

Lập trình Web Front-end 1

| Front-end Web Development 1 |

Nguyễn Huy Hoàng – Bùi Thị Phương Thảo

[02 . 2019]

Dùng kèm Lập trình Web Front-end 1, Khoa Công nghệ thông tin, Cao đẳng Công nghệ Thủ Đức

Bài 2. TỔNG QUAN VỀ HTML và CSS



2.5. GIỚI THIỆU CSS

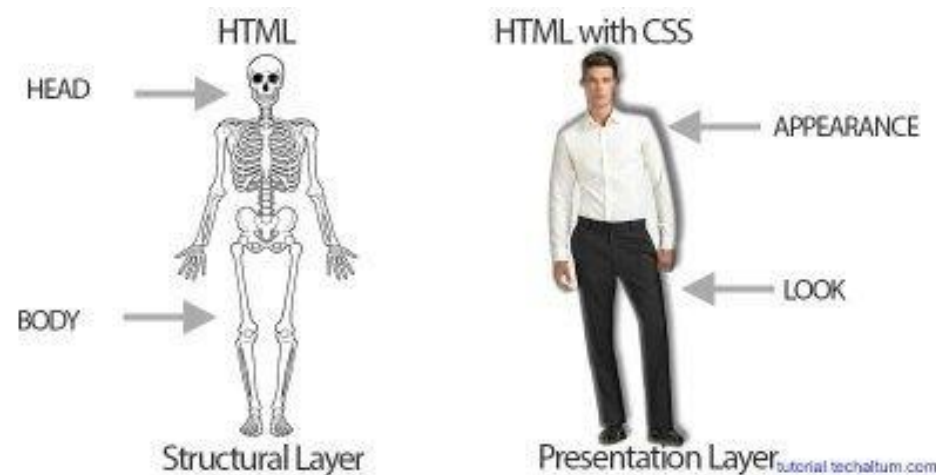


FACULTY OF
INFORMATION TECHNOLOGY
THU DUC COLLEGE OF TECHNOLOGY



CSS là gì?

- CSS là chữ viết tắt của **Cascading Style Sheets**, là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XML, XHTML, SVG hay UML.



Tại sao phải sử dụng CSS

- Khi chưa có CSS, tài liệu HTML sẽ bao gồm nội dung và các thành phần định kiểu cho các nội dung đó □ làm rối mã HTML □ sử dụng CSS để tách phần định kiểu và nội dung thành hai phần rời nhau.

```
<p align="right"><b><i>Lorem ipsum dolor sit amet</i></b></p>
```



```
<p>Lorem ipsum dolor sit amet</p>
```

&

```
p{  
  font-weight: bold;  
  font-style: italic;  
  text-align: right;  
}
```

Tại sao phải sử dụng CSS

- CSS tạo ra các kiểu dáng có thể áp dụng cho nhiều trang web □ tránh phải lặp lại việc định dạng cho các trang web giống nhau.
- CSS giúp kiểm soát bố cục của nhiều trang web cùng một lúc và bố cục của trang web trên nhiều thiết bị với nhiều kích thước khác nhau.

Cú pháp

```
selector{  
    property1: value;  
    property2: value;  
}
```

selector trỏ đến phần tử HTML cần định kiểu.
selector có thể là tên thẻ, tên class, tên id.

Ví dụ:

```
p {  
    color: red;  
    text-align: center;  
}
```



2.6. PHÂN LOẠI CSS



External CSS

- External CSS là khi chúng ta sử dụng một tập tin css độc lập và mỗi trang web cần sử dụng tập tin css này thì phải tham chiếu đến tập tin đó bằng cách sử dụng thẻ <link>.

Ví dụ: trong thẻ <head> của tài liệu HTML chúng ta chèn liên kết sau:

```
<link rel="stylesheet" type="text/css" href="đường dẫn đến tập tin  
css">
```

Internal CSS

- Internal CSS được sử dụng để định kiểu riêng cho từng trang web.
- Internal CSS được định nghĩa bằng cách đặt toàn bộ mã nguồn CSS trong thẻ <style> đặt trong thẻ <head> của tài liệu HTML

Ví dụ:

```
<head>
  <meta charset="utf-8">
  <title>My first website</title>
  <style>
    body {
      background-color: linen;
    }
    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
</body>
```

Inline CSS

- Inline CSS được sử dụng để định kiểu riêng cho một phần tử HTML.
- Inline CSS được định nghĩa bằng cách thêm thuộc tính *style* vào phần tử cần định kiểu.

Ví dụ:

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

- Mức độ ưu tiên của CSS sẽ giảm dần theo thứ tự sau: Inline CSS - Internal CSS - External CSS - Style của trình duyệt.



2.7. CSS SELECTORS



CSS selectors

- Selectors được dùng để chọn các phần tử trong tài liệu HTML mà chúng ta muốn định kiểu CSS.
- Danh sách selector CSS cơ bản:

| Selector | Ví dụ | Mô tả |
|-----------|-------|--|
| * | * | Chọn tất cả các phần tử html. |
| tag | p | Chọn tất cả các thẻ cùng loại. |
| tag1 tag2 | div p | Chọn tất cả phần tử <tag2> bên trong phần tử <tag1>. |

CSS selectors

| Selector | Ví dụ | Mô tả |
|---------------------|------------|---|
| tag1,tag2 | div,p | Chọn tất cả phần tử <tag1> và <tag2>. |
| tag1+tag2 | div+p | Chọn tất cả phần tử <tag2> được đặt kế và sau phần tử <tag1>. |
| tag1>tag2 | div>p | Chọn tất cả phần tử <tag2> có phần tử cha là <tag1>. |
| .class | .container | Chọn tất cả các phần tử có cùng tên class. |
| #id | #wrapper | Chọn tất cả các phần tử có cùng tên id. |

CSS selectors

| Selector | Ví dụ | Mô tả |
|----------------------|----------------|--|
| :active | a:active | Chọn các liên kết được kích hoạt. |
| :after | p:after | Thêm nội dung ngay phía sau phần tử. |
| :before | p:before | Thêm nội dung ngay phía trước phần tử. |
| :first-child | p:first-child | Chọn phần tử đầu tiên của phần tử cha chứa nó. |
| :first-letter | p:first-letter | Chọn ký tự đầu tiên của một phần tử. |

CSS selectors

| Selector | Ví dụ | Mô tả |
|---------------------|--------------|---|
| :first-line | p:first-line | Chọn dòng đầu tiên của một phần tử. |
| :hover | a:hover | Chọn các liên kết được hover (di chuyển chuột lên phần tử). |
| :link | a:link | Định dạng cho tất cả liên kết khi chưa được click. |
| :visited | a:visited | Định dạng cho các phần tử đã được click. |
| [Thuộc tính] | [target] | Chọn tất cả các phần tử có cùng thuộc tính (attribute). |



2.8. CÁC CSS CƠ BẢN



CSS Color

- Màu sắc trong CSS thường được xác định bởi **tên màu** hoặc giá trị **RGB** hoặc giá trị **Hexadecimal**.
- **Tên màu**: không phân biệt chữ hoa, chữ thường. HTML và CSS hỗ trợ 140 tên màu tiêu chuẩn. Ví dụ: red, green, blue, yellow, ...
- **Giá trị RGB**: được chỉ định bằng công thức: `rgb(red, green, blue)`. Mỗi tham số red, green, blue xác định cường độ của màu đó và có giá trị từ 0 đến 255. Ví dụ: `rgb(100, 50, 10)`.

CSS Color

- **Giá trị Hexadecimal**: được chỉ định bằng cách sử dụng giá trị màu RGB ở dạng thập lục phân #RRGGBB, trong đó RR (màu đỏ), GG (màu xanh lá cây), BB (màu xanh dương) là các giá trị thập lục phân từ 00 đến FF. Ví dụ: #FFA500, #808080.

CSS Background

- Thuộc tính background được sử dụng để định dạng màu nền, ảnh nền cho phần tử.
- **Cấu trúc:** `selector{
background-....: value;
}`
- **Các thuộc tính:**
 - ◆ `background-color: color` - Định màu nền cho phần tử.
Ví dụ: `background-color : #ff0000;`

CSS Background

◆ `background-image: url(đường dẫn hình)`

- Định hình nền cho phần tử.

Ví dụ: `background-image: url(ico_arrow.gif);`

◆ `background-repeat: repeat-x | repeat-y | repeat | no-repeat;`

- Xác định hình ảnh nền được lặp lại như thế nào.

Ví dụ: `background-repeat: repeat-x;`

CSS Background

◆ `background-position: left | right | top | bottom | px | %;`

□ Xác định vị trí của ảnh nền cho phần tử.

Ví dụ: `background-position: left top;`

◆ `background-attachment: fixed | scroll;`

□ Xác định nền được cố định hoặc cuộn so với trang.

Ví dụ: `background-attachment: fixed;`

CSS Background

- Có thể viết rút gọn các thuộc tính trên bằng cách sử dụng thuộc tính background.

background: color image position repeat attachment;

Ví dụ: background: url(ico_arrow.gif) repeat-x left top;

CSS Border

- Thuộc tính border được dùng để định dạng kiểu, màu sắc và độ dày của đường viền cho phần tử.
- **Cấu trúc:** `selector{
border-....: value;
}`
- **Các thuộc tính:**
 - ◆ `border-color: color;`
 - Xác định màu sắc của đường viền. Mặc định là màu nội dung

CSS Border

◆ `border-style`: `dotted` | `dashed` | `solid` | `double` | `groove` | `ridge` | `inset` | `outset`;

□ Xác định kiểu của đường viền.

◆ `border-width`: `px` | `thin` | `medium` | `thick`;

□ Xác định độ dày của đường viền.

- Có thể viết rút gọn các thuộc tính trên bằng cách sử dụng thuộc tính `border`.

`border`: `width style color`;

CSS Border

Xác lập `border-color`,
`border-style`, `border-width`
cho 4 cạnh có giá trị khác
nhau như thế nào?

```
top: green double 4px  
right: red groove 4px  
bottom: blue solid 3px  
left: red double 4px
```



https://www.w3schools.com/css/css_border_color.asp

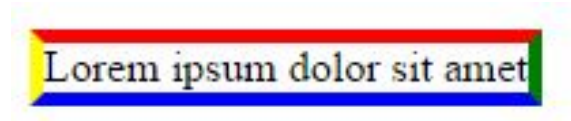
CSS Border

- Có thể định dạng đường viền riêng cho từng cạnh của phần tử bằng cách sử dụng các thuộc tính sau:
 - border-top [right | bottom | left]-color
 - border-top [right | bottom | left]-style
 - border-top [right | bottom | left]-width
- Khi xác lập **border-color**, **border-style**, **border-width** cho 4 cạnh có **giá trị khác nhau**, chúng ta có thể viết dưới dạng rút gọn như sau:

CSS Border

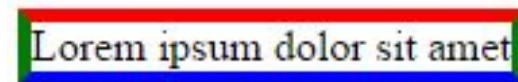
- Khi 4 cạnh có giá trị khác nhau □ giá trị của các thuộc tính trên sẽ là: **top right bottom left**;

Ví dụ: border-color: red green blue yellow;



- Khi right và left có giá trị giống nhau □ giá trị của các thuộc tính trên sẽ là: **top {right left} bottom**;

Ví dụ: border-color: red green blue;



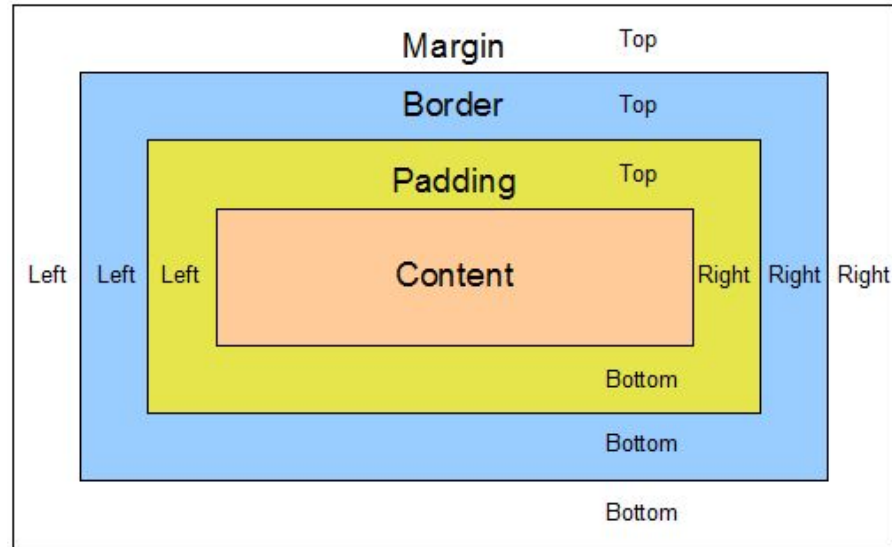
CSS Border

- Khi right và left giống nhau, top và bottom giống nhau □ giá trị của các thuộc tính trên sẽ là: {**top bottom**} {**right left**};

Ví dụ: border-color: red green; 

|| CSS Margin

- Thuộc tính **margin** được sử dụng để xác định không gian xung quanh phần tử hay nói cách khác là xác định **kích thước của khoảng trống bên ngoài đường viền**.



|| CSS Margin

- **Cấu trúc:**

```
selector{  
    margin-....: value;  
}
```

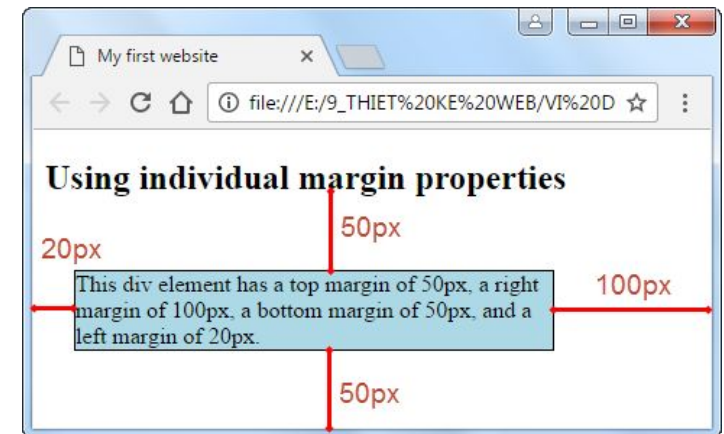
- **Các thuộc tính:**

- ◆ `margin-top` [`right` | `bottom` | `left`]: `value`[px | pt | % | cm] ☐ Xác định lề trên [phải | dưới | trái] cho phần tử.

CSS Margin

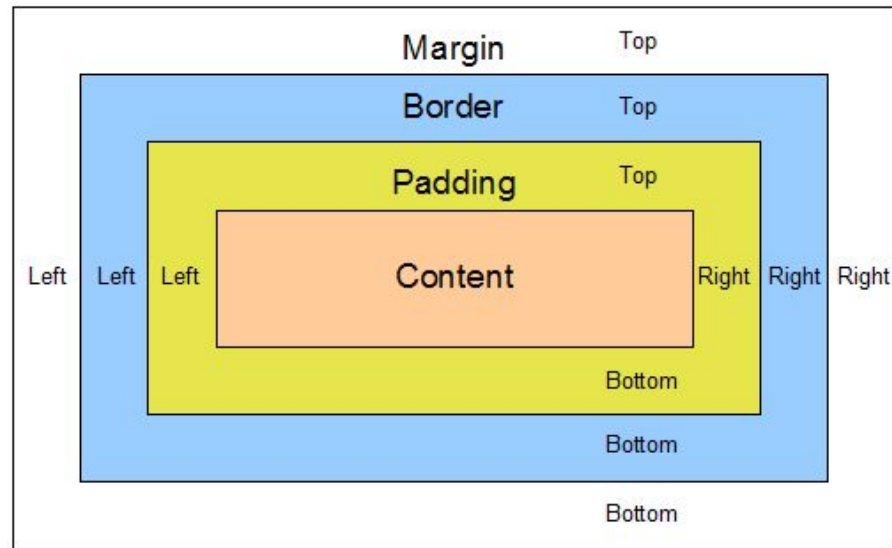
- Có thể định lề cho các cạnh với các giá trị khác nhau bằng cách sử dụng thuộc tính `margin` (tương tự cách viết rút gọn của `border-width`).

Ví dụ: `margin: 50px 100px 50px 20px;`



CSS Padding

- Thuộc tính **padding** được sử dụng để xác định không gian giữa nội dung và đường viền của phần tử.



|| CSS Padding

- Cấu trúc:

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

- Các thuộc tính:

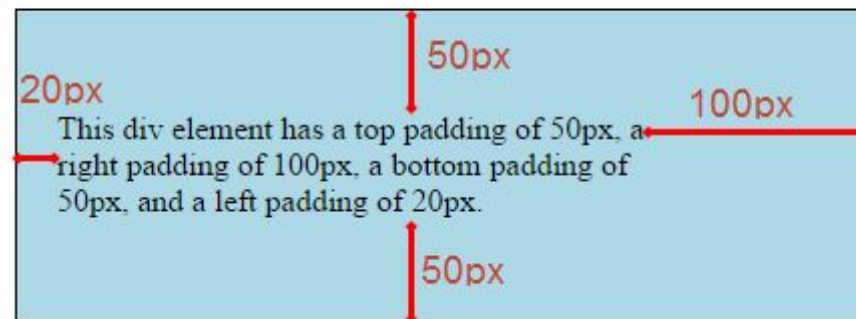
- ◆ `padding-top` [`right` | `bottom` | `left`]: `value`[px | pt | % | cm] □ Xác định vùng đệm bên trên [phải|dưới|trái] cho phần tử.

|| CSS Padding

- Có thể định vùng đệm cho các cạnh với các giá trị khác nhau bằng cách sử dụng thuộc tính **padding** (tương tự cách viết rút gọn của **border-width**).

Ví dụ: padding: 50px 100px 50px 20px;

Using individual padding properties



|| CSS Width/ Height

- Thuộc tính width và height được sử dụng để thiết lập chiều rộng và chiều cao của một phần tử.

- **Cấu trúc:**

```
selector{  
    width: value;  
    height: value;
```

```
}
```

- **Ví dụ:**

```
div {  
    height: 100px;  
    width: 400px;  
    background-color: powderblue;  
    padding: 10px;  
}
```

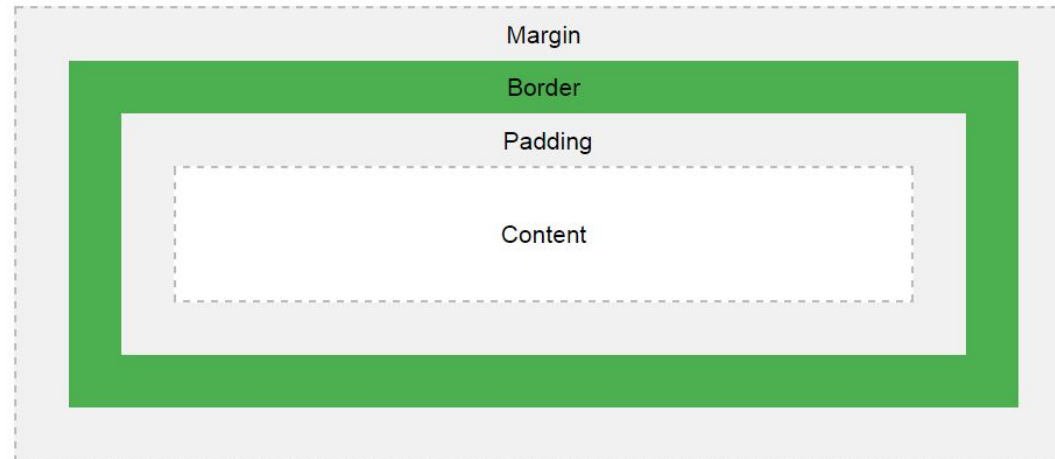


Using individual padding properties

This div element has a top padding of 50px, a right padding of 100px, a bottom padding of 50px, and a left padding of 20px.

CSS Box Model

- Tất cả các phần tử HTML có thể được coi là các hộp. Mô hình hộp CSS về cơ bản là một hộp bao quanh mọi phần tử HTML. Nó bao gồm: lề, đường viền, vùng đệm và nội dung thực tế.



CSS Box Model

- **Kích thước của phần tử:** Khi đặt thuộc tính width và height của phần tử, chúng ta chỉ cần đặt width và height của phần nội dung (content). Nhưng khi tính toán kích thước đầy đủ của phần tử, chúng ta phải cộng thêm phần padding, border và margin vào:
 - Tổng chiều rộng = width + left padding + right padding + left border + right border + left margin + right margin.
 - Tổng chiều cao = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS Box Model

- Ví dụ:

```
div {  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;  
}
```

□ Tổng chiều rộng của phần tử <div> trên là: 320px (width) + 20px (left + right padding) + 10px (left + right border) + 0px (left + right margin) = **350px**

CSS Text

- CSS text được sử dụng để định dạng màu, canh lề, trang trí, viết hoa, ... cho văn bản.
- Các thuộc tính:
 - ◆ `color: color;`
 - Xác định màu cho văn bản.

Ví dụ: `h1{color: #44aa00;}`

Lorem ipsum

|| CSS Text

◆ `text-align: left | right | center | justify;`

□ Canh lề cho văn bản.

Ví dụ: `h1{text-align : center;}`

Lorem ipsum

Lorem ipsum dolor sit amet

◆ `text-decoration: underline | overline | line-through | none;`

□ Trang trí cho văn bản.

Ví dụ: `h1{text-decoration : underline;}`

Lorem ipsum

CSS Text

◆ `text-transform` : `capitalize` | `uppercase` | `lowercase`;

□ Xác định việc viết hoa của văn bản.

Ví dụ: `h1{text-transform : uppercase;}`

LOREM IPSUM

◆ `text-indent`: `value`;

□ Xác định khoảng thụt lề của dòng đầu tiên trong đoạn văn bản.

Ví dụ: `p{text-indent : 30px;}`

Lorem ipsum dolor sit amet Lorem ipsum
dolor sit amet Lorem ipsum dolor sit amet Lorem
ipsum dolor sit amet

CSS Text

◆ `letter-spacing` : **value**;

□ Xác định khoảng cách giữa các kí tự.

Ví dụ: `h1{letter-spacing : 0.5em;}`

L o r e m i p s u m

◆ `word-spacing`: **value**;

□ Xác định khoảng cách giữa các từ trong khối văn bản.

Ví dụ: `p{word-spacing : 20px;}`

Lorem ipsum dolor sit amet Lorem
ipsum dolor sit amet Lorem ipsum dolor
sit amet Lorem ipsum dolor sit amet

CSS Text

◆ `line-height: value;`

□ Xác định chiều cao của 1 dòng văn bản.

Ví dụ: `p{line-height : 2;}`

Lorem ipsum dolor sit amet Lorem ipsum dolor
sit amet Lorem ipsum dolor sit amet Lorem
ipsum dolor sit amet

CSS Font

- CSS font được sử dụng để xác định họ font, độ đậm, kích thước và phong cách của văn bản.
- Trong CSS có 2 họ font:
 - Generic family: một nhóm các font có cùng họ (serif, sans-serif, monospace...).
 - Font family: tên cụ thể của một font (Times New Roman, Arial, ...).

CSS Font

- Cấu trúc: `selector{
 font-...: value;
}`

- Các thuộc tính:

- ◆ `font-family: font;`

- Xác định kiểu chữ cho văn bản.

Ví dụ: `h1{font-family: Verdana, Geneva, sans-serif;}`

Lorem ipsum

CSS Font

◆ `font-style: italic | oblique;`

□ Xác định phong cách cho văn bản.

Ví dụ: `h1{font-style : italic;}`

Lorem ipsum

◆ `font-size: size-name | value;`

□ Xác định kích thước chữ cho văn bản.

Ví dụ: `p{font-size : 2em;}`

Lorem ipsum dolor sit amet Lorem
ipsum dolor sit amet Lorem ipsum
dolor sit amet Lorem ipsum dolor sit
amet

CSS Font

◆ `font-weight: number | bold | bolder | lighter;`

□ Xác định độ đậm cho văn bản.

Ví dụ: `p{font-weight : bold;}`

Lorem ipsum dolor sit amet Lorem ipsum dolor sit
amet Lorem ipsum dolor sit amet Lorem ipsum dolor
sit amet

◆ `font-variant: normal | small-caps;`

□ Xác định việc chuyển đổi văn bản sang chữ in hoa nhỏ.

Ví dụ: `p{font-variant : small-caps;}`

LOREM IPSUM DOLOR SIT AMET LOREM IPSUM DOLOR SIT AMET
LOREM IPSUM DOLOR SIT AMET LOREM IPSUM DOLOR SIT AMET

CSS Font

- Khi sử dụng nhiều thuộc tính định dạng font cùng một lúc, chúng ta có thể sử dụng cách viết rút gọn sau:

`font` : `font-style` | `font-variant` | `font-weight` | `font-size` | `font-family`;

CSS List

- CSS list cho phép thiết lập các điểm đánh dấu cho danh sách.
- **Cấu trúc:** `selector{
 list-style-....: value;
}`
- **Các thuộc tính:**
 - ◆ `list-style-type: circle | disc | square | upper-roman | upper-latin | ...;`
 - Xác định kiểu đánh dấu cho danh sách.

CSS List

◆ `list-style-position: inside | outside;`

□ Xác định vị trí của điểm đánh dấu so với nội dung.

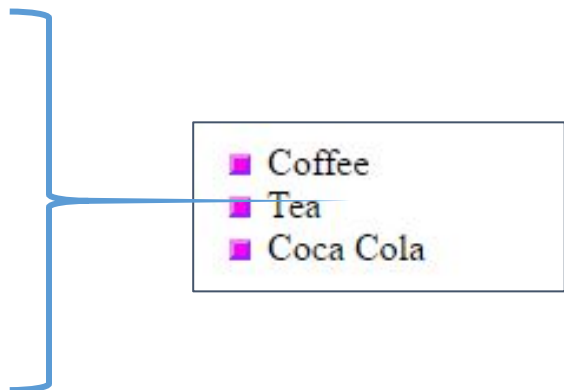
◆ `list-style-image: url(đường dẫn đến ảnh);`

□ Xác định một ảnh làm điểm đánh dấu cho danh sách.

• Ví dụ:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```
ul {
  list-style: square url("sqpurple.gif");
}
```



■ Coffee
■ Tea
■ Coca Cola

CSS Table

- **Collapse border**: gộp đường viền của table thành đường viền đơn

Ví dụ:

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



| Firstname | Lastname |
|-----------|----------|
| Peter | Griffin |
| Lois | Griffin |

- **Hoverable** dùng để dùng di chuột lên một hàng bất kỳ của table thì hàng đó sẽ thay đổi trạng thái ☐ sử dụng pseudo-class cho các hàng trong table.

CSS Table

Ví dụ:

```
table {  
  border-collapse: collapse;  
  width: 100%;  
}  
th, td {  
  padding: 8px;  
  text-align: left;  
  border-bottom: 1px solid #ddd;  
}  
tr:hover{background-color:#f5f5f5}
```



| First Name | Last Name | Points |
|------------|-----------|--------|
| Peter | Griffin | \$100 |
| Lois | Griffin | \$150 |

- **Striped table** : định dạng các hàng có màu luân phiên □ sử dụng selector **`nth-child()`** và thêm màu nền cho tất cả các hàng chẵn (hoặc lẻ).

CSS Table

Ví dụ:

```
table {  
  border-collapse: collapse;  
  width: 100%;  
}  
th, td {  
  text-align: left;  
  padding: 8px;  
}  
tr:nth-child(even){background-color: #f2f2f2}
```



| First Name | Last Name | Points |
|------------|-----------|--------|
| Peter | Griffin | \$100 |
| Lois | Griffin | \$150 |
| Joe | Swanson | \$300 |



2.8. CSS3 CƠ BẢN

Giới thiệu CSS3

- CSS3 là bản nâng cấp của CSS và là tiêu chuẩn mới của CSS. CSS3 hỗ trợ cho website một cách toàn diện nhất nhờ **thành phần mới** thay vì phải dùng tới các thành phần bổ sung bên ngoài như JavaScript, JQuery, Flash.
- CSS3 vẫn còn đang được phát triển và được hầu hết các trình duyệt web hỗ trợ.

Giới thiệu CSS3

- Một số module quan trọng trong CSS3:
 - Selectors
 - Box Model
 - Backgrounds and Borders
 - Text Effects
 - 2D/3D Transformations
 - Animations
 - Multiple Column Layout
 - User Interface



CSS3 selector

| Selector | Ví dụ | Mô tả |
|----------------------------------|--------------------------------|--|
| <code>[attribute^=value]</code> | <code>img[src^="bnr_"]</code> | Chọn tất cả phần tử với thuộc tính có giá trị bằng "value". |
| <code>[attribute\$=value]</code> | <code>img[src\$=".gif"]</code> | Chọn tất cả phần tử với thuộc tính có giá trị kết thúc bằng "value". |
| <code>[attribute*=value]</code> | <code>img[src*=""]</code> | Chọn tất cả phần tử với thuộc tính có giá trị đặc biệt bằng "value". |
| <code>:first-of-type</code> | <code>p:first-of-type</code> | Chọn phần tử con đầu tiên hoặc duy nhất trong các phần tử cha. |
| <code>:last-of-type</code> | <code>p:last-of-type</code> | Chọn phần tử con cuối cùng hoặc duy nhất trong các phần tử cha. |



CSS3 selector

| Selector | Ví dụ | Mô tả |
|-----------------------------|-----------------------|---|
| :nth-child(n) | p:nth-child(3) | Chọn phần tử thứ "n" trong phần tử cha. |
| :nth-last-child(n) | p:nth-last-child(3) | Chọn phần tử thứ "n" tính từ phần tử cuối trong phần tử cha. |
| :nth-last-of-type(n) | p:nth-last-of-type(3) | Chọn phần tử thứ "n" từ phần tử cuối trong phần tử cha. |
| :last-child | p:last-child | Chọn phần tử cuối cùng trong phần tử cha. |
| :root | :root | Chọn phần tử gốc của văn bản. |
| :enabled | input:enabled | Chọn phần tử được kích hoạt, thường sử dụng cho các phần tử của form. |



CSS3 selector

| Selector | Ví dụ | Mô tả |
|-----------------------|----------------|---|
| :disabled | input:disabled | Chọn phần tử bị vô hiệu hóa, thường sử dụng cho các phần tử của form. |
| :checked | input:checked | Chọn phần tử được check, thường sử dụng cho các phần tử của form. |
| :not(selector) | :not(p) | Chọn tất cả ngoại trừ selector trong ngoặc. |
| ::selection | ::selection | Chọn phần tử được người dùng chọn. |

CSS3 Border

- `border-radius: value;`
 - Được sử dụng để bo tròn góc các phần tử trên trang web.
 - Có thể chỉ định độ bo góc riêng cho từng góc bằng cách sử dụng các thuộc tính *border-top-left-radius*, *border-top-right-radius*, *border-bottom-right-radius* và *border-bottom-left-radius* hoặc xác lập giá trị cho *border-radius* bằng 2, 3 hoặc 4 giá trị.

|| CSS3 Border

- border-radius: value;

- Ví dụ:

```
#rcorners1 {  
    border-radius: 25px;
```



```
#rcorners1 {  
    border-radius: 15px 50px 30px 5px;
```



```
#rcorners1 {  
    border-radius: 15px 50px 30px;
```



```
#rcorners1 {  
    border-radius: 15px 50px;
```



|| CSS3 Border

- `border-image: source slice width outset repeat;`
 - Được sử dụng để thiết lập một ảnh làm đường viền cho phần tử.
 - source* \equiv url: Xác định đường dẫn đến ảnh dùng làm đường viền.
 - slice* \equiv number | % | fill: Chỉ định lát cắt ảnh như thế nào.
 - width* \equiv number | % | px: Xác định độ dày của ảnh viền.

CSS3 Border

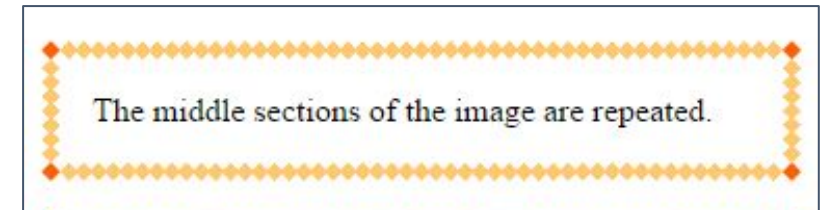
- `border-image`: *source slice width outset repeat*;

outset \equiv normal | small-caps: Xác định giá trị image vượt ra ngoài vùng giới hạn của border.

repeat \equiv stretch | stretch | round | space : Xác định ảnh viền được lặp lại, làm tròn hay kéo dài.

Ví dụ:

```
#borderimg1 {  
  border: 10px solid transparent;  
  padding: 15px;  
  -webkit-border-image: url(border.png) 30 round;  
  -o-border-image: url(border.png) 30 round;  
  border-image: url(border.png) 30 round;  
}
```



CSS3 Background

- `background-size`: `auto` | `length` | `cover` | `contain`;

□ Xác định kích thước của ảnh nền.

length \equiv `value(px|%|...)`: Xác định chiều rộng, còn chiều cao sẽ tự động chỉnh theo sao cho đúng tỷ lệ.

cover: Tự chia tỷ lệ tới kích thước lớn nhất của chiều rộng hoặc chiều cao để phù hợp với vùng nội dung.

contain: Tự chia tỷ lệ tới kích thước nhỏ nhất của chiều rộng hoặc chiều cao để phù hợp với vùng nội dung.

|| CSS3 Background

- `background-size: auto | length | cover | contain;`

Ví dụ:

```
p {  
    background: url("img_flwr.gif");  
    background-size: 100px 70px;  
    background-repeat: no-repeat;  
    padding-top: 20px;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CSS3 Background

- **background-origin**: padding-box | border-box | content-box;
 - Xác định vị trí tương đối của background so với vùng nội dung.
- padding-box**: Hình nền bắt đầu từ cạnh trên bên trái của vùng padding.
- border-box**: Hình nền bắt đầu từ cạnh trên bên trái của vùng border.

|| CSS3 Background

- `background-origin: padding-box | border-box | content-box;`
`content-box`: Hình nền bắt đầu từ cạnh trên bên trái của vùng nội dung.

Ví dụ:

```
#example1 {  
    border: 10px solid black;  
    padding: 35px;  
    background: url(img_flwr.gif);  
    background-repeat: no-repeat;  
    background-origin: content-box;  
}
```



CSS3 Background

- `background-clip`: `padding-box` | `border-box` | `content-box`;

☐ Xác định vùng hiển thị màu của background.

`padding-box`: Background được giới hạn trong vùng padding.

`border-box`: Background được giới hạn trong vùng border.

`content-box`: Background được giới hạn trong vùng nội dung.

|| CSS3 Background

- `background-clip: padding-box | border-box | content-box;`

Ví dụ:

```
#example1 {  
    border: 5px dotted black;  
    padding: 20px;  
    background: yellow;  
    background-clip: content-box;  
}
```



|| CSS3 Color

- **RGBA colors** \equiv **rgba** (red, green, blue, alpha)

alpha: xác định độ mờ đục, là một số giữa 0,0 (hoàn toàn trong suốt) và 1,0 (đục hoàn toàn).

Ví dụ:

```
#p1{background-color:rgba(255,0,0,1);}
#p2{background-color:rgba(255,0,0,0.7);}
#p3{background-color:rgba(255,0,0,0.3);}
```



|| CSS3 Color

- **HSL colors** \equiv **hsl** (**hue**, **saturation**, **lightness**)

hue: Là chỉ số độ trên bánh xe màu (từ 0 đến 360): 0 (hoặc 360) là màu đỏ, 120 là màu xanh, 240 là màu xanh lam.

saturation: là độ bão hòa, là giá trị phần trăm: 100% là màu đầy đủ.

lightness: là độ sáng, là giá trị phần trăm; 0% là màu tối (đen) và 100% là màu trắng.

CSS3 Color

- **HSLA colors** \equiv **hsla** (**hue**, **saturation**, **lightness**, **alpha**)
- **Opacity**
 - **opacity**: **[0.0 , 1.0]**; □ thiết lập độ mờ đục cho toàn bộ phần tử (cả màu nền và văn bản sẽ bị mờ đục hoặc trong suốt).
- **Gradient** □ Cho phép hiển thị sự chuyển màu giữa hai hoặc nhiều màu sắc được chỉ định
 - **linear-gradient**(**direction**, **color-stop1**, **color-stop2**, ...);
 - **linear-gradient**(**angel**, **color-stop1**, **color-stop2**, ...);

|| CSS3 Color

- Gradient

- radial-gradient(*shape size at position, start-color, ..., last-color*);
- repeating-linear-gradient(*angel, color-stop1, color-stop2, ...*);
- repeating-radial-gradient(*shape size at position, start-color, ..., last-color*);

|| CSS3 Shadow

- `text-shadow: horizontal vertical blur color;`

horizontal, vertical : Xác định vị trí bóng đổ theo chiều ngang và chiều dọc. Có thể mang giá trị âm.

blur: Xác định độ mờ biên của bóng đổ.

color: Xác định màu cho bóng đổ.

Ví dụ:

```
h1 {  
  color: white;  
  text-shadow: 1px 1px 2px black,  
              0 0 25px blue,  
              0 0 5px darkblue;  
}
```



Text-shadow effect!

|| CSS3 Shadow

- `box-shadow`: *horizontal vertical blur color*;

horizontal, vertical : Xác định vị trí bóng đổ theo chiều ngang và chiều dọc. Có thể mang giá trị âm.

blur: Xác định độ mờ biên của bóng đổ.

color: Xác định màu cho bóng đổ.

Ví dụ:

```
#boxshadow {  
  -moz-box-shadow: 1px 2px 4px rgba(0, 0, 0, 0.5);  
  -webkit-box-shadow: 1px 2px 4px rgba(0, 0, 0, .5);  
  box-shadow: 1px 2px 4px rgba(0, 0, 0, .5);  
  padding: 10px;  
  background: white;  
}
```



CSS3 Text

- `text-overflow: clip | ellipsis;` □ xác định hình thức hiển thị của phần nội dung bị tràn.

clip: Cắt bỏ phần nội dung bị tràn.

ellipsis: Sử dụng dấu ... để thay thế phần nội dung bị tràn.

Ví dụ:

```
p.test1 {  
  white-space: nowrap;  
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: clip;  
}  
  
p.test2 {  
  white-space: nowrap;  
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```



`text-overflow: clip;`

This is some long text that will

`text-overflow: ellipsis;`

This is some long text that w...

|| CSS3 Text

- `word-wrap: normal | break-word;` □ Cho phép các từ quá dài có thể bị ngắt và đẩy xuống dòng tiếp theo.

Ví dụ:

```
p.test {  
  width: 11em;  
  border: 1px solid #000000;  
  word-wrap: break-word;  
}
```



This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

CSS3 Text

- `word-break: break-all / keep-all;` ☐ Làm cho những chữ trong một từ không còn là một thể thống nhất.

Ví dụ:

```
p.test1 {  
  width: 140px;  
  border: 1px solid #000000;  
  word-break: keep-all;  
}  
  
p.test2 {  
  width: 140px;  
  border: 1px solid #000000;  
  word-break: break-all;  
}
```



This paragraph
contains some text.
This line will-break-
at-hyphens.

This paragraph contain
s some text. The lin
es will break at any c
haracter.



TỔNG KẾT KEY POINTS



FACULTY OF
INFORMATION TECHNOLOGY
THU DUC COLLEGE OF TECHNOLOGY



|| Những điểm cần nhớ

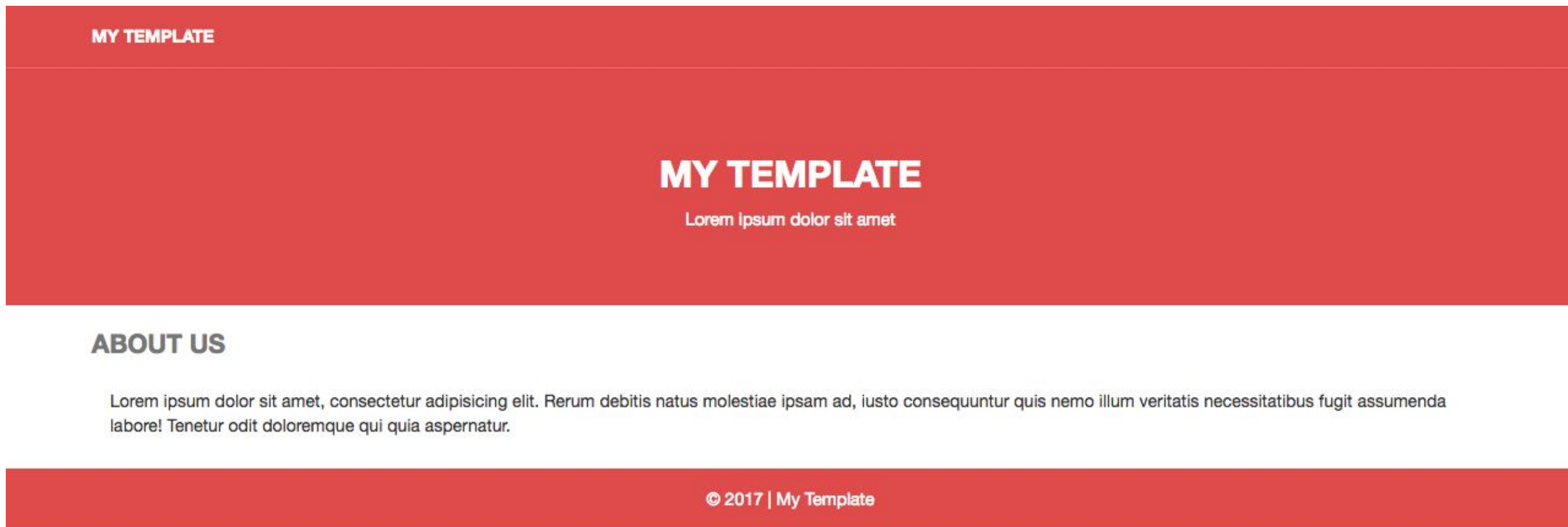
- Cú pháp CSS.
- Phân loại CSS.
- CSS selector cơ bản.
- Các CSS cơ bản
- CSS3 selector cơ bản.
- Các CSS3 cơ bản.



BÀI TẬP

Bài tập

Thực hiện giao diện như hình:



Thanks for your attention!



FACULTY OF
INFORMATION TECHNOLOGY
THU DUC COLLEGE OF TECHNOLOGY

Phone: (+848) 22 158 642

Email: fit@tdc.edu.vn

Website: fit.tdc.edu.vn

Facebook: facebook.com/tdc.fit

Youtube: youtube.com/fit-tdc