

Chapter 2

CSS Cascading Style Sheets



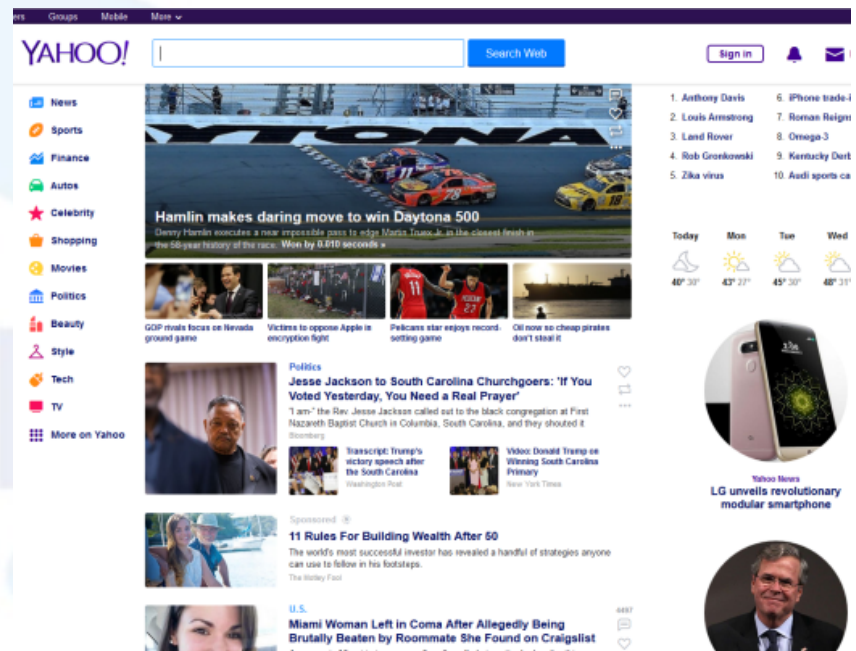
Lectured by:
Nguyễn Hữu Hiếu

Introduction

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



1995



2016

Introduction

Sample text

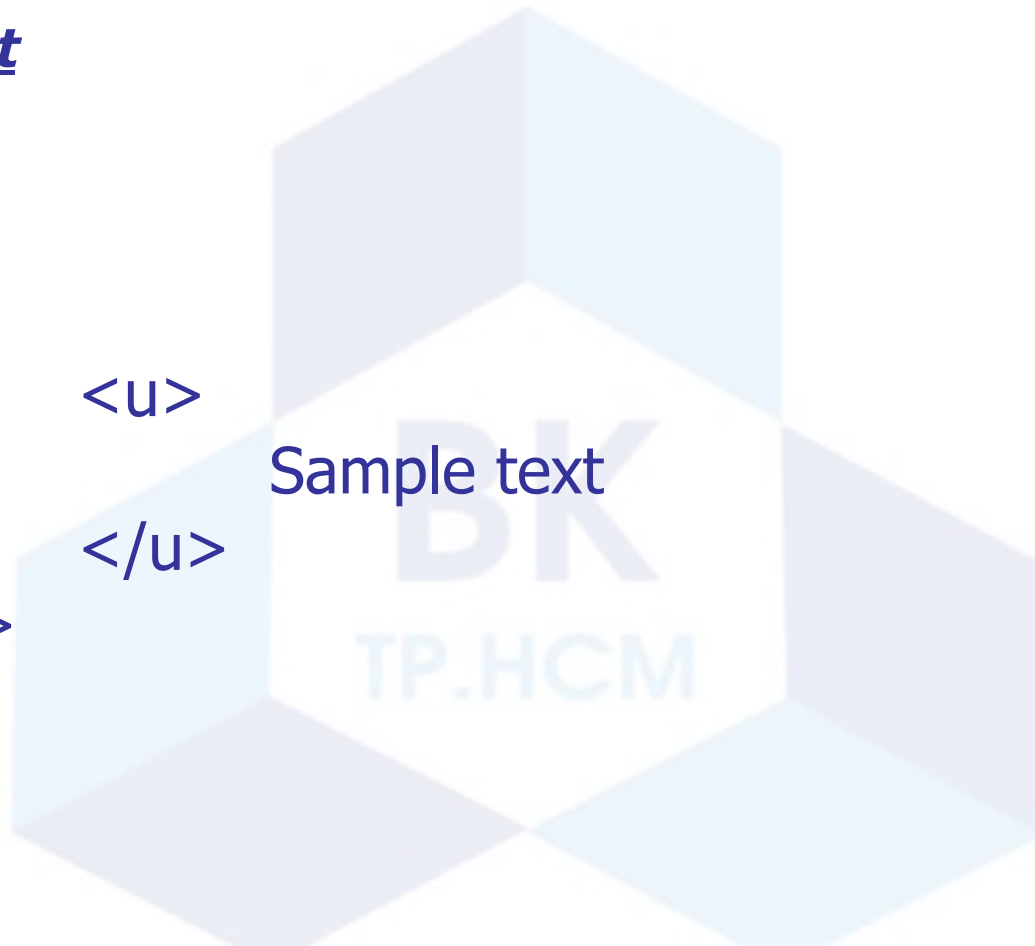
 <i>

<u>

Sample text

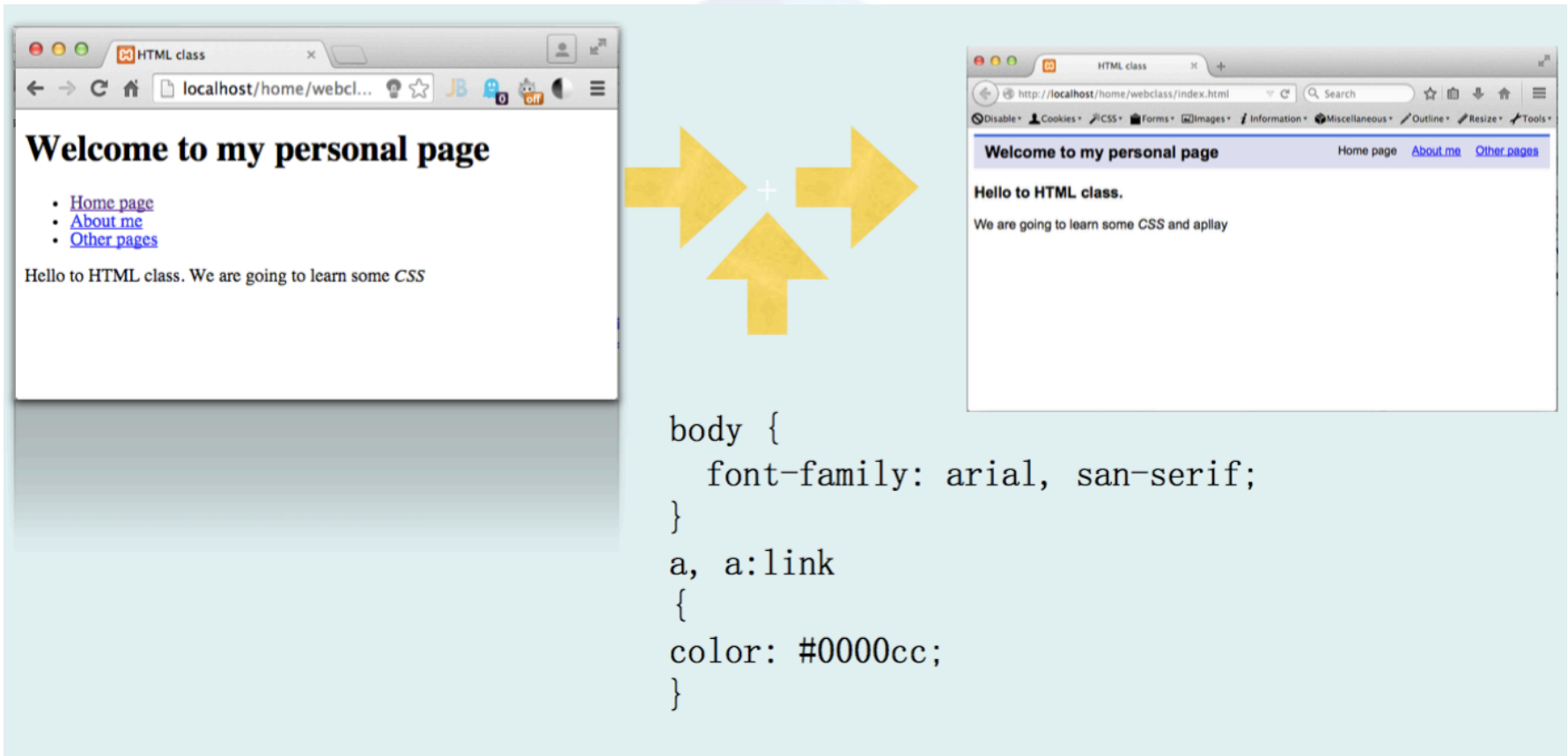
</u>

</i>



Introduction

- Transforming 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>
```

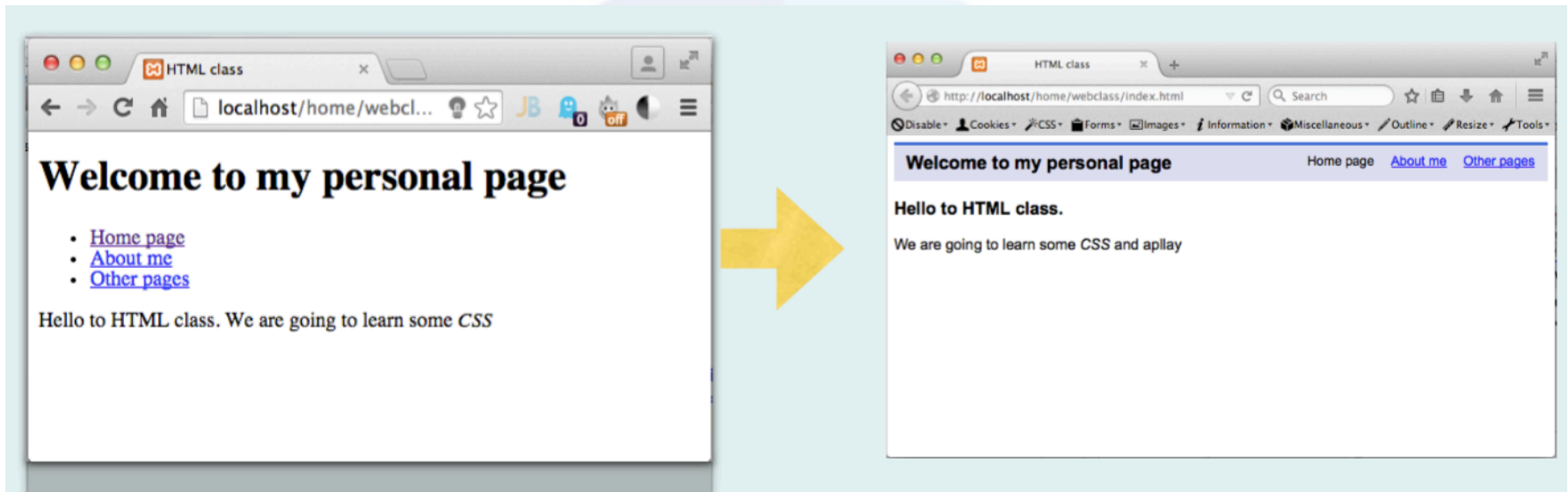
Welcome to my personal page

- [Home page](#)
- [About me](#)
- [Other pages](#)

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

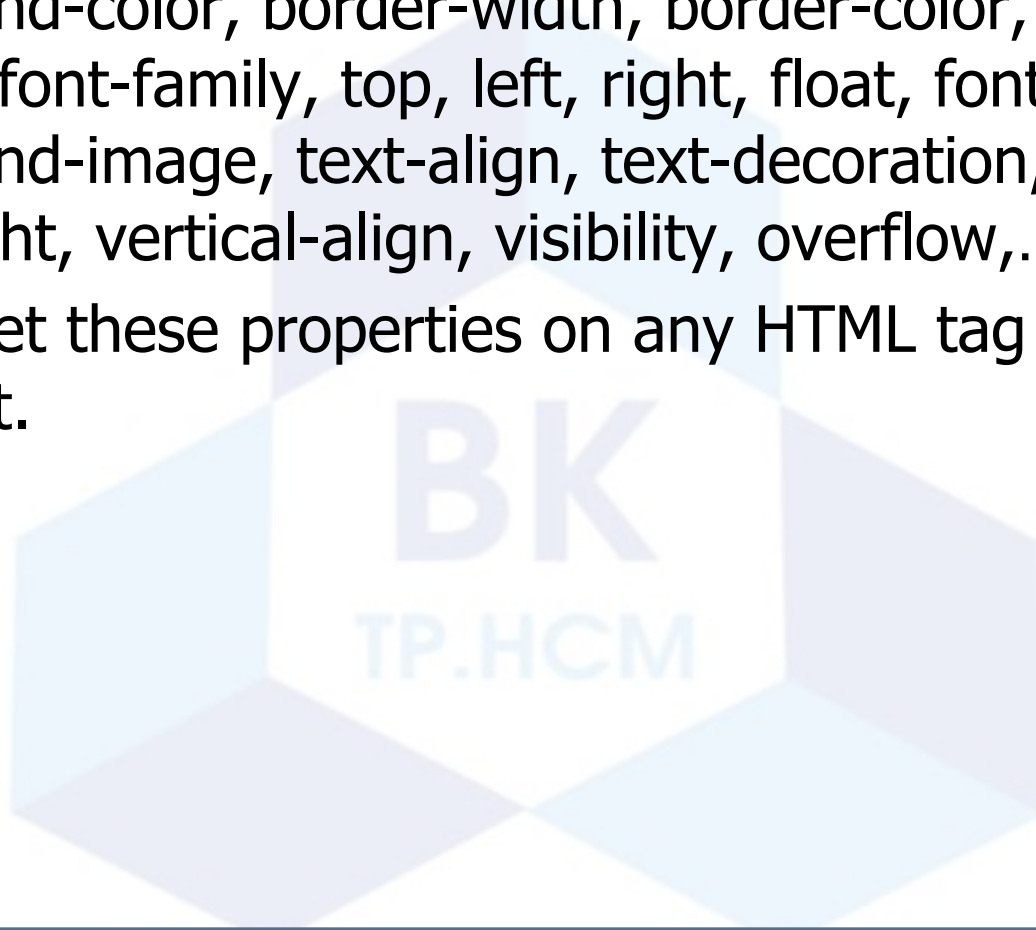
Introduction

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

```
<p class="intro intro2" id="firstname">text</p>
```

```
<span class="intro"></span>
```

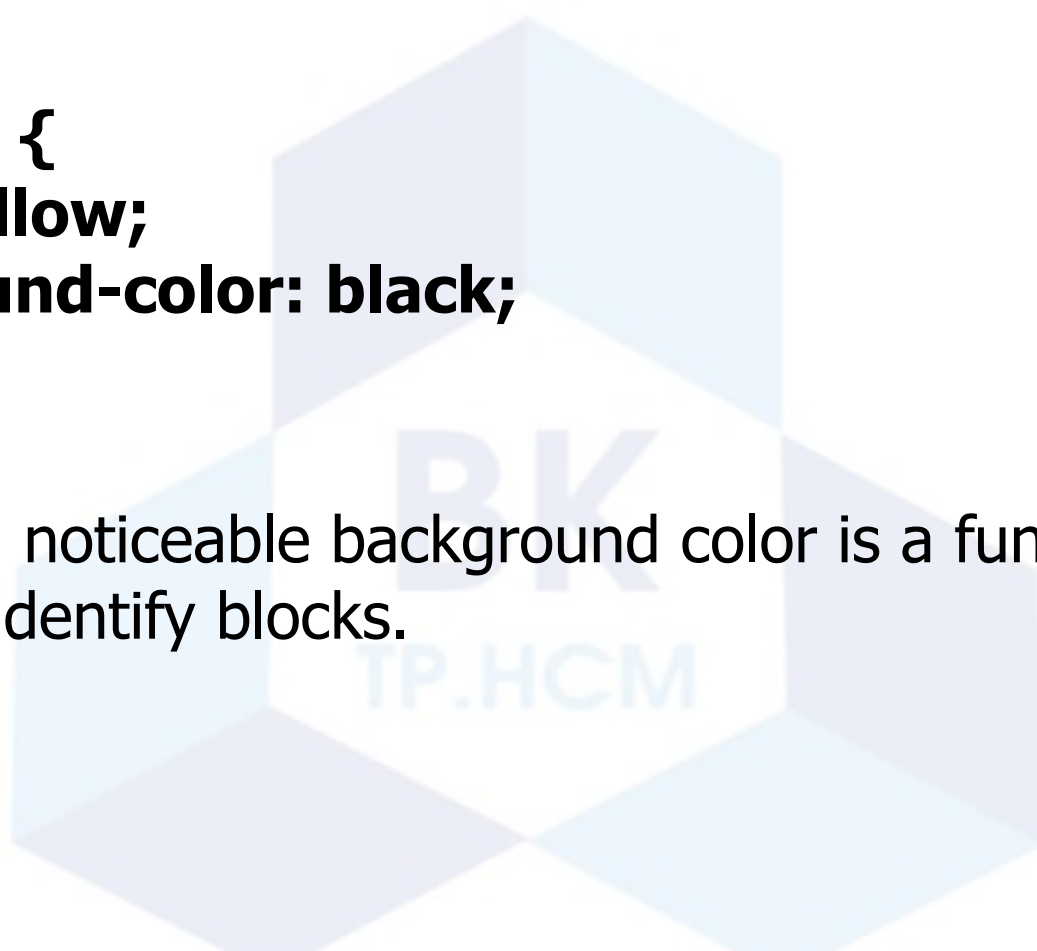


CSS Rules

- Multiple tags with same styling

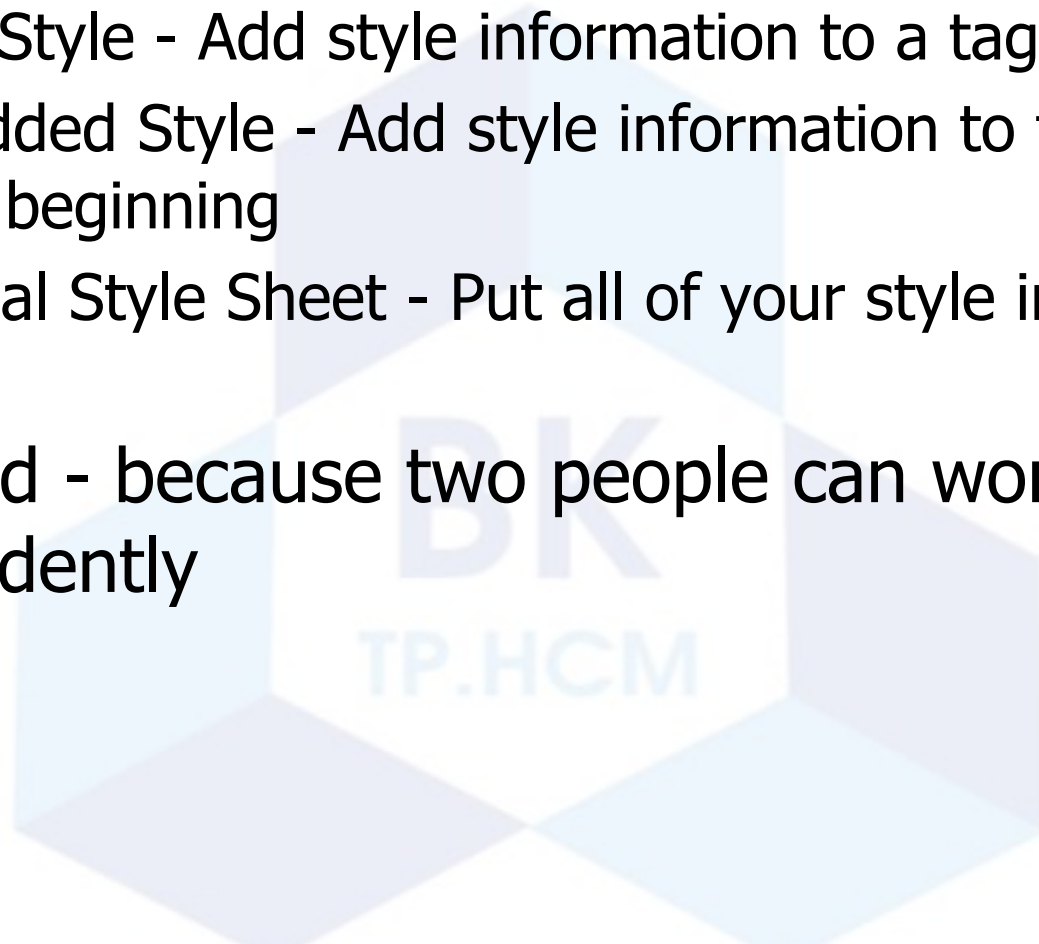
```
h1, h2, h3 {  
  color: yellow;  
  background-color: black;  
}
```

- 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

```
<!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>
```




```
<div class="div1">
  <p id="pid1" class="p1 p2">
    <p class="p2">
      <div class="div2">
        text here
      </div>
    </p>
  </p>
  <p class="p2">
    <div class="p1">
      text here 2
    </div>
  </p>
</div>
```

.div1 > .p2 {...}

p div {...}

.p1 div, .p2 {...}

p#pid1 { color: red }

p > #pid1, p { color: blue}

BK
TP.HCM

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)



Fonts

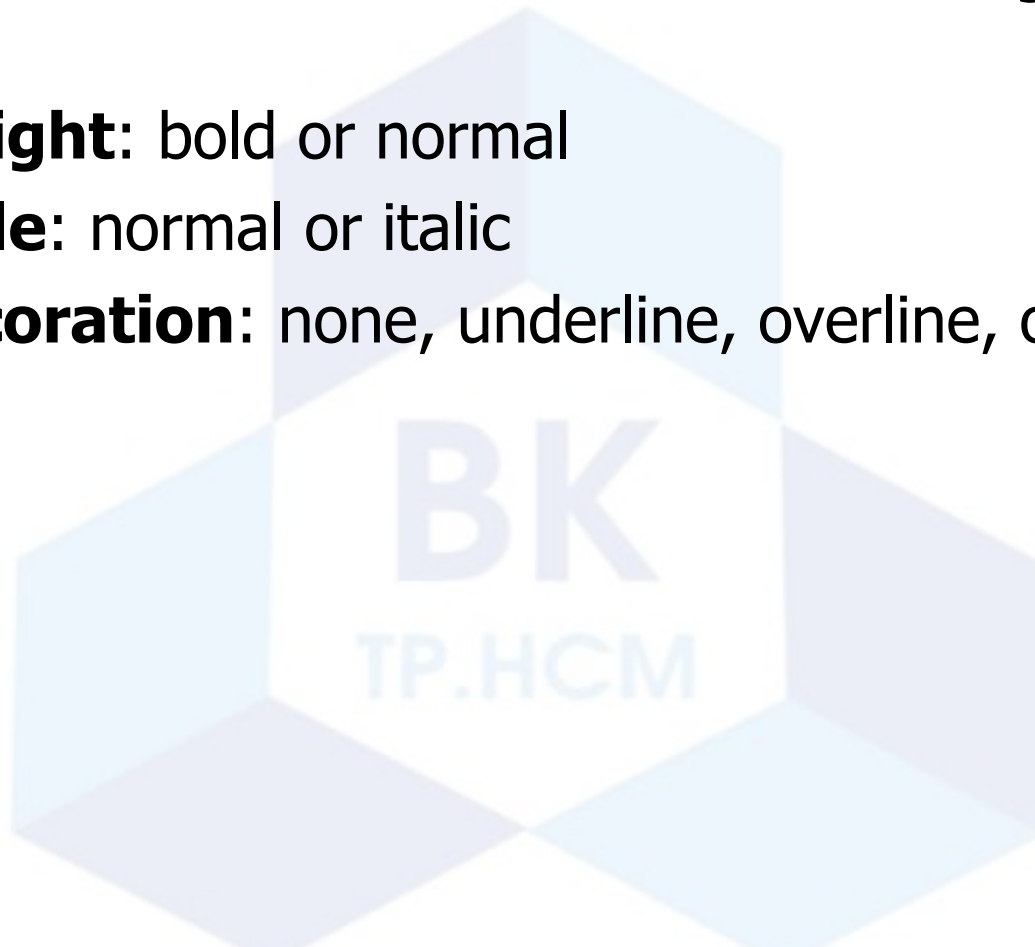
```
body {  
  font-family: "Trebuchet MS, Helvetica, Arial, sans-serif";  
  font-size: x-large;  
}
```

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



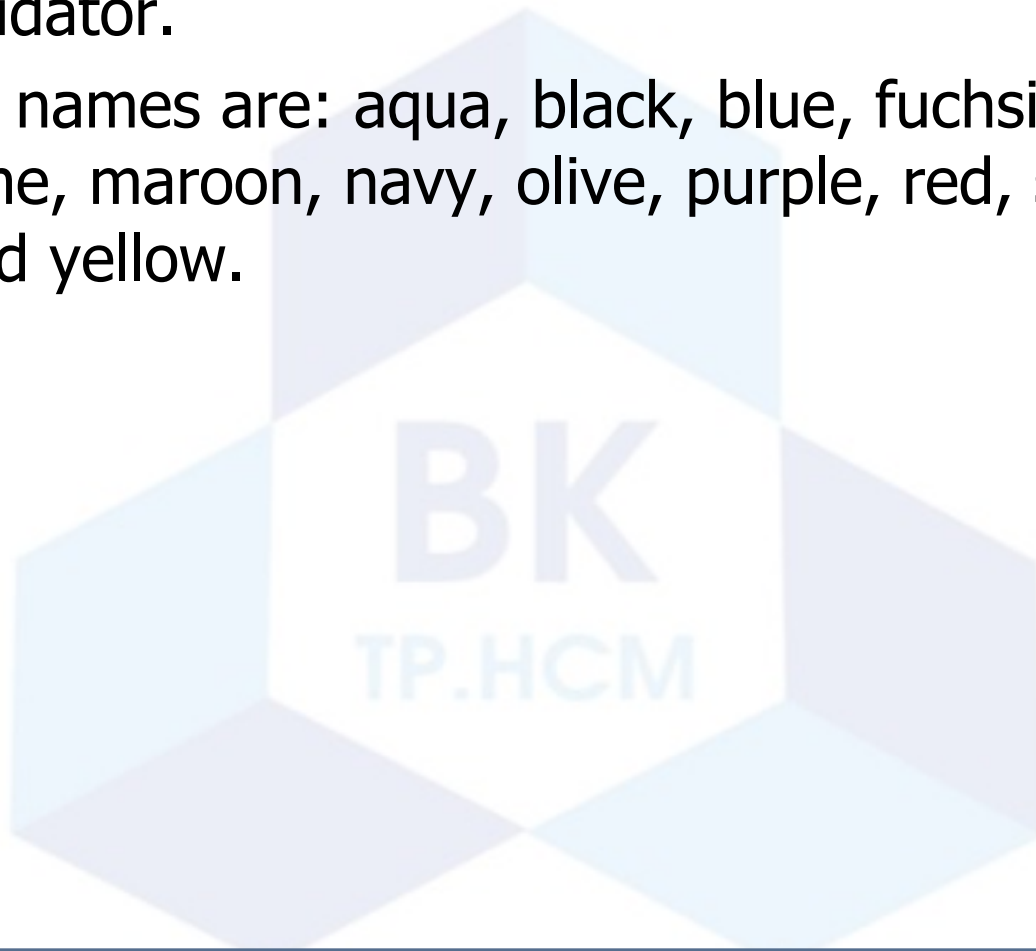
Font Factors

- **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
<u>AliceBlue</u>	<u>#F0F8FF</u>	
<u>AntiqueWhite</u>	<u>#FAEBD7</u>	
<u>Aqua</u>	<u>#00FFFF</u>	
<u>Aquamarine</u>	<u>#7FFFD4</u>	
<u>Azure</u>	<u>#F0FFFF</u>	
<u>Beige</u>	<u>#F5F5DC</u>	
<u>Bisque</u>	<u>#FFE4C4</u>	
<u>Black</u>	<u>#000000</u>	
<u>BlanchedAlmond</u>	<u>#FFEBCD</u>	
<u>Blue</u>	<u>#0000FF</u>	
<u>BlueViolet</u>	<u>#8A2BE2</u>	
<u>Brown</u>	<u>#A52A2A</u>	

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

link - trước khi truy cập: Trạng thái này áp dụng cho các liên kết chưa từng được người dùng nhấp vào hoặc truy cập trước đó. Khi một liên kết ở trạng thái :link, nó sẽ hiển thị theo kiểu dáng CSS mà bạn đã định nghĩa cho trạng thái này. Ví dụ: bạn có thể đặt màu sắc, gạch chân, hoặc kiểu chữ khác cho các liên kết chưa được truy cập.

visited - sau khi đã được truy cập: Trạng thái này áp dụng cho các liên kết mà người dùng đã nhấp vào và truy cập thành công trang web được liên kết đến. Sau khi một liên kết đã được truy cập, trình duyệt sẽ ghi nhớ điều này và áp dụng kiểu dáng CSS mà bạn đã định nghĩa cho trạng thái :visited. Mục đích chính của :visited là để người dùng dễ dàng nhận biết các liên kết họ đã xem qua, thường bằng cách thay đổi màu sắc để phân biệt với các liên kết chưa truy cập.

hover - khi chuột của bạn ở trên nó nhưng bạn chưa nhấp: Trạng thái này kích hoạt khi con trỏ chuột của người dùng di chuyển đến và nằm trên một phần tử (thường là liên kết hoặc nút), nhưng người dùng chưa thực hiện thao tác nhấp chuột. :hover thường được sử dụng để tạo hiệu ứng tương tác, giúp người dùng nhận biết phần tử đó là có thể tương tác được. Ví dụ, bạn có thể thay đổi màu nền, màu chữ, hoặc thêm hiệu ứng bóng đổ khi chuột di chuyển qua liên kết hoặc nút.

active - bạn đã nhấp vào nó và bạn vẫn chưa thấy trang mới: Trạng thái này xảy ra ngay khi người dùng bắt đầu nhấp chuột vào một phần tử (thường là liên kết hoặc nút) và giữ chuột, trước khi nhả chuột ra và trang mới được tải. :active thường được sử dụng để cung cấp phản hồi trực quan tức thì cho người dùng khi họ tương tác với phần tử, ví dụ như thay đổi màu nền đậm hơn hoặc tạo hiệu ứng "ấn xuống" để cho thấy rằng thao tác nhấp chuột đã được ghi nhận. Trạng thái này chỉ tồn tại trong một khoảng thời gian rất ngắn, trong quá trình nhấp chuột.

div: Đặc điểm "phá vỡ sự căn chỉnh" (breaks justification): Điều này có nghĩa là khi bạn sử dụng thẻ <div>, nó sẽ tự động tạo một dòng mới trước và sau nó. Các phần tử nằm trong thẻ <div> sẽ hiển thị trên một dòng riêng, và các phần tử khác sẽ không thể "chen" vào bên cạnh nó theo chiều ngang. Nói cách khác, <div> chiếm toàn bộ chiều rộng có sẵn trong phần tử cha của nó theo mặc định.

Tags and Attributes

div dùng để:

Chia bố cục trang: Tạo các phần chính của trang web như header, footer, sidebar, content area, v.v.
Nhóm các phần tử liên quan: Gộp các phần tử HTML lại thành một nhóm để dễ dàng áp dụng cùng một kiểu dáng hoặc hiệu ứng CSS.

Tạo cấu trúc khối: Xây dựng bố cục trang dạng cột, lưới, hoặc các bố cục phức tạp khác.

- As CSS was introduced, they introduced two new tags that are pretty much there to serve as handles for styling

thẻ không tự động tạo dòng mới. Nó chỉ chiếm không gian vừa đủ cho nội dung bên trong nó. Các phần tử sẽ nằm trên cùng một dòng với các phần tử nội tuyến khác (bao gồm cả văn bản thông thường) và có thể "chen" vào bên cạnh nhau theo chiều ngang.

- <div> - A block tag (breaks justification)
- - An inline tag that does not break justification

span dùng để: Định kiểu cho một phần nhỏ của văn bản: Ví dụ, bạn muốn thay đổi màu sắc, kiểu chữ, hoặc kích thước của một vài từ trong một đoạn văn bản.

Áp dụng kiểu dáng cho một phần tử nội tuyến: Ví dụ, bạn muốn thay đổi màu nền của một biểu tượng nhỏ hoặc một đoạn văn bản ngắn nằm trong dòng.

Nhóm các phần tử nội tuyến: Gộp các phần tử nội tuyến lại thành một nhóm để dễ dàng áp dụng cùng một kiểu dáng CSS.

- 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
#footer {
  font-style: italic;
  font-family: Times, serif;
}
```

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

```
<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>
```

This is a paragraph.

This looks like a paragraph, but it's actually a div.

This is another paragraph.

This is another div.

Think

This is a paragraph.

This looks like a paragraph, but it's actually a div.

This is another paragraph.

This is another 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.

bạn nên áp dụng kiểu dáng trực tiếp lên thẻ khối chứa nó (ví dụ: <p>, <div>, <h1>, v.v.).

Giá trị margin: auto; có thể được sử dụng để căn giữa phần tử khối theo chiều ngang (thường dùng với width được xác định).

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

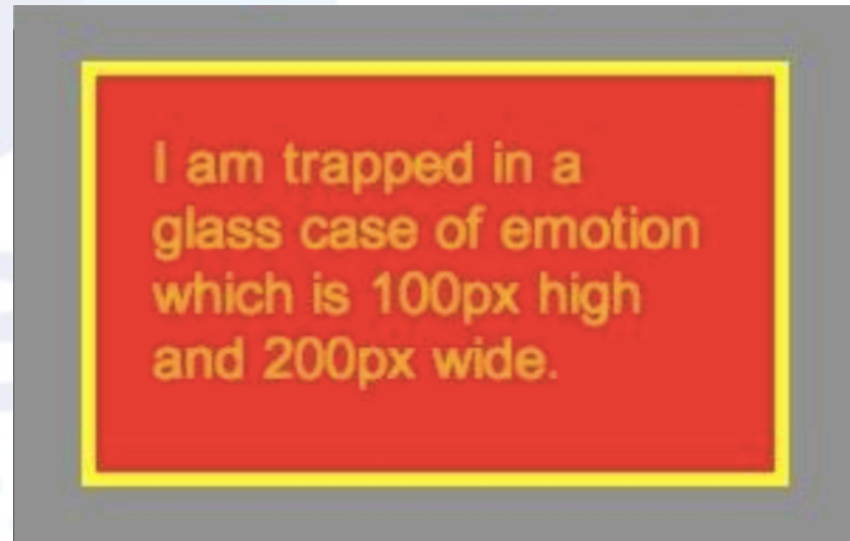
CSS Box Model

```
<p class="trapped">
```

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

```
</p>
```

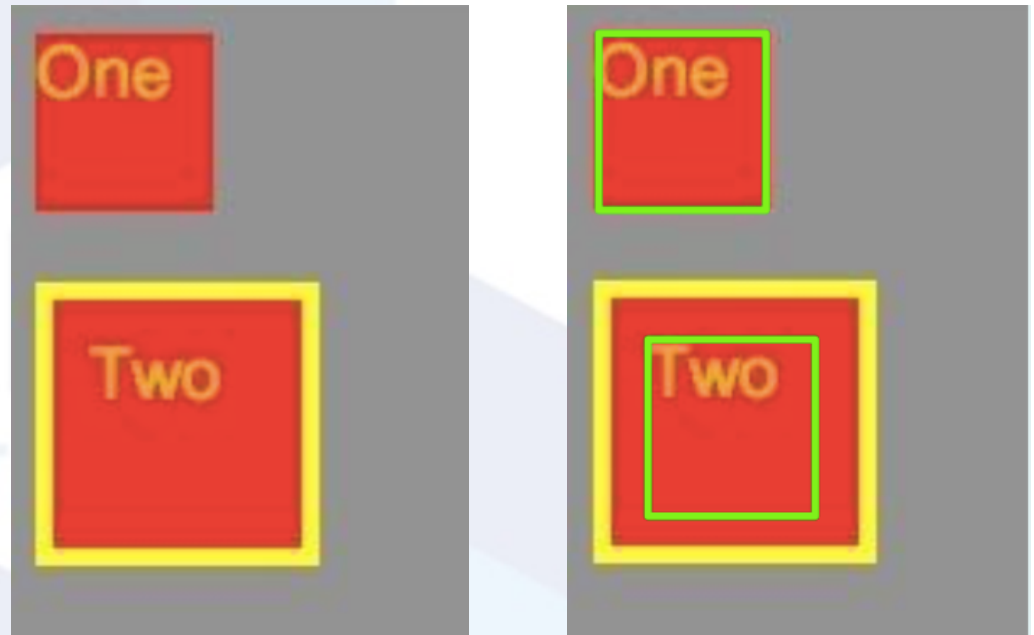
```
.trapped {  
  height: 100px;  
  width: 200px;  
  margin: 20px;  
  border: 5px solid yellow;  
  background:red;  
  padding: 20px;  
  font-family:Arial;  
  color:orange;  
  font-size:20px;  
}
```



CSS Box Model

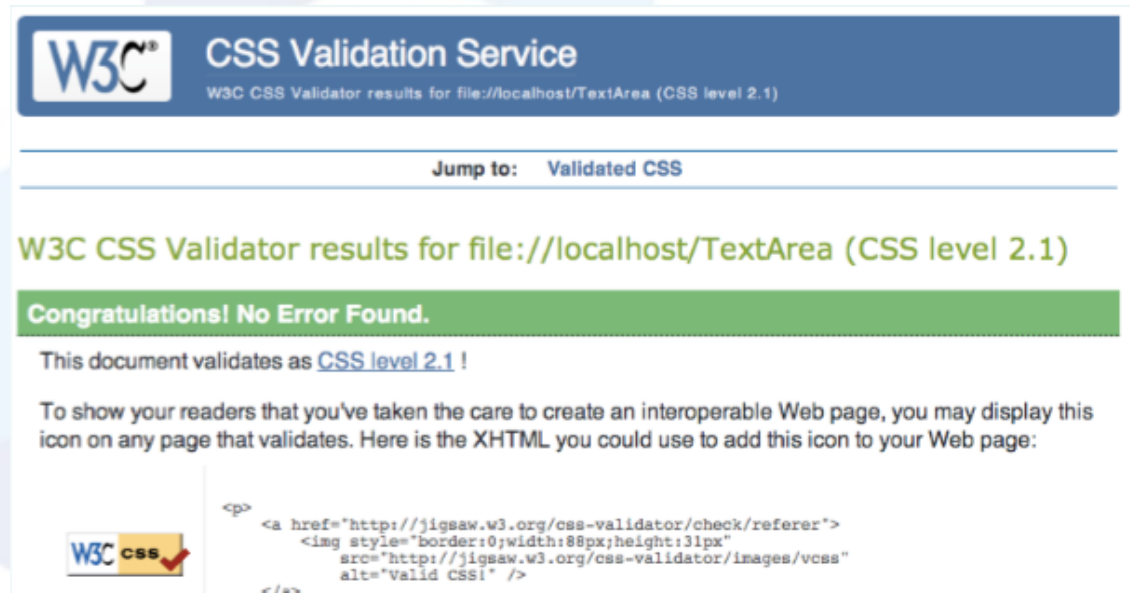
```
<p class="trapped">One</p>  
<p class="trapped2">Two</p>
```

```
.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

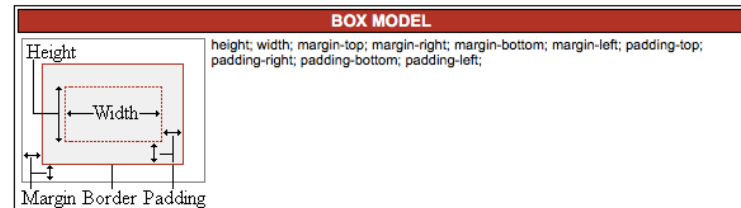
- <http://www.lesliefranke.com/files/reference/csscheatsheet.html>

SYNTAX	
Syntax	
selector {property: value;}	
External Style Sheet	
<link rel="stylesheet" type="text/css" href="style.css" />	
Internal Style	
<style type="text/css">	
selector {property: value;}	
</style>	
Inline Style	
<tag style="property: value">	

GENERAL	
Class	String preceded by a period
ID	String preceded by a hash mark
div	Formats structure or block of text
span	Inline formatting
color	Foreground color
cursor	Appearance of the cursor
display	block; inline; list-item; none
overflow	How content overflowing its box is handled
	visible; hidden; scroll; auto
visibility	visible; hidden

FONT	
font-style	Italic, normal
font-variant	normal, small-caps
font-weight	bold, normal, lighter, bolder, integer (100-900)
font-size	Size of the font
font-family	Specific font(s) to be used

TEXT	
letter-spacing	Space between letters
line-height	Vertical distance between baselines
text-align	Horizontal alignment
text-decoration	blink, line-through, none, overline, underline
text-indent	First line indentation
text-transform	capitalize, lowercase, uppercase
vertical-align	Vertical alignment
word-spacing	Spacing between words



BORDER	
border-width	Width of the border
border-style	dashed; dotted; double; groove; inset; outset; ridge; solid; none
border-color	Color of the border

POSITION	
clear	Any floating elements around the element? both, left, right, none
float	Floats to a specified side left, right, none
left	The left position of an element auto, length values (pt, in, cm, px)
top	The top position of an element auto, length values (pt, in, cm, px)
position	static, relative, absolute
z-index	Element above or below overlapping elements? auto, integer (higher numbers on top)

BACKGROUND	
background-color	Background color
background-image	Background image
background-repeat	repeat, no-repeat, repeat-x, repeat-y
background-attachment	Background image scroll with the element? scroll, fixed
background-position	(x y), top, center, bottom, left, right

LIST	
list-style-type	Type of bullet or numbering in the list disc; circle; square; decimal; lower-roman; upper-roman; lower-alpha; upper-alpha; none
list-style-position	Position of the bullet or number in a list inside; outside
list-style-image	Image to be used as the bullet in a list

Layout

Side content...	Main 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

float (Thuộc tính Float - "trôi nổi")

Di chuyển phần tử ra khỏi dòng chảy thông thường (Moving it out of the normal flow): Khi bạn "float" một phần tử, bạn loại bỏ nó khỏi dòng chảy tài liệu thông thường. Dòng chảy tài liệu thông thường là cách trình duyệt sắp xếp các phần tử HTML trên trang theo thứ tự chúng xuất hiện trong mã nguồn (từ trên xuống dưới, từ trái sang phải đối với các ngôn ngữ đọc từ trái sang). Khi một phần tử bị float, nó không còn chiếm chỗ trong dòng chảy này theo cách thông thường nữa.

Các giá trị của float:

left: Phần tử sẽ trôi về bên trái của vùng chứa của nó. Các nội dung khác (văn bản, hình ảnh, các phần tử khác) sẽ bao quanh nó ở bên phải (nếu có đủ không gian).

right: Phần tử sẽ trôi về bên phải của vùng chứa. Các nội dung khác sẽ bao quanh nó ở bên trái.

none: (Giá trị mặc định) Phần tử không bị float, nó vẫn nằm trong dòng chảy tài liệu thông thường.

Hiệu ứng lên các phần tử khác (Elements that follow a floated element will move up): Các phần tử HTML xuất hiện sau một phần tử bị float trong mã nguồn sẽ "tràn lên" và cố gắng đặt bên cạnh phần tử float đó, nếu có đủ không gian. Điều này có nghĩa là văn bản hoặc các phần tử nội tuyến có thể bao quanh phần tử float, tạo ra các bố cục dạng cột hoặc bố cục mà văn bản ôm lấy hình ảnh.

clear (Thuộc tính Clear - "xóa float")

Ngăn chặn phần tử di chuyển lên cạnh phần tử float (Stop elements moving up next to a floated element): Thuộc tính clear cho phép bạn ngăn chặn một phần tử HTML di chuyển lên và đặt bên cạnh một phần tử đã bị float. Nói cách khác, clear "xóa bỏ" hiệu ứng float từ các phần tử trước đó, buộc phần tử hiện tại phải hiển thị ở một vị trí mới, bên dưới các phần tử float.

Các giá trị của clear:

left: Xóa float bên trái. Phần tử sẽ được đẩy xuống dưới bất kỳ phần tử nào đang float về bên trái ở phía trên nó.

right: Xóa float bên phải. Phần tử sẽ được đẩy xuống dưới bất kỳ phần tử nào đang float về bên phải ở phía trên nó.

both: Xóa cả float bên trái và bên phải. Phần tử sẽ được đẩy xuống dưới bất kỳ phần tử nào đang float về cả hai phía ở phía trên nó.

none: (Giá trị mặc định) Không xóa float. Phần tử có thể di chuyển lên cạnh các phần tử float nếu có không gian.

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

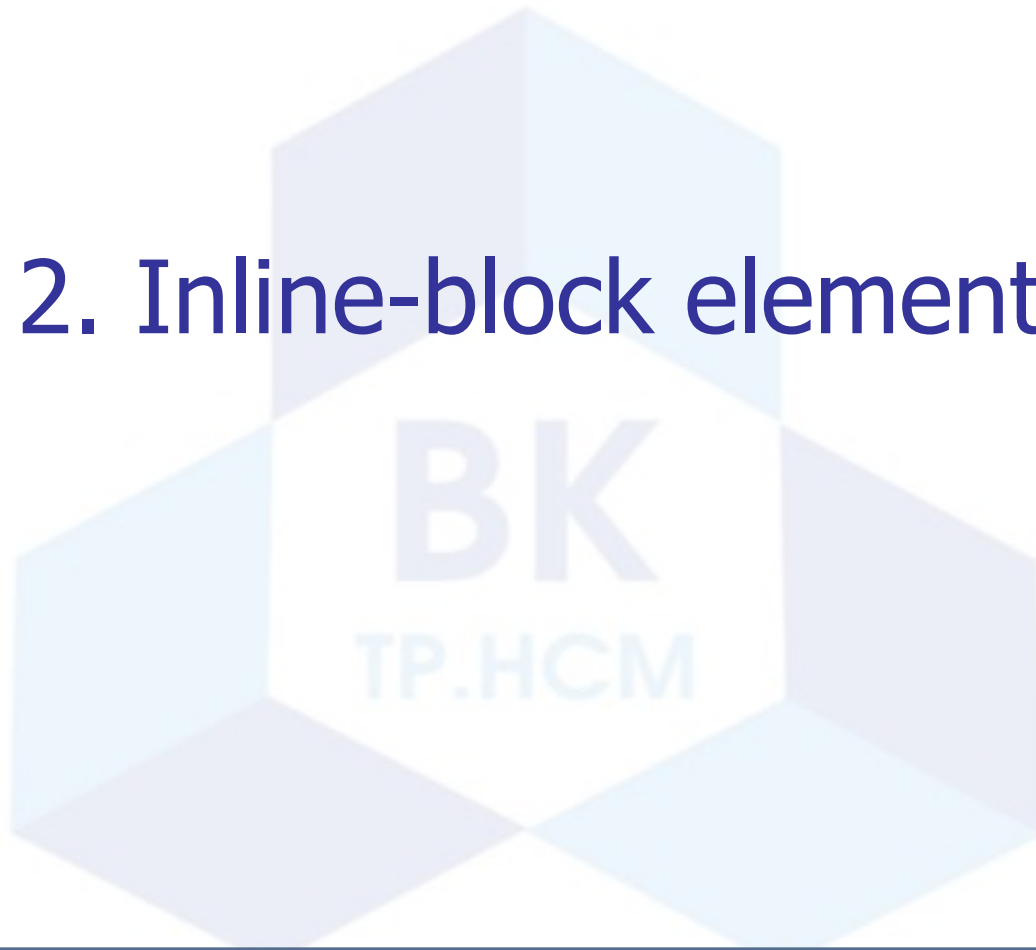
```
<div class="row">
  <div class="column side">
    <h2>Side</h2>
    <p>content...</p>
  </div>

  <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

Giống như inline elements nhưng có chiều rộng và chiều cao (like inline elements but they can have a width and height): Đây là điểm mấu chốt để hiểu về inline-block. Phần tử inline-block kết hợp các đặc tính tốt nhất của cả phần tử inline và phần tử block:

Giống Inline: Giống như phần tử inline, phần tử inline-block không tạo ra dòng mới. Chúng sẽ nằm trên cùng một dòng với các phần tử khác (cả inline và inline-block) nếu có đủ không gian. Chúng chỉ chiếm vừa đủ chiều rộng cần thiết cho nội dung của chúng.

Giống Block: Tuy nhiên, khác với phần tử inline, phần tử inline-block cho phép bạn thiết lập width (chiều rộng) và height (chiều cao). Phần tử inline thông thường thì không thể thiết lập chiều rộng và chiều cao một cách trực tiếp.

Sử dụng thuộc tính display (Using display property): Để biến một phần tử HTML thành phần tử inline-block, bạn sử dụng thuộc tính CSS display và đặt giá trị của nó là inline-block.

2. Inline-block elements

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>
```

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






3. flexbox

The logo of the University of Bucheon (BK) in Ho Chi Minh City (TP.HCM) is visible as a large, light blue watermark in the background. It consists of a hexagonal shape composed of six smaller hexagons, with the letters 'BK' and 'TP.HCM' in the center.

BK
TP.HCM

3. flexbox

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

				
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/>

Flexible Box Layout Module (Mô-đun bố cục hộp linh hoạt): Flexbox là một mô-đun bố cục mạnh mẽ trong CSS3. Nó được thiết kế để làm cho việc thiết kế các cấu trúc bố cục linh hoạt và đáp ứng trở nên dễ dàng hơn, đặc biệt là trong các ứng dụng web phức tạp và trên nhiều loại thiết bị khác nhau.

Dễ dàng thiết kế bố cục linh hoạt và đáp ứng (Easier to design flexible responsive layout structure): Flexbox giúp giải quyết nhiều vấn đề bố cục phức tạp mà trước đây thường phải dùng các kỹ thuật như float hoặc positioning (định vị) để xử lý. Với Flexbox, bạn có thể dễ dàng:

Căn chỉnh các phần tử: Căn chỉnh các phần tử con bên trong một container theo chiều ngang hoặc chiều dọc, hoặc cả hai.

Phân phối không gian: Phân phối không gian trống giữa các phần tử con một cách linh hoạt.

Sắp xếp lại thứ tự: Thay đổi thứ tự hiển thị của các phần tử con mà không cần thay đổi cấu trúc HTML.

Tạo bố cục đáp ứng (responsive): Dễ dàng điều chỉnh bố cục để phù hợp với các kích thước màn hình khác nhau (ví dụ: từ máy tính để bàn đến điện thoại di động).

Không cần dùng float hoặc positioning (without using float or positioning): Một trong những ưu điểm lớn nhất của Flexbox là nó giảm thiểu sự cần thiết phải sử dụng các thuộc tính float và positioning cho mục đích bố cục. float và positioning có thể khá phức tạp và đôi khi gây ra các vấn đề bố cục khó lường. Flexbox cung cấp một cách tiếp cận trực quan và dễ quản lý hơn cho việc tạo bố cục.

3. flexbox

Side	Main Content
content...	content

```
<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>
```

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






4. Grid

The logo is a large, light blue watermark centered on the slide. It consists of a hexagon formed by six smaller hexagons. In the center of this hexagon, the letters 'BK' are written in a large, bold, sans-serif font, with 'TP.HCM' written in a smaller, regular sans-serif font directly below it.

BK
TP.HCM

4. Grid

- CSS Grid Layout (aka “Grid” or “CSS Grid”), is a two-dimensional grid-based layout system that, compared to any web layout system of the past, completely changes the way we design user interfaces.

				
57.0	16.0	52.0	10	44

- More about flexbox:
 - <https://css-tricks.com/snippets/css/complete-guide-grid/>

CSS Grid Layout (Bố cục lưới CSS): CSS Grid Layout là một hệ thống bố cục dựa trên lưới hai chiều (two-dimensional). Điều này có nghĩa là Grid cho phép bạn kiểm soát bố cục trang web cả theo chiều ngang (hàng) và chiều dọc (cột) cùng một lúc.

Thay đổi hoàn toàn cách thiết kế giao diện người dùng (Completely changes the way we design user interfaces): So với bất kỳ hệ thống bố cục web nào trước đây (bao gồm cả float, inline-block và flexbox), CSS Grid mang đến một cách tiếp cận hoàn toàn mới và mạnh mẽ hơn để thiết kế giao diện người dùng (UI). Grid giúp bạn tạo ra các bố cục phức tạp, tinh tế và có cấu trúc rõ ràng một cách dễ dàng và hiệu quả.

Hệ thống dựa trên lưới (Grid-based layout system): Điểm cốt lõi của CSS Grid là khái niệm lưới (grid). Bạn định nghĩa một cấu trúc lưới trên container (vùng chứa) của mình, và sau đó bạn có thể đặt các phần tử con vào các ô (cells) trên lưới đó. Lưới được tạo thành từ các hàng (rows) và cột (columns).

So sánh với Flexbox: Trong khi Flexbox chủ yếu là hệ thống bố cục một chiều (tốt cho việc sắp xếp các phần tử theo hàng HOẶC cột), Grid là hệ thống hai chiều, cho phép bạn kiểm soát bố cục cả theo hàng VÀ cột đồng thời. Điều này làm cho Grid đặc biệt phù hợp cho việc tạo ra các bố cục trang web tổng thể phức tạp, trong khi Flexbox thường được sử dụng tốt hơn cho việc bố cục các thành phần nhỏ hơn bên trong trang.

4. Grid

Side	Main Content
content...	content

```
<div class="grid-container">
  <div class="grid-item">
    <h2>Side</h2>
    <p>content...</p>
  </div>
  <div class="grid-item">
    <h2>Main Content</h2>
    <p>content...</p>
  </div>
</div>
```

```
.grid-container {
  display: grid;
  grid-template-columns: 200px 400px;
}
.grid-item {
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 10px
}
```

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>

