



# PHÁT TRIỂN ỨNG DỤNG WEB

---

## BÀI 4\_1

### CSS (Casscading Style Sheets)



# Nội dung

---



- **Giới thiệu CSS**
- **Định nghĩa Style**
- **Sử dụng và Phân loại CSS**
- **Selector trong CSS và phạm vi ảnh hưởng**



# Lợi ích khi dùng CSS



- Thời khóa biểu quả khoa HTTT

	Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7	CN
Tiết 1	TKWeb 1			LTĐT			TK CSDL
Tiết 2	HĐH	TK CSDL	TKWeb 1		TKWeb 1	CSDL	
Tiết 3	CSDL	LTĐT	HĐH	TK CSDL		LTĐT	TKWeb 1
Tiết 4			CSDL		HĐH		



# Giới thiệu về CSS

---



- CSS = Cascading Style Sheets
- Dùng để mô tả cách hiển thị các thành phần trên trang WEB
- Sử dụng tương tự như dạng TEMPLATE
- Có thể sử dụng lại cho các trang web khác
- Có thể thay đổi thuộc tính từng trang hoặc cả site nhanh chóng (cascading)





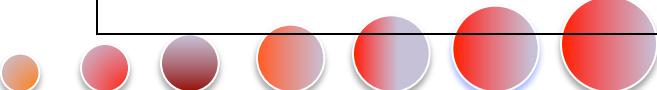
# Định nghĩa Style

Dạng 1:

```
<tag style=
  “property1:value1;
  property2:value2;
  .....
  propertyN:valueN;”>...</tag>
```

Ví dụ:

```
<h1 style="color : blue; font-family : Arial;"> ĐHCNTT
</h1>
```



# Định nghĩa Style (tt)

---



Dạng 2:

**H2{**

**FONT-WEIGHT: bold;**  
**FONT-SIZE: 16pt;**  
**COLOR: white;**  
**FONT-STYLE: italic;**  
**FONT-FAMILY: Arial;**  
**BACKGROUND-COLOR: red;**  
**font-color: white**

**}**



# Định nghĩa Style



Dạng 3:

```
SelectorName{  
    property1:value1;  
    property2:value2;  
    .....  
    propertyN:valueN;}  
  
<tag class =  
    “SelectorName”>  
    .....  
</tag>
```

```
.TieuDe1 {  
    color: red;  
    font-family: Verdana, sans-  
    serif; }  
<h1 class=“TieuDe1”>  
ĐHCNTT </h1>
```



# Ghi chú trong style

---



- Giống Ghi chú trong C++
- Sử dụng /\*Ghichú\*/
- Ví dụ:

**SelectorName**

```
{      property1:value1; /*Ghichu1*/  
      property2:value2; /*Ghichu2*/  
.....  
      propertyN:valueN;  
}
```



# Phân loại CSS

---



## Gồm 3 loại CSS

- 1. Inline Style Sheet
- 2. Embedding Style Sheet
- 3. External Style Sheet



# Inline style Sheet

- Định nghĩa style trong thuộc tính style của từng tag HTML
- Theo cú pháp **dạng 1**

```
<tag style="property1: value1;  
...property N:value N;">...  
</tag>
```

Ví dụ:

```
<H1 STYLE="color: yellow">This is yellow </H1>
```



# Embedding Style Sheet



- Nhúng trong tag <style> của trang HTML
- **Định nghĩa style theo cú pháp Dạng 2**

```
<head>
<style type="text/css" media="all | print | screen" >
<!--
    TagName{
        property 1:value1;
        property 2:value2;
        .....
        property N: valueN;
    }
....>
</style>
</head>
```



# Ví dụ



```
<HTML><HEAD>  
<TITLE>Embedded Style Sheet  
</TITLE><STYLE TYPE="text/css">  
<!--  
P{color: green;font-size: 12pt;font-family: Arial;}  
H2{color: Red;}  
--></STYLE>  
</HEAD>  
<BODY BGCOLOR="#FFFFFF"><H2>This is red</H2>  
<P>this is green, 12 pt. and Garamond.</P>  
</BODY></HTML>
```



# External Style Sheet



- Mọi style đều lưu trong file có phần mở rộng là \*.CSS (được sử dụng phổ biến)
- Định nghĩa style theo cú pháp **dạng 2,3**
- Tạo liên kết đến file CSS
  - 1.Trong trang HTML: Liên kết bằng tag link.

Cú pháp:

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



# External Style Sheet (tt)



2.Trong trang HTML: Liên kết bằng tag style với @import url.

## Cú pháp

```
<head>  
  
<style type="text/css" media="all | print | screen" >  
    @import url(URL);  
</style>  
  
</head>
```



# Ví dụ

---

- Trong file style.CSS

```
H2{
```

```
    FONT-WEIGHT: bold;
```

```
    FONT-SIZE: 16pt;
```

```
    COLOR: white;
```

```
    FONT-STYLE: italic;
```

```
    FONT-FAMILY: Arial;
```

```
    BACKGROUND-COLOR: red;
```

```
    font-color: white
```

```
}
```



# Ví dụ (tt)



- Sử dụng style.CSS

```
<html>
<head><title>Cascading Style Sheets
</title>
<link REL="stylesheet" HREF="style.css" type="text/css">
</head>
<body>
    <h2>This is an H2 </h2>
</body>
</html>
```



# So sánh, đánh giá

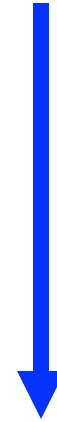
	Inline style sheet	Embedding style sheet	External style sheet
Khai báo	Kiểu 1	Kiểu 2	Kiểu 3
Cú pháp	<pre>&lt;p style="color:red;"&gt;     ĐHCNTT &lt;/p&gt;</pre>	<pre>&lt;style type="text/ css"&gt; .tieude1{color=r ed;} &lt;/style&gt; &lt;p class="tieude1"&gt;     ĐHCNTT &lt;/p&gt;</pre>	<pre>&lt;link rel="stylesheet" href="style.css"&gt; &lt;p class ="tieude1"&gt;     ĐHCNTT &lt;/p&gt;</pre>
Ưu điểm	Dễ quản lý style theo từng tag	<ul style="list-style-type: none"> <li>+ Dễ quản lý style theo từng tài liệu web</li> <li>+ Không cần thêm các trang thông tin khác cho style</li> </ul>	<ul style="list-style-type: none"> <li>+ Thiết lập style cho nhiều tài liệu</li> <li>+ Thông tin các style được trình duyệt cache lại</li> </ul>
Khuyết điểm	Cần khai báo style trong từng tag của tài liệu	Cần khai báo lại style lại cho các trang khác	<ul style="list-style-type: none"> <li>+ Tốn thời gian download file .css -&gt; làm chậm quá trình biên dịch web ở trình duyệt trong lần đầu tiên sử dụng</li> </ul>

# Độ ưu tiên



➤ Thứ tự độ ưu tiên áp dụng định dạng style dùng trong các trang web (Độ ưu tiên giảm dần)

- 1. Inline style sheet
- 2. Embedding style sheet
- 3. External style sheet
- 4. Browser Default



# Selector trong CSS và phạm vi ảnh hưởng

- Selector

- Là tên 1 style tương ứng với một thành phần được áp định dạng
- Ví dụ:

```
.TieuDe1 {  
    color: red;  
    font-family: Verdana, sans-serif;  
}
```

```
<h1 class="TieuDe1"> ĐHCNTT</h1>
```



# Selector trong css



Selector	Ví dụ	Mô tả
*	*	Chọn tất cả các thành phần html
tag	P	Chọn tất cả các thành phần cùng loại tag
tag01 tag02	div p	Chọn tất cả thành phần <tag02> bên trong thành phần <tag01>.
tag01,tag02	div,p	Chọn tất cả thành phần <tag01> và <tag02>.
tag01+tag02	div+p	Chọn tất cả thành phần <tag02> được đặt kế và sau thành phần <tag01>.
tag01>tag02	div>p	Chọn tất cả thành phần <tag02> có thành phần cha là <tag01>.
.class	.className	Chọn tất cả các thành phần có cùng tên class.
#id	#idName	Chọn tất cả các thành phần có cùng tên id.



# Selector trong css (tt)



Selector	Ví dụ	Mô tả
:active	a:active	Chọn các liên kết được kích hoạt.
:after	p:after	Thêm nội dung ngay phía sau thành phần.
:before	p:before	Thêm nội dung ngay phía trước thành phần.
:first-child	p:first-child	Chọn thành phần đầu tiên của thành phần cha chứa nó.
:first-letter	p:first-letter	Chọn ký tự đầu tiên của một thành phần.
:first-line	p:first-line	Chọn dòng đầu tiên của một thành phần.
:focus	input:focus	Thành phần sẽ focus khi được chọn.
:hover	a:hover	Chọn các liên kết được hover (di chuyển chuột lên thành phần).
:lang(mã ngôn ngữ)	p:lang(vi)	Chọn thành phần với thuộc tính ngôn ngữ có giá trị bắt đầu với "mã ngôn ngữ".



# Selector trong css (tt)



Selector	Ví dụ	Mô tả
:link	a:link	Định dạng cho tất cả liên kết khi chưa được click.
:visited	a:visited	Định dạng cho các thành phần đã được click.
[Thuộc tính]	[target]	Chọn tất cả các thành phần có cùng thuộc tính (attribute). <a href="#">Ví dụ</a>
[Thuộc tính]=ngôn ngữ	[lang =vi]	Được sử dụng để chọn tất cả các thành phần với thuộc tính có giá trị = "ngôn ngữ". <a href="#">Ví dụ</a>
[Thuộc tính~=giá trị]	[title~="myWeb"]	Chọn tất cả các thành phần với thuộc tính chứa một giá trị. <a href="#">Ví dụ</a>
[Thuộc tính=giá trị]	[target=_blank]	Chọn tất cả các thành phần với thuộc tính bằng giá trị. <a href="#">Ví dụ</a>
[Thuộc tính^=giá trị]	[title^="myWeb"]	Chọn các thành phần bắt đầu bằng "myWeb". <a href="#">Ví dụ</a>



# selector “\*”

```
* {  
    property: value;  
}
```

**Selector \***  
**VNPT**

```
<html>  
<head>  
<script> * {  
    color: #ff0000;  
}  
</script>  
</head>  
<body>  
<p>Selector *</p>  
<div> VNPT  
</div >  
</body>  
</html>
```





# selector theo tên tag

```
p{  
    property: value;  
}
```

**Selector \***  
**VNPT**

```
<html>  
<head>  
<script> * {  
    color: #ff0000;  
}  
</script>  
</head>  
<body>  
<p>Selector *</p>  
<div> VNPT  
</div >  
</body>  
</html>
```

# selector “tag01, tag02”

---

```
<html>
<head>
<script> div,p {
    color: #ff0000;
}
</script>
</head>
<body>
<p>Selector *</p>
<div> VNPT
</div >
</body>
</html>
```

**Selector \***  
**VNPT**



# selector “tag01 + tag02”

```
<body>
```

```
<p>Đây là thành phần p nằm trước  
thành phần div</p>
```

```
<div>Đây là thành phần div</div>
```

```
<p>Đây là thành phần p nằm kế và  
sau thành phần div</p>
```

```
<p>Đây là thành phần p nằm sau  
thành phần div</p>
```

```
</body>
```

```
div+p {  
    color: #ff0000;  
}
```

Đây là thành phần p nằm trước thành phần div

Đây là thành phần div

**Đây là thành phần p nằm kế và sau thành phần div**

Đây là thành phần p nằm sau thành phần div



# selector “tag01 > tag02”

```
<body>
<div>
<p>Đây là thành phần p nằm bên
trong thành phần div</p>
<p>Đây là thành phần p nằm bên
trong thành phần div</p>
</div>
</body>
```

```
div>p {
  color: #ff0000;
}
```

Đây là thành phần p nằm bên trong thành phần div  
Đây là thành phần p nằm bên trong thành phần div

# selector “.class, #id”

```
tag.tênid {  
    property: value;  
}
```

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

```
<body>  
<p id="left"> P left</p>  
<p class="right">P right </p>  
<p class="top">P top</p >  
</body>
```

```
p#left{ color: red; }  
p.right{color: blue;}  
.top{color: green ; }
```

P left  
P right  
P top

# Selector “:after”

```
<body>  
<p>  
tập đoàn viễn thông</p>  
</body>
```

```
p:after{  
    content:"-VNPT"  
}
```

tập đoàn viễn thông-VNPT

# Selector “:last-child”

```
<body>
<ul type="none">
<li>Nokia</li>
<li>Apple</li>
<li>LG</li>
</ul>
</body>
```

```
ul li:last-child{    color:
red;
}
```

Nokia  
Apple  
**LG**

# Selector “:focus”

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

```
/* mouse over link */  
a:hover {  
    color: hotpink;  
}  
/* selected link */  
a:active {  
    color: blue;
```



# Ví dụ về element

```
<http>
<head>
    <style type="text/css">
        P{color:red}
        Em{color:blue}
    </style>
</head>
<body>
    <p>DHCNTT</p>
    <p>He thong thong tin<em>csdl</em></p>
</body>
</http>
```



# Ví dụ ID rules

```
<http>
<head>
    <style type="text/css">
        #id1{color:red}
        #id2{color:blue}
    </style>
</head>
<body>
    <p id="id1">ĐHCNTT</p>
    <p id="id2">He thong thong tin</p>
</body>
</http>
```





# Class rules

---

```
<http>
<head>
    <style type="text/css">
        .maunen{background-color:red;}
    </style>
</head>
<body>
    <h1 class="maunen">ĐHCNTT</h1>
    <p class="maunen">Khoa HTTT</p>
</body>
</http>
```



# Ví dụ element.class

---

```
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
    p.trai {text-align: left}
    p.phai {text-align: right}
    p.giua {text-align: center}
</style>
</head>
<body>
<p class="trai">Đoạn văn này được canh lề trái.</p>
<p class="phai">Đoạn văn này được canh lề phải.</p>
<p class="giua">Đoạn văn này được canh lề giữa.</p>
</body>
```



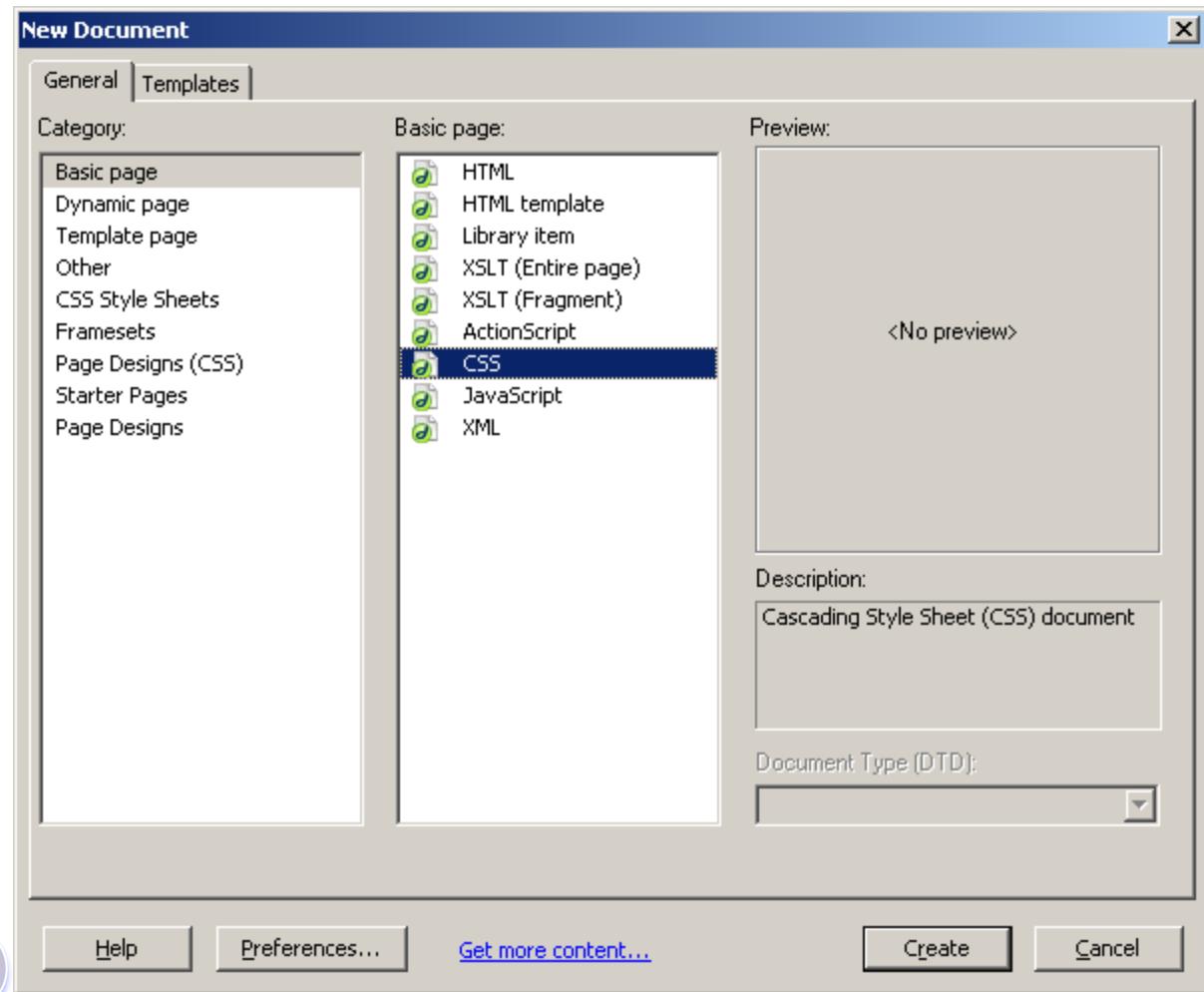
# Tạo css trong Dreamweaver

## ● Bước 1: Tạo mới một file .css

+ file -> new

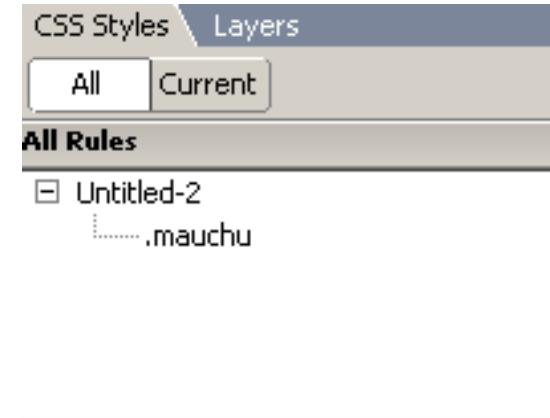
-> Basic page

-> CSS



# Tạo css trong Dreamweaver (tt)

- Bước 2: Định nghĩa style mới
  - Chọn menu window->style – New css rule

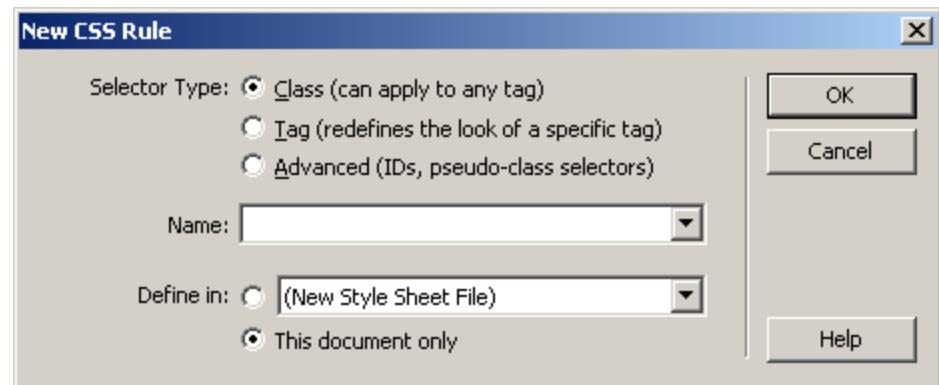


New css  
rule

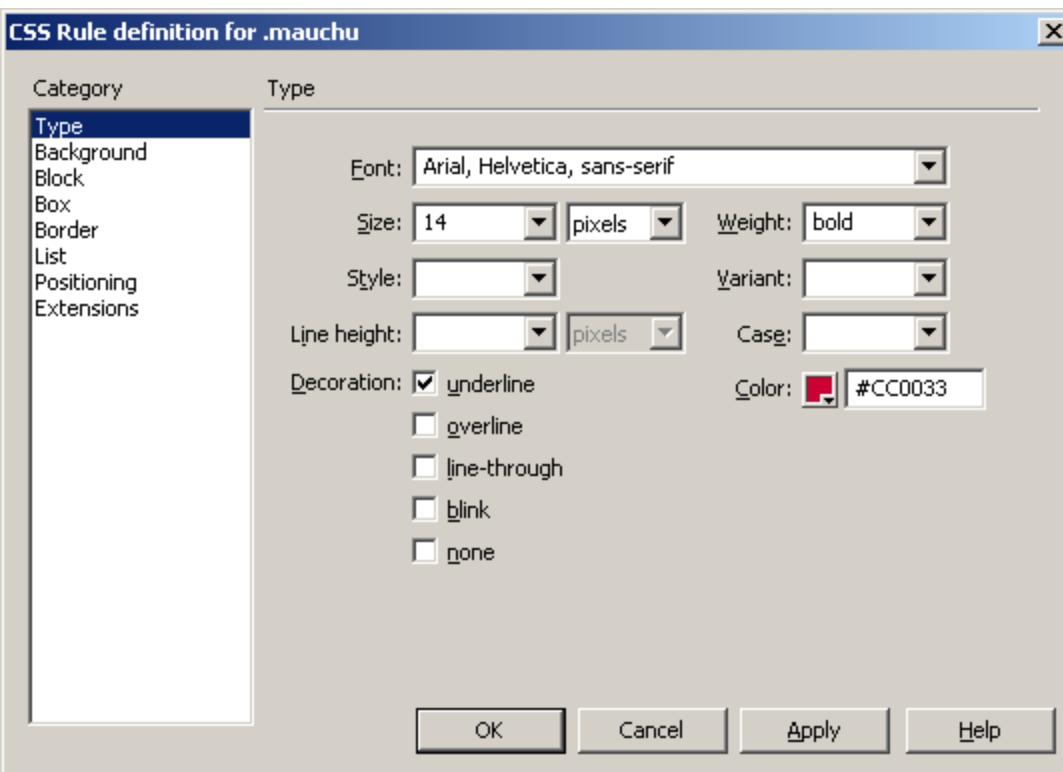


# Tạo css trong Dreamweaver (tt)

- **Class:** Định nghĩa style kiểu class
- **Tag:** Định nghĩa style cho một tag nhất định
- **Advanced:** Định nghĩa style có ID nhất định
- **Name:** Tên của style



# Tạo css trong Dreamweaver (tt)



```
.mauchu{  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 14px;  
    font-weight: bold;  
    color: #CC0033;  
    text-decoration: underline;  
    background-color: #99FF33;  
}
```

**Bước 3:** Nếu cần định nghĩa thêm style ta quay lại bước 2

**Bước 4:** Lưu file css lại



# CSS định màu chữ



- Dùng: color: màu;

**Màu chữ là red**

```
<html>
<style>
h2{
    color: red;
}
</style>
<body>
<h2> MÀU CHỮ LÀ RED </h2>
</body>
</html>
```



# CSS nền

---



- background-color : lightblue;
- background-image : url("paper.gif");
- background-repeat : no-repeat;
  - repeat-x;
  - reppeat-y;
  - repeat-x repeat-y;
- background-position: right top

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```



# Tạo đường viền

---



- ❖ Ta dùng thuộc tính border
  - ❖ border: 1px solid red;
  - ❖ border-style: solid;
  - ❖ border-width: 1px;
  - ❖ border-left: 1px solid red
  - ❖ border-left-style: groove
  - ❖ border-left-width:



# Tạo đường viền

---



- ❖ border: 1px solid red;
- ❖ border-style: solid; //loại đường viền
- ❖ border-width: 1px; //kích thước đường viền
- ❖ border-left: 1px solid red
- ❖ border-left-style: groove
- ❖ border-left-width:
- ❖ border-color: red; //màu đường viền



# Loại đường viền : border-style

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

A ridge border.

An inset border

An outset border

No border.

A hidden border.

A mixed border.



# Tạo đường viền cho table

```
<table>
<tr>
    <th>Số thứ tự</th>
    <th>Tên khách hàng</th>
</tr>
<tr>
    <td>1</td>
    <td>Nguyễn Xuân Tùng</td>
</tr>
</table>
```

```
table{
border-collapse: collapse;
}
table,td,th{
border:1px solid black;
}
```

Số thứ tự	Tên khách hàng
1	Nguyễn Xuân Tùng

# Tạo đường viền với gốc bo

- border-radius: topleft topright bottomright  
bottomleft;

```
<body>
<div> Next</div>
</body>
```



Next

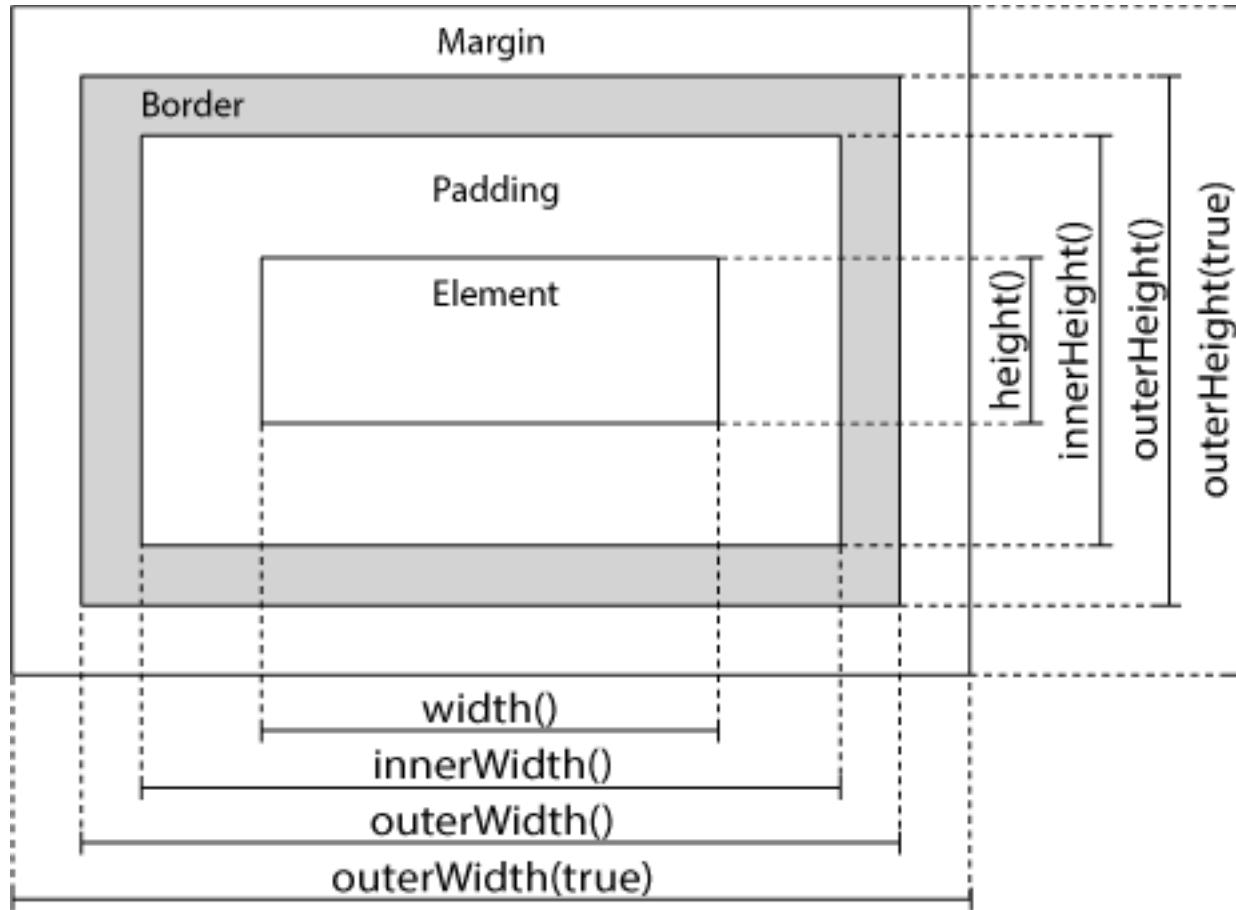
```
div{
background-color: green;
width: 80px;
height: 30px;
border-radius: 5px 5px 5px 5px;
color: white;
}
```

# Bo tròn gốc cho thẻ div

```
<!DOCTYPE html><html>
<head>
<style>
div
{
    border:2px solid #a1a1a1;
    padding:10px 40px; //trên, phải, dưới, trái
    background:#dddddd;//màu nền
    width:300px;//kích thước chiều ngang
    border-radius:25px;//bo tròn 4 gốc
}</style></head>
<body>
<div>The border-radius property allows you to add rounded corners to
elements.</div>
</body></html>
```

[Kết quả](#)

# Dimension



# Dimension



Margin

Border

Padding

Content

# CSS - Dimension

---



- **width:** Định kích thước chiều ngang
- **height:** Định kích thước chiều đứng
- **max-width:** chiều ngang tối đa, nếu kích thước của màn hình đủ. Ngược lại kích thước bằng kích thước tối đa của màn hình



# margin

---



- Định khoảng cách từ viền đến các đối tượng bên ngoài
  - margin: top right bottom left;
  - margin-top: 30px;
  - margin-right: 40px;
  - margin-bottom: 50px;
  - margin-left: 60px;
- margin:inherit ; *thừa hưởng từ đối tượng cha*
- margin: auto; *Nằm giữa đối tượng chứa nó*



# Margin - ví dụ



margin top

```
<head><style type="text/css">
div {
    border: 1px solid red;
}
p {
    background: #00CC00;
    margin-top: 10px;
}
</style>
</head>
<body>
<div>
<p>margin top</p>
</div></body>
```



# padding

---



- Định khoảng cách từ viền đến các đối tượng hiển thị bên trong
  - padding: top right bottom left;
  - padding-top: 30px;
  - padding-right: 40px;
  - padding-bottom: 50px;
  - padding-left: 60px;



# Padding - ví dụ



top padding 50px, right padding 30px, bottom padding 50px, left padding 80px.

```
<style>
div { border: 1px solid black;
background-color: lightblue;
padding: 50px 30px 50px 80px;
}
</style>
</head>
<body>
<div>top padding 50px, right padding 30px, bottom padding 50px, left padding 80px.
</div>
</body>
```



# CSS text

---



- Màu chữ: color: red;
- Canh lề: text-align: center; // left, right, justify
- text-decoration:
  - none; // không gạch chân cho tag <a>
  - Overline; // gạch trên
  - Line-through; // gạch giữa
  - Underline; // chạch chân



# CSS text - ví dụ

---



```
<style>
h1 {
    text-decoration: overline;
}
h2 {
    text-decoration: line-through;
}
H3{
    text-decoration: underline;
}
</style>
</head>
<body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
</body>
```

**This is heading 1**

**This is heading 2**

**This is heading 3**



# CSS text

---



- text-transform:
  - ❖ uppercase; //Hiển thị chữ hoa
  - ❖ lowercase; //hiển thị chữ thường
  - ❖ capitalize; // chữ hoa mỗi từ
- text-indent: 50px; //thụt đầu dòng
- letter-spacing: 3px; // khoảng cách giữa các ký tự
- word-spacing: 10px; // khoảng cách giữa các từ
- line-height: 1.8; //khoảng cách giữa các dòng
- text-shadow: 3px 2px red; // tạo bóng đổ



# Font chữ

---



- font-family: Georgia; // Loại font
- Font-style: kiểu font
  - Normal; //bình thường
  - Italic ; //nghiêng
  - Oblique;// xiên
- Font-size:20px; //kích thước chữ (40px/16=2.5em)
- font-weight: độ lớn chữ
  - normal;//bình thường
  - bold; //chữ mập



# CSS link

---



```
<style>
/* màu link */
a:link { color: blue;}
/* màu sau khi click */
a:visited { color: green;}
/* rê chuột */
a:hover { color: hotpink;}
/* Chọn link */
a:active { color: blue;}
```



# CSS list

---



```
ul.a {  
    list-style-type: circle;  
}  
  
ul.b {  
    list-style-type: square;  
}  
  
ol.c {  
    list-style-type: upper-roman;  
}  
  
ol.d {  
    list-style-type: lower-alpha;  
}
```

Example of unordered lists:

- Coffee
  - Tea
  - Coca Cola
- 
- Coffee
  - Tea
  - Coca Cola

Example of ordered lists:

- I. Coffee
  - II. Tea
  - III. Coca Cola
- 
- a. Coffee
  - b. Tea
  - c. Coca Cola



# CSS List



- List với mục chọn là image

```
ul {  
    list-style-image: url('sqpurple.gif');  
}
```

- Vị trí hiện thị

```
ul {  
    list-style-position: inside;  
}  
  
ul {  
    list-style-position: outside;  
}  
}
```

The following list has list-style-position: inside:

- Coffee
- Tea
- Coca Cola

The following list has list-style-position: outside:

- Coffee
- Tea
- Coca Cola

# CSS display

---



- Dùng để định dạng cách hiển thị

- Hiển thị trên cùng dòng: inline

```
li {  
    display: inline;  
}
```

- Hiển thị theo khối, đứng độc lập một hàng so với thành phần trước và sau đối tượng: block
  - Hiển thị theo khối nhưng trên cùng dòng: inline-block



# CSS position

---



- ✧ Thuộc tính position xác định vị trí tương đối và tuyệt đối cho thành phần, vị trí này phụ thuộc vào các giá trị khai báo của thành phần và thành phần bao ngoài nó.
- ✧ Các giá trị đi kèm:
  - `position: relative`
  - `position: absolute`
  - `position: fixed`
  - `position: static`



# Position kết hợp với các giá trị

---



- **top**: định vị trí bên trên của thành phần.
- **right**: định vị trí bên phải của thành phần.
- **bottom**: định vị trí bên dưới của thành phần.
- **left**: định vị trí bên trái của thành phần.

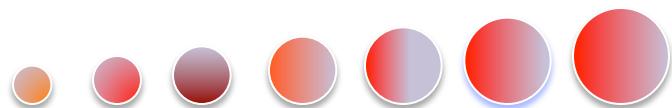


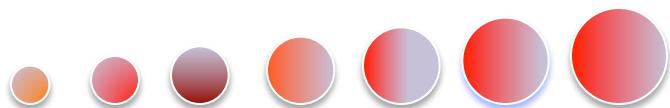
# Ví dụ

---

- File: .html

```
<div>
    <p class="tagA">Thành phần A.</p>
    <p class="tagB">Thành phần B.</p>
</div>
```







# Bài tập



	Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7	CN
Tiết 1	TKWeb 1			LTĐT			TK CSDL
Tiết 2	HDH	TK CSDL	TKWeb 1		TKWeb 1	CSDL	
Tiết 3	CSDL	LTĐT	HDH	TK CSDL		LTĐT	TKWeb 1
Tiết 4			CSDL		HDH		

