http://jigsaw.w3.org/css-validator

- The validator can save you time and sanity

CSS Cheatsheet

- http://www.lesliefranke.com/files/reference/csscheatsheet.html

Layout

Side	Main Content
content...	content

1. Float and Clear

1. Float and Clear

- In addition to managing boxes, another powerful technique you can use to organize your layout on the page involves combining floating and clearing using the `float` and `clear` properties.

- Floating an element is a way of moving it out of the normal flow of the document. Elements that follow a floated element will move up and set next to the floated element if there is room. Value: left, right

- The `clear` property enables you to stop elements moving up next to a floated element. Value: left, right, both

1. Float and Clear

- Suppose that you have two paragraphs and only want the first one to set next to a floated element, even though both paragraphs would fit next to the floated element, you can "clear" the second one so it will be positioned under the floated element rather than along side of it.

- Over the next few pages we'll look at these two properties more closely.

1. Float and Clear

Side	Main Content
content...	content

```
<div class="row">
  <div class="column side">
    <h2>Side</h2>
    <p>content...</p>
  </div>

  <div class="column middle">
    <h2>Main Content</h2>
    <p>content</p>
  </div>

  <div class="clear"></div>
</div>
```
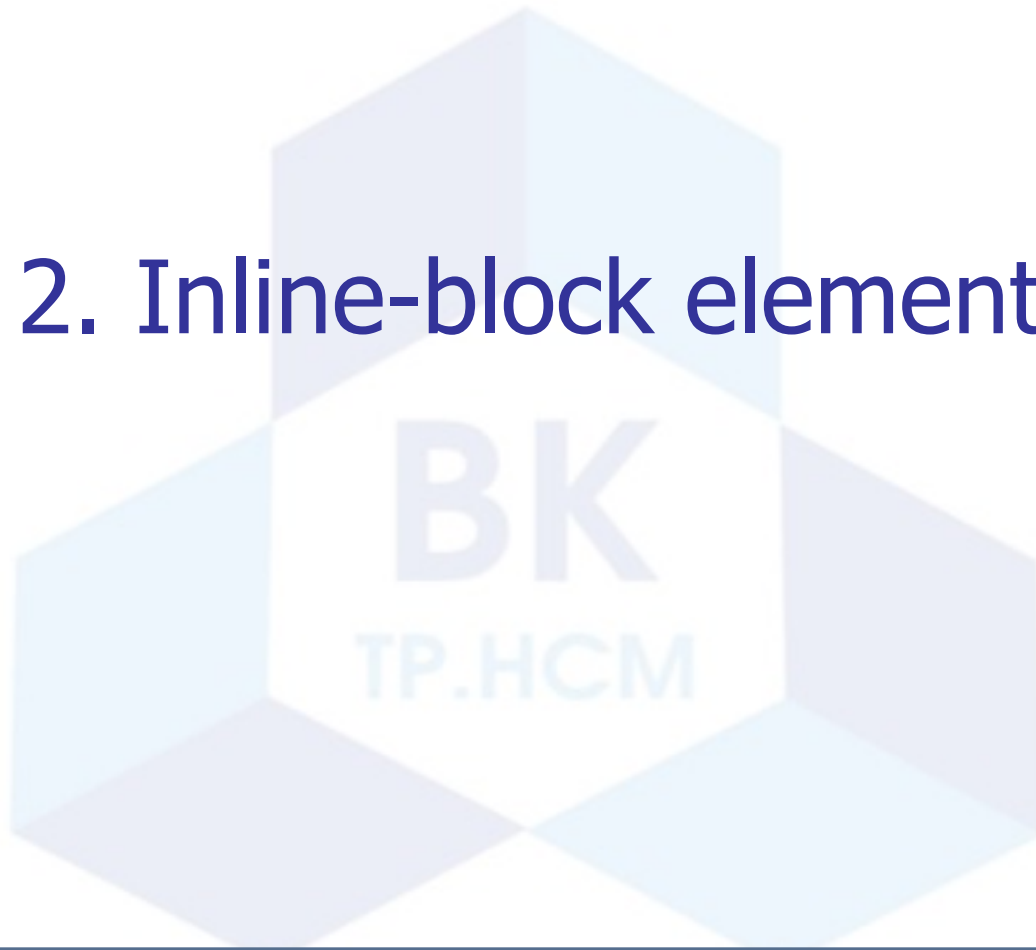
```
.column {
  float: left;
  padding: 10px;
  border: 1px black solid;
}
.column.side {
  width: 200px;
}
.column.middle {
  width: 400px;
}
.clear {
  clear: both;
}
```

2. Inline-block elements

2. Inline-block elements

- inline-block elements are like inline elements but they can have a width and height

- Using display property

- Example - display: inline-block

2. Inline-block elements

Side	Main Content
content...	content

```html
<div class="row">
  <div class="column side">
    <h2>Side</h2>
    <p>content...</p>
  </div><div class="column middle">
    <h2>Main Content</h2>
    <p>content</p>
  </div>
</div>
```

```css
.column {
  padding: 10px;
  border: 1px black solid;
  display: inline-block;
}
.column.side {
  width: 200px;
}
.column.middle {
  width: 400px;
}
.clear {
  clear: both;
}
```

3. flexbox

3. flexbox

- The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

🌐	e	🦊	🧭	O
29.0	11.0	22.0	10	48

- More about flexbox:

  - https://www.w3schools.com/css/css3_flexbox.asp

  - https://css-tricks.com/snippets/css/a-guide-to-flexbox/

3. flexbox

Side	Main Content
content...	content

```html
<div class="flex-container">
  <div class="column side">
    <h2>Side</h2>
    <p>content...</p>
  </div>
  <div class="column middle">
    <h2>Main Content</h2>
    <p>content</p>
  </div>
</div>
```

```css
.flex-container {
  display: flex;
}
.column {
  padding: 10px;
  border: 1px black solid;
}
.column.side {
  width: 200px;
}
.column.middle {
  width: 400px;
}
```

Summary

- CSS Layout is its own art and science
- CSS Basics are well established and well supported in all modern browsers
- The box model is pretty straightforward - and allows nice design within the standards with reasonable effort levels.
- Site layout and markup is further evolving - mostly to make it increasingly possible to support desktop like experiences on the web.
- These innovations will naturally cause incompatibilities - which make things interesting and frustrating at times.

Tài Liệu Tham Khảo

- [1] Stepp,Miller,Kirst. Web Programming Step by Step.(1st Edition, 2009) Companion Website: http://www.webstepbook.com/

- [2] W3Schools, http://www.w3schools.com/html/default.asp