



Chương 2 - Tạo dựng trang Web

Người trình bày: NGUYỄN ĐỨC TUẤN

Khoa Công nghệ Thông tin

Hà nội, 30/08/2022



1.1. Tạo cấu trúc trang web bằng HTML

1.2. Định dạng, trình bày trang web bằng CSS

1.3. Kỹ thuật dàn trang

1.1. Tạo cấu trúc trang web bằng HTML

■ Giới thiệu

- Để tạo ra một trang web tĩnh, cần sử dụng HTML (*HyperText Markup Language*) để bố cục cho trang
- Sử dụng CSS để trang trí cho trang web
- HTML là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web.
- Phiên bản phổ biến hiện nay là HTML 4.01 (1999).
- Hiện nay phiên bản HTML 5.0 đã được phát triển và bắt đầu được sử dụng

1.1. Tạo cấu trúc trang web bằng HTML

■ Thẻ (tag) HTML

- Thẻ HTML là một trong các thành phần quan trọng tạo nên một trang Web.
- Mỗi thẻ được sử dụng để tạo ra một phần tử nhất định, từ đó cấu trúc nên một trang Web
- Thẻ mở, thẻ đóng: `<html>...</html>`, `<body>...</body>` thường đi với nhau thành cặp
- Thẻ Tự đóng: `
`

1.1. Tao cấu trúc trang web bằng HTML

■ Một trang web bằng HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Tiêu đề của trang web</title>
  </head>
  <body>
    <h1>Cấu trúc của một trang web bằng HTML</h1>
    <p>Đây là phần nội dung của website</p>
  </body>
</html>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ DOCTYPE

- Thông báo để trình duyệt web biết phiên bản HTML đang được sử dụng
- HTML có nhiều phiên bản khác nhau như HTML5, HTML 4, HTML 3, XHTML... Khai báo <DOCTYPE> được các trình duyệt sử dụng để xác định loại văn bản HTML.
 - Khai báo <DOCTYPE> phải được đặt trước tất cả các thẻ khác.
 - Khai báo <DOCTYPE> không phải là thẻ.

1.1. Tạo cấu trúc trang web bằng HTML

■ DOCTYPE

- HTML5 là phiên bản mới nhất và đang được sử dụng phổ biến trên hầu hết các website.
- HTML phiên bản 5 (hay HTML5, lưu ý số 5 viết liền với HTML) khai báo <DOCTYPE> sẽ như sau

```
<!DOCTYPE html>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Meta

- Metadata (siêu dữ liệu) là dạng dữ liệu miêu tả về dữ liệu.
- Thẻ <meta> cung cấp siêu dữ liệu về tài liệu HTML.
- Siêu dữ liệu không hiển thị thông tin cho người dùng nhưng bộ máy tìm kiếm có thể đọc được
- Thẻ <meta> thường được sử dụng để mô tả trang, từ khoá, tác giả của tài liệu, lần cập nhật cuối cùng và các siêu dữ liệu khác.

1.1. Tạo cấu trúc trang web bằng HTML

■ Meta – các thuộc tính

Thuộc tính	Giá trị	Mô tả
charset	<i>character_set</i>	Xác định loại mã hoá ký tự cho tài liệu HTML
content	<i>text</i>	Cung cấp giá trị liên kết với http-equiv hoặc thuộc tính tên
http-equiv	content-type default-style refresh	Cung cấp HTTP header cho thông tin/giá trị của thuộc tính nội dung
name	application-name author description generator keywords	Xác định tên cho siêu dữ liệu
scheme	<i>format/URI</i>	Không được hỗ trợ trong HTML5 Chỉ định scheme để giải thích các giá trị của các thuộc tính nội dung

1.1. Tạo cấu trúc trang web bằng HTML

■ Meta – các ví dụ

Ví dụ 1 – Định nghĩa các từ khoá cho máy tìm kiếm:

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

Ví dụ 2 – Định nghĩa mô tả cho trang web của bạn:

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Meta – các ví dụ

Ví dụ 3 – Định nghĩa tác giả của trang:

```
<meta name="author" content="I am Superman">
```

Ví dụ 4 – Tải lại trang sau 30 giây:

```
<meta http-equiv="refresh" content="30">
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Meta – các ví dụ

```
<head>
<meta charset="UTF-8">
<meta name="description" content="Hướng dẫn chém gió">
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
<meta name="author" content="I am Superman">
</head>
```

1.1. Tao cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

- Thẻ P – phân đoạn văn bản
- Mỗi đoạn văn bản sẽ được xuống dòng và cách nhau với một tỷ lệ nhất định.

`<p> Đây là một đoạn văn sẽ được phân đoạn theo tỷ lệ nhất định. </p>`

`<p> Học HTML căn bản dành cho người mới bắt đầu. </p>`

1.1. Tạo cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

- Thẻ hr – tạo đường gạch ngang
- Thẻ hr dùng để tạo một đường gạch ngang, tương tự thẻ br thẻ hr cũng chỉ dùng thẻ mở hoặc thẻ đóng

```
<p> Dưới đây là một đường gạch ngang <hr/> </p>
```

1.1. Tao cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

- Thẻ pre – định dạng sẵn
- Thẻ pre thường được dùng để đăng một câu đối thoại hoặc in một đoạn mã để cho dễ phân biệt với các văn bản thông thường.

```
<pre> Thẻ định dạng sẵn </pre>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

– Thẻ b – Bôi đậm

```
<p> Hướng dẫn học <b>HTML</b> căn bản </p>
```

– Thẻ strong – nhấn mạnh

– Thẻ strong tương tự như thẻ b. Nhưng có ý nhấn mạnh hơn giúp các công cụ tìm kiếm dễ dàng phân biệt.

```
<p> Đến khi nào thì mới hết <strong>FA</strong> </p>
```


1.1. Tạo cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

- Thẻ big – chữ lớn

```
<p> Hướng dẫn học <big>lập trình</big> web </p>
```

- Thẻ small – chữ nhỏ
- Ngược lại với thẻ big, thẻ small dùng để định dạng chữ nhỏ hơn bình thường

```
<p> Hướng dẫn học <small>lập trình</small> web </p>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

- Thẻ i – in nghiêng

```
<p> Hướng dẫn học <i>lập trình</i> web </p>
```

- Thẻ u – gạch dưới chữ viết

```
<p> Hướng dẫn học <u>lập trình</u> web </p>
```

- Thẻ code – định dạng một đoạn mã

```
<p> Thẻ Meta được khai báo nằm trong cặp thẻ <code><head> </head></code></p>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

- Thẻ del – gạch ngang chữ

```
<p> Bạn không nên <del>lười nhác</del> web </p>
```

- Thẻ sub – chữ nhỏ ở dưới

```
<p> Giải phương trình sau  $X^{2} + 1$  </p>
```

- Thẻ sup – chữ ở trên đầu dạng số mũ

```
<p> Giải phương trình sau  $2X^{3} + 3X + 1$  </p>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Văn bản – các thẻ liên quan

- Thẻ span – tương tự thẻ P nhưng có ý nghĩa rộng hơn bao quanh một vùng văn bản

```
<span> Hướng dẫn học lập trình thiết kế website </span>
```

- Thẻ mark – tô sáng chữ viết

```
<p> Hướng dẫn học lập trình web <mark>HTML</mark> </p>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Văn bản – căn lề

- sử dụng thuộc tính text-align với các giá trị left, right, center và justify

```
<p style="text-align: center"> Căn giữa văn bản </p>
```

- Định nghĩa phong chữ sử dụng

```
<p style="font-family: Arial"> Font chữ Arial </p>
```

- Kích thước chữ

```
<p style="font-size: 14px"> Chữ có kích thước 14px</p>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Liên kết

- Các trang thường được liên kết với nhau bằng một liên kết (link) thông qua thẻ <a>

```
<a href="http://BlogLoi.com" target="_blank" title="Học lập trình web" rel="follow, index" >BlogLoi.com</a>
```

- Trong đó:

- href : là đường link đến website đích
- title : là tiêu đề cho đường link
- rel : là các thuộc tính dành cho SEO. follow, index là cho phép các công cụ tìm kiếm như google, yahoo vào website của bạn để cập nhật bài viết mới

1.1. Tạo cấu trúc trang web bằng HTML

■ Liên kết

- Các trang thường được liên kết với nhau bằng một liên kết (link) thông qua thẻ <a>

```
<a href="http://BlogLoi.com" target="_blank" title="Học lập trình web" rel="follow, index" >BlogLoi.com</a>
```

- target : gồm các thuộc tính sau (giá trị mặc định là _self)
 - **_blank** : chuyển link qua tab mới
 - **_self** : chuyển link trên tab hiện tại
 - **_parent** : chuyển link đến tab cha của tab hiện tại
 - **_top** : nhảy tới tab hiện tại và thường dùng trong iframe chạy tới trang gốc luôn

1.1. Tạo cấu trúc trang web bằng HTML

■ Hình ảnh và video

- Thẻ để chèn các hình ảnh vào các trang web
- Thẻ img chỉ có thẻ mở mà không có thẻ đóng

```

```

■ Trong đó:

- src="hinh.jpg" là đường dẫn hình ảnh trong website
- width : chiều rộng hình ảnh
- height : chiều cao hình ảnh
- border : đường viền xung quanh hình ảnh
- alt : Mô tả về hình ảnh . Thuộc tính này rất quan trọng trong SEO vì vậy bạn nên có

1.1. Tạo cấu trúc trang web bằng HTML

■ Hình ảnh và video

- Thẻ <video> để chèn các video mp4 vào các trang web
- Thẻ video có thẻ mở và thẻ đóng

```
<video width="100" height="100"> <source src="movie.mp4"> </video>
```

■ Trong đó:

- source src="movie.mp4" là đường dẫn video trong thư mục website.
- width : chiều rộng hình ảnh
- height : chiều cao hình ảnh

1.1. Tạo cấu trúc trang web bằng HTML

■ Danh sách

- Thẻ ol – danh sách theo số thứ tự

```
<ol style="list-style-type: property's value"> </ol>
```

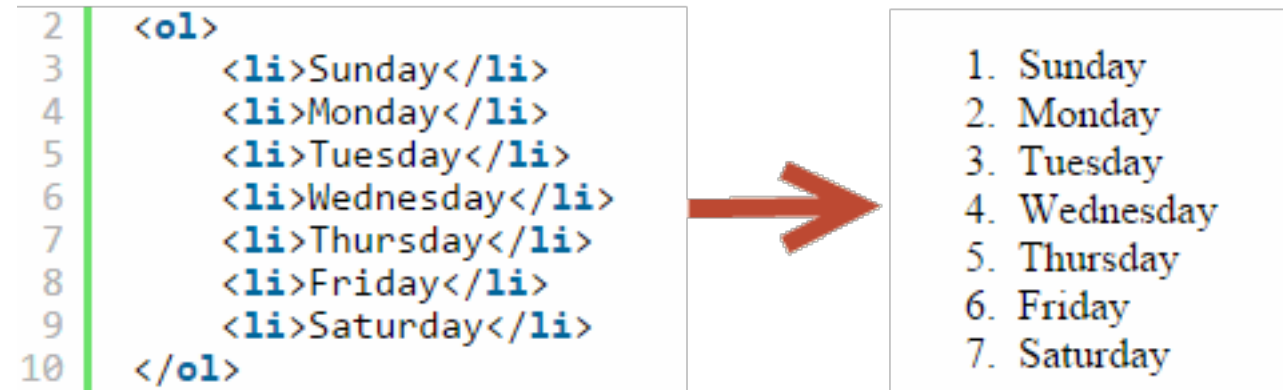
■ Trong đó property's value bao gồm các giá trị

Property's Value	Example
decimal	1, 2, 3, ...
lower-alpha	a, b, c, ...
upper-alpha	A, B, C, ...
lower-roman	i, ii, iii, ...
upper-roman	I, II, III, ...

1.1. Tạo cấu trúc trang web bằng HTML

■ Danh sách

- Mỗi giá trị nằm trong cặp thẻ



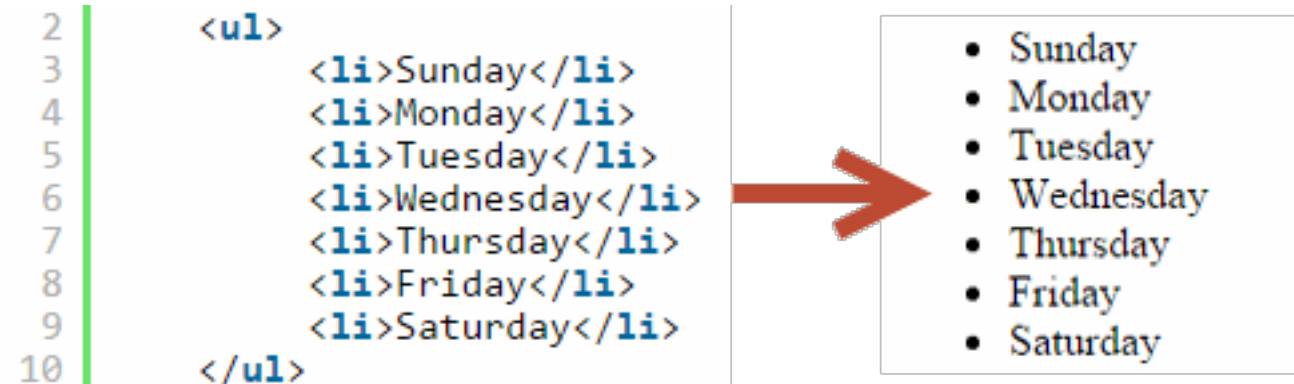
1.1. Tạo cấu trúc trang web bằng HTML

■ Danh sách

- Thẻ ul – danh sách theo ký hiệu

`<ul style="list-style-type : property's name"> `

- Trong đó property's name bao gồm các thuộc tính sau
 - Square
 - Disc (hình tròn tô)
 - Circle



1.1. Tạo cấu trúc trang web bằng HTML

■ Danh sách

- Các thẻ có thể lồng nhau

```
2      <ul>
3          <li>Sunday
4              <ul style="list-style-type: square">
5                  <li>HTML</li>
6                  <li>CSS</li>
7                  <li>PHP</li>
8              </ul>
9          </li>
10         <li>Monday</li>
11         <li>Tuesday</li>
12         <li>Wednesday</li>
13         <li>Thursday</li>
14         <li>Friday</li>
15         <li>Saturday</li>
16     </ul>
```



- Sunday
 - HTML
 - CSS
 - PHP
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday

1.1. Tạo cấu trúc trang web bằng HTML

■ Danh sách

- Thẻ dl – danh sách định nghĩa
- Tương tự 2 thẻ trên, thẻ <dl> có giá trị nằm trong các thẻ con <dt> và <dd>

```
15 <dl>
16     <dt>Common noun</dt>
17     <dd>It is a name of an Object in general such as pencil</dd>
18     <dt>Proper noun</dt>
19     <dd>It is a unique name of a person or a place</dd>
20 </dl>
```

1.1. Tạo cấu trúc trang web bằng HTML

■ Các đối tượng đa phương tiện

- Thẻ embed – nhúng đối tượng (flash, nhạc)
- Dùng để nhúng các đối tượng flash, video mp3, mp4.

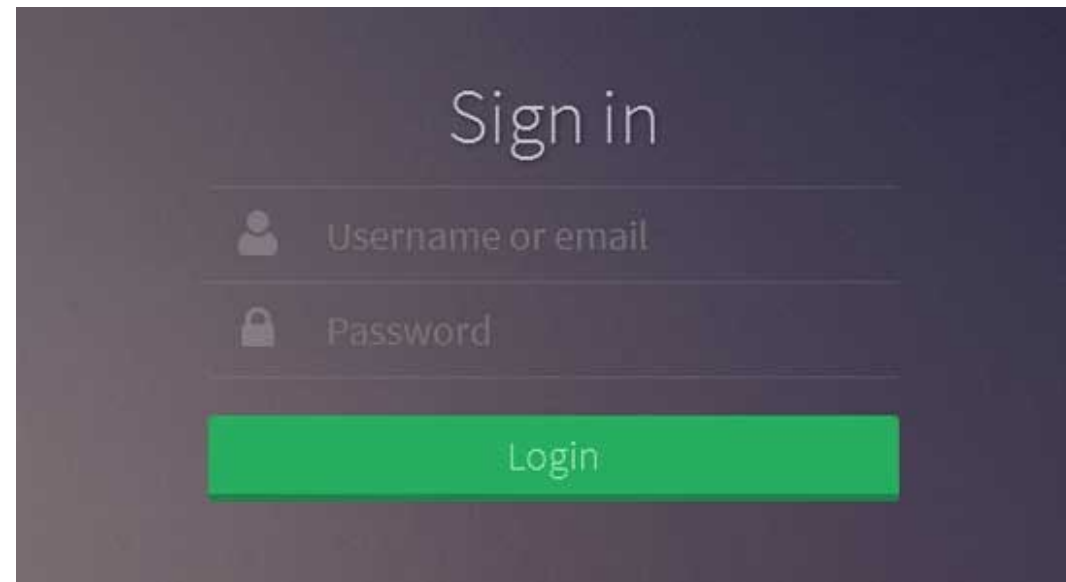
```
<embed src="nhac.mp3" width="400" height="300"> </embed>
```

```
<embed src="tao-bang-trong-sql-server2012.mp4" width="400" height="300"> </embed>
```

2.1.8 Form và các thành phần Form

■ Form và các thành phần

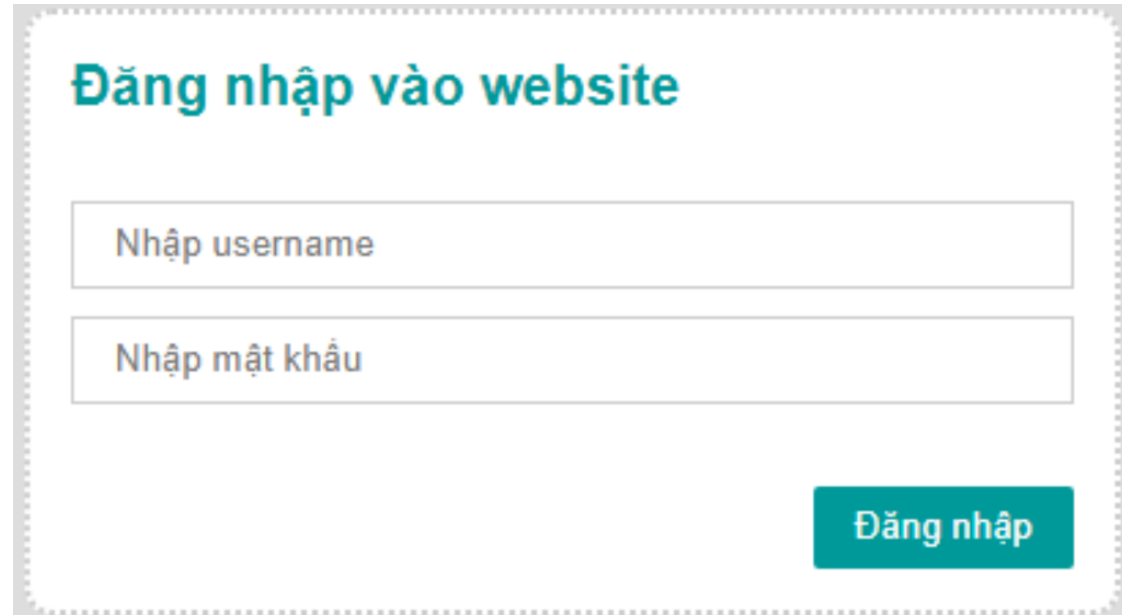
- Chứa các điều khiển khác để nhập liệu
- HTML form được định nghĩa bằng cặp thẻ `<form></form>`
- Các thẻ nằm giữa cặp thẻ `<form></form>` được gọi là các thành phần của Form

A screenshot of a "Sign in" form. The form is centered on a dark blue background. It has the title "Sign in" at the top. Below the title are two input fields: the first is labeled "Username or email" with a user icon, and the second is labeled "Password" with a lock icon. At the bottom of the form is a green rectangular button with the text "Login" in white.

2.1.8 Form và các thành phần Form

■ Form và các thành phần

- Chứa các điều khiển khác để nhập liệu
- HTML form được định nghĩa bằng cặp thẻ <form></form>
- Các thẻ nằm giữa cặp thẻ <form></form> được gọi là các thành phần của Form

A screenshot of a login form titled "Đăng nhập vào website" in blue text. Below the title are two input fields: the first is labeled "Nhập username" and the second is labeled "Nhập mật khẩu". At the bottom right of the form is a blue button with the text "Đăng nhập" in white.

2.1.8 Form và các thành phần Form

■ Form và các thành phần

- Trường Text và password

```
<input type="text" name="username" size="30" />
```

```
<input type="password" name="pass" size="30" />
```

■ Checkbox (Hộp kiểm)

- Cho phép chọn nhiều thành phần từ danh sách đưa ra

```
<input type="checkbox" name="opt1" /> Send my to email
```

```
<input type="checkbox" name="opt2" /> Send my to phone
```

2.1.8 Form và các thành phần Form

■ Form và các thành phần

- Fieldset và legend: để nhóm các thành phần trong Form; legend: định nghĩa chú thích cho Fieldset

<form>

<fieldset>

<legend>Dòng chú thích</legend>

Nội dung form

</fieldset>

</form>

■ Radio button

- Radio chỉ phép phép chọn một thành phần từ danh sách đưa ra, các phần tử trong danh sách phải có cùng tên

<input type="radio" name="gender" checked /> Male <input type="radio" name="gender" /> Female

2.1.8 Form và các thành phần Form

■ File Upload

- Để tải các tệp lên server

```
<input type="file" name="FILE" size="40" />
```

■ Button

```
<input type="button" name="button" value="Click me" />
```

```
<input type="submit" name="submit" value="Submit" />
```

```
<input type="reset" name="cancel" value="Reset" />
```

2.1.8 Form và các thành phần Form

■ DropDown list

- Tạo một danh sách thả xuống

```
<select name="job">  
  <option value="Student">Student</option>  
  <option value="Bussiness">Bussiness</option>  
  <option value="Manager">Manager</option>  
  <option value="Other">Other...</option>  
</select>
```

■ Vùng nhập liệu văn bản (textarea)

- `<textarea name="comment" cols="30" rows="5"></textarea>`

2.1.8 Form và các thành phần Form

■ Submit button

- Nút bấm để gửi thông tin từ form kết hợp với thuộc tính action

```
<form action="/action_page.aspx">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

2.1.8 Form và các thành phần Form

■ Thuộc tính method - GET

- `<form action="/action_page.php" method="get">`
- Phương thức mặc định khi gửi dữ liệu đi từ Form
- Dữ liệu sẽ hiển thị trên thanh address

`/action_page.aspx?firstname=Mickey&lastname=Mouse`

2.1.8 Form và các thành phần Form

■ Thuộc tính method - GET

- Đặt dữ liệu lên URL
- Kích thước của URL giới hạn bởi 3000 ký tự
- Không sử dụng GET với các dữ liệu nhạy cảm (mật khẩu)
- Hữu dụng khi người dùng muốn ghi dấu kết quả
- GET hữu dụng cho dữ liệu thông thường

2.1.8 Form và các thành phần Form

■ Thuộc tính method - POST

- Không hiển thị dữ liệu lên thanh address
- không giới hạn dung lượng dữ liệu được gửi đi

2.1.8 Form và các thành phần Form

■ Tạo form sau

ĐĂNG KÝ	
Full name	<input type="text" value="Nhập fullname"/>
Username	<input type="text" value="Nhập tên"/>
Password	<input type="password" value="Nhập mật khẩu"/>
Nhập lại Password	<input type="password" value="Nhập lại mật khẩu"/>
Giới tính	<input type="radio"/> Nam <input type="radio"/> Nữ
Ngày sinh	<input type="text" value="Ngày 1"/> <input type="text" value="Tháng 1"/> <input type="text" value="Năm 2015"/>
Địa chỉ	<input type="text"/>
Avatar	<input type="button" value="Choose File"/> No file chosen
Sở thích	<input type="checkbox"/> Xem Phim <input type="checkbox"/> Thẻ Thao <input type="checkbox"/> Web
<input type="button" value="Reset"/> <input type="button" value="OK"/>	

2.1.8 Form và các thành phần Form

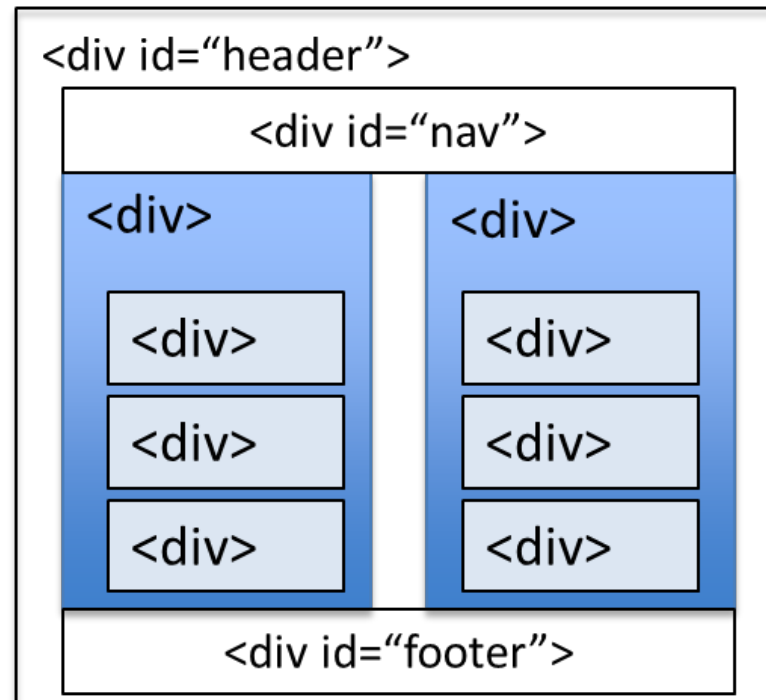
■ HTML Table

- `<table><table>`
- `<tr></tr>`: dòng trong bảng.
- `<td></td>`: cột trong bảng
- `Border` = “đường viền của bảng”
- `Colspan`: số cột bị chiếm.
- `Rowspan`: số dòng bị chiếm

2.1.9 Nhóm các thành phần với Div & Span

■ Nhóm các thành phần với Div & Span

- Thẻ `<div>` (viết tắt của từ division) được sử dụng để tạo một khu vực kiểu block trên website.
- Được sử dụng để gom nhóm tập hợp các phần tử trên website



2.1.9 Nhóm các thành phần với Div & Span

■ Nhóm các thành phần với Div & Span

- Thẻ `<div>` (viết tắt của từ division) được sử dụng để tạo một khu vực kiểu block trên website.
- Được sử dụng để gom nhóm tập hợp các phần tử trên website

```
<div>  
    <p>  
        Any kind of content here. Such as &lt;p>, &lt;table>.  
        You name it!  
    </p>  
</div>
```

2.1.9 Nhóm các thành phần với Div & Span

■ Nhóm các thành phần với Div & Span

- Thẻ `` được dùng để gom nhóm các thẻ inline HTML
- Span thường dùng để định dạng cho một chữ hoặc một chuỗi chữ.
- Span không làm thay đổi nội dung và hiển thị của đoạn văn đó.

```
<div id="header">
  <div id="userbar"> Hi there, <span class="username">Chris Marasti-
    Georg</span> |
    <a href="/edit-profile.html">Profile</a> |
    <a href="http://www.bowl.sk.com/_ah/logout?...">Sign out</a>
  </div>
  <h1><a href="/">Bowl<span class="sk">SK</span></a></h1>
</div>
```

2.1.10 Tối ưu trang web cho các bộ máy tìm kiếm (SEO On Page)

- **Tối ưu trang web cho các bộ máy tìm kiếm (SEO On Page)**
 - Là kỹ thuật tối ưu hóa trang web cho các công cụ tìm kiếm để có được thứ hạng cao hơn trong kết quả hiển thị
 - Tối ưu nội dung và mã nguồn HTML của một trang Web
 - Bao gồm: tối ưu URL thân thiện, tối ưu tiêu đề, tối ưu từ khóa

2.1.10 Tối ưu trang web cho các bộ máy tìm kiếm (SEO On Page)

■ Tối ưu URL thân thiện

- URL cần có từ khóa liên quan đến nội dung của trang
- Ví dụ:
 - URL chưa được tối ưu `nguyenvana.vn/p=231`
 - URL đã được tối ưu SEO: `nguyenvana.vn/hoc-lap-trinh-web-co-ban`
- Giữ cho URL ngắn và đơn giản
 - Danh từ, động từ, tính từ
 - Ví dụ: `hoc-tot-lap-trinh-web`

2.1.10 Tối ưu trang web cho các bộ máy tìm kiếm (SEO On Page)

- Tiêu đề luôn bắt đầu với từ khóa
 - Đây là nội dung khá quan trọng
 - Khi tìm kiếm kết quả trả về:
 - Tiêu đề, mô tả và URL
 - Các kết quả đầu tiên luôn có đặc điểm: từ khóa ở đầu tiêu đề, URL và mô tả
 - Tiêu đề nên chứa những tính từ mô tả và các con số: best, offer, buy, cheap, powerful, easy way, easy method,

2.1.10 Tối ưu trang web cho các bộ máy tìm kiếm (SEO On Page)

- Từ khóa luôn nằm trong đoạn đầu tiên của bài viết
- Đặt từ khóa quan trọng trong thẻ H1
- Sử dụng từ khóa biến thể hay dạng Latent Semantic Indexing (LSI – từ khóa liên quan hoặc đồng nghĩa với từ khóa chính)

2.1.10 Tối ưu trang web cho các bộ máy tìm kiếm (SEO On Page)

- Sử dụng liên kết nội bộ giúp tăng thứ hạng cho bài viết:
 - Sử dụng 2-3 liên kết nội bộ trong một bài viết
 - Giúp điều hướng người đọc đến các tài nguyên khác (liên quan đến bài viết mà họ đang đọc) nhằm tăng tỉ lệ người duyệt (visitor)
 - Giúp cho các công cụ tìm kiếm dễ dàng thu thập dữ liệu trang web
 - Giúp cho các con bộ nhanh chóng hiểu được bài viết, nội dung

2.1.10 Tối ưu trang web cho các bộ máy tìm kiếm (SEO On Page)

- Tối ưu trải nghiệm người dùng:
 - Viết đúng chính tả và cú pháp
 - Nội dung bài viết phải đủ dài
 - Nội dung bài viết chất lượng: người đọc hài lòng sẽ chia sẻ, giúp tăng thứ hạng của website

2.2. Định dạng, trình bày trang web bằng CSS

- CSS là từ viết tắt của *Cascade Style Sheet*, là ngôn ngữ giúp trình duyệt hiểu được các thiết lập định dạng và bố cục cho trang web.
- CSS cho phép điều khiển thiết kế của nhiều thành phần HTML chỉ với duy nhất 1 vùng chọn CSS.
- Điều này giúp giảm thiểu thời gian thiết kế và chỉnh sửa, khi có thể tách biệt được cấu trúc (HTML) và định dạng (CSS).

2.2. Định dạng, trình bày trang web bằng CSS

- Có 03 cách để sử dụng CSS

- "Inline CSS": Áp dụng trực tiếp trên một đối tượng nhất định bằng thuộc tính style:

```
<span style="font-weight:bold; text-decoration:underline; color:#FF0000;">
```

Đoạn text cần in đậm, gạch chân, màu đỏ

```
</span>
```

2.2. Định dạng, trình bày trang web bằng CSS

■ Có 03 cách để sử dụng CSS

- "Internal CSS":
- Đặt CSS ở đầu trang Web để áp dụng kiểu dáng cho toàn bộ trang ấy, khi đó chỉ cần đặt đoạn CSS vào trong cặp thẻ <style> rồi đặt vào trong phần header của Web (giữa <head> và </head>):

```
<style type="text/css"> body {font-family:verdana; color:#0000FF;}
```

/ Kiểu chữ trong trang Web là "Verdana", màu chữ thông thường là màu xanh dương */*

```
</style>
```

2.2. Định dạng, trình bày trang web bằng CSS

- Có 03 cách để sử dụng CSS

- "External CSS": Đặt các thuộc tính CSS vào một tệp tin riêng biệt (*.css), khi đó có thể tham chiếu đến từ nhiều trang Web khác nhau:

```
body {font-family:verdana; color:#0000FF;}
```

```
<link rel="stylesheet" type="text/css" href="style.css"/>
```


1.2. Định dạng, trình bày trang web bằng CSS

- **Mức độ ưu tiên của CSS**

- Mức độ ưu tiên khi áp dụng CSS như sau:

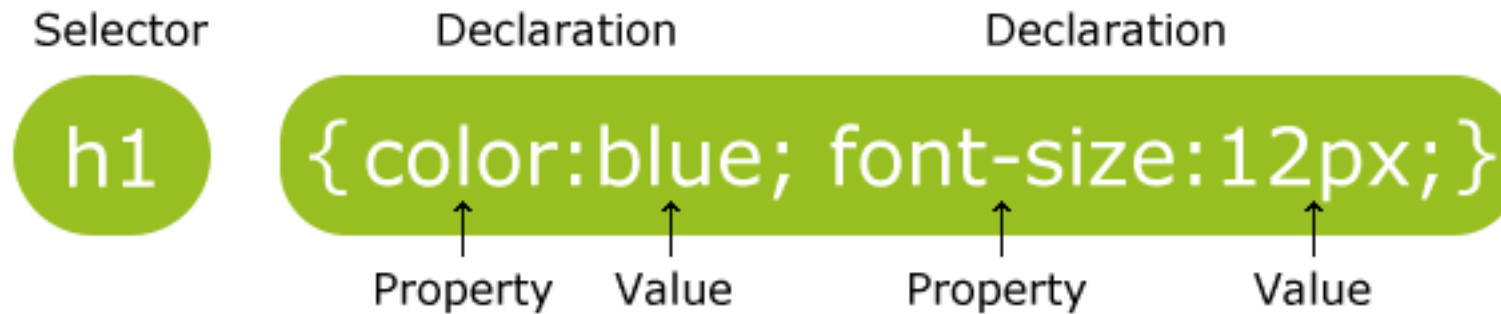
External CSS < Internal CSS < Inline CSS

1.2. Định dạng, trình bày trang web bằng CSS

- Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác.
- Sau đó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.



1.2. Định dạng, trình bày trang web bằng CSS



1.2. Định dạng, trình bày trang web bằng CSS

```
body {  
    background: #ffffff;  
    /* trang Web sẽ có nền màu trắng */  
    font-family: Verdana;  
    /* font chữ mặc định là Verdana */  
    color: #ff0000;  
    /* màu chữ mặc định là màu đỏ */  
}
```

1.2. Định dạng, trình bày trang web bằng CSS

■ Cấu trúc một đoạn CSS

```
vùng chọn {  
    thuộc tính : giá trị;  
    thuộc tính: giá trị;  
    .....  
}
```

- **Vùng chọn:** là cách xác định các thẻ HTML dựa trên cấu trúc phân cấp của HTML.
- Vùng chọn có thể được tạo nên dựa trên nhiều yếu tố như định danh (id), tên lớp (class), quan hệ cha – con – hậu duệ...

1.2. Định dạng, trình bày trang web bằng CSS

■ Cấu trúc một đoạn CSS

```
vùng chọn {  
    thuộc tính : giá trị;  
    thuộc tính: giá trị;  
    .....  
}
```

- **Thuộc tính:** là yếu tố bạn muốn thay đổi ở các thẻ HTML thuộc vùng chọn.
- **Giá trị:** mỗi thuộc tính sẽ yêu cầu một giá trị khác nhau.
- Có thể là 1 từ khóa định sẵn (none, block), một tên màu hay mã màu (black, white, #000, #FFFFFF), hay một giá trị kích thước tính bằng px, em, rem, %.

1.2. Định dạng, trình bày trang web bằng CSS

■ Các vùng chọn

Thuộc tính	Ví dụ	Mô tả
<i>.class</i>	.intro	Chọn tất cả các thành phần với class="intro"
<i>#id</i>	#firstname	Chọn tất cả các thành phần với id="firstname"
*	*	Chọn tất cả các thành phần
<i>Element</i>	p	Chọn tất cả các thành phần <p>
<i>element,element</i>	div, p	Chọn tất cả các thành phần <div> và tất cả thành phần <p>
<i>element element</i>	div p	Chọn tất cả các thành phần <p> bên trong thành phần <div>

1.2. Định dạng, trình bày trang web bằng CSS

■ Các vùng chọn

- #ID: dấu # được sử dụng để chọn một thẻ HTML có định danh cụ thể (ID)
- Không thể tái sử dụng

```
<div id="vi-du">
```

```
<h2>Thạch Phạm Blog</h2>
```

```
<p>Cách. WordPress blog lớn nhất Việt Nam.</p>
```

```
<div>
```

```
<p class="grand-child">Yêu màu tím. Sống nội tâm. Hay khóc thầm. Nhưng không gay.</p>
```

```
</div>
```

```
</div>
```


1.2. Định dạng, trình bày trang web bằng CSS

■ Các vùng chọn

- Dấu “.” đại diện cho tên lớp.
- Nhiều thẻ HTML có thể có cùng tên lớp.
- sự khác nhau giữa ID và Class là: ID chỉ cho phép chọn 1 phần tử, còn Class cho phép bạn chọn nhiều thẻ HTML cùng lúc.

```
<div id="vi-du">
```

```
  <h2>Thạch Phạm Blog</h2>
```

```
  <p>Cạch. WordPress blog lớn nhất Việt Nam.</p>
```

```
</div>
```

```
  <p class="grand-child">Yêu màu tím. Sống nội tâm. Hay khóc thầm. Nhưng không gay.</p>
```

```
</div>
```

```
</div>
```

1.2. Định dạng, trình bày trang web bằng CSS

- Các vùng chọn

- **.Class**: vùng chọn giúp xác định tất cả các thẻ HTML cùng thuộc một lớp

.intro	Chọn tất cả các thành phần có class="intro"
--------	---

1.2. Định dạng, trình bày trang web bằng CSS

■ Các vùng chọn

- **A B**: là vùng chọn hậu duệ (descendant).
- Vùng chọn này sẽ xác định tất cả các thẻ B nằm bên trong thẻ A, không kể phân cấp, ngôi thứ như là con, là cháu, là chắt chút chít gì đó.

div p Chọn tất cả `<p>` nằm bên trong các thành phần `<div>`

1.2. Định dạng, trình bày trang web bằng CSS

■ Các vùng chọn

- **A > B**: là vùng chọn con (child), và có nét tương đồng với vùng chọn hậu duệ.
- vùng chọn này chỉ xác định tất cả các thẻ B là con trực tiếp của thẻ A, chứ không nằm bên trong thẻ nào khác

`div > p`

Chọn tất cả `<p>` mà cha mẹ là thành phần `<div>`

1.2. Định dạng, trình bày trang web bằng CSS

■ Các vùng chọn

- **A + B**: Phạm vi áp dụng là những element liền kề ngay sau selector được khai báo(X).
- Trong ví dụ có thể hiểu là: Css áp dụng cho thẻ p đầu tiên đứng ngay sau thẻ ul sẽ có color: red;.

```
ul + p { color: red; }
```

1.2. Định dạng, trình bày trang web bằng CSS

- Các vùng chọn
 - X:visited và X:link

```
a:link { color: red; }  
a:visited { color: purple; }
```

1.2. Định dạng, trình bày trang web bằng CSS

- Các vùng chọn

- X[href="foo"]

- Ví dụ bên dưới: định dạng cho tất cả các thẻ liên kết đến http://

```
a[href="http://net.tutsplus.com"] {  
    color: #1f6053; /* nettuts green */  
}
```

X[href*="nettuts"]

```
a[href*="tuts"] {    color: #1f6053; /*  
nettuts green */}
```

1.2. Định dạng, trình bày trang web bằng CSS

■ Các vùng chọn

<code>div</code> matches <code>div</code> elements <code><div></code>	<code>#welcome</code> # matches the id <code><div id="welcome"></code>	<code>div .button</code> match any <code>.button</code> that's a child of a <code>div</code>
<code>.button</code> . matches the class <code></code>	<code>div.button</code> match <code>div</code> s with class "button" <code><div class="button"></code>	<code>div > .button</code> match any <code>.button</code> that's a <u>direct</u> child of a <code>div</code>
<code>a:hover</code> matches <code>a</code> elements that the cursor is hovering over	<code>ul li:first-child</code> match the first item of a list. there's <code>last-child</code> too.	<code>a[href^="http"]</code> match links where the <code>href</code> attribute starts with "http" (external links)
<code>:checked</code> matches if a checkbox or radio button is checked	<code>tr:nth-child(odd)</code> match every other row of a table (make stripes!)	<code>div:not(#header)</code> match all <code>div</code> s except the one with id "header"

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

Thuộc tính	Ví dụ	Mô tả
<code>background</code>	<code>background: #ff0000;</code>	Định dạng nền (background) cho thành phần.
<code>border</code>	<code>border: 1px solid #ff0000;</code>	Định dạng đường viền cho thành phần.
<code>border-collapse</code>	<code>border-collapse: collapse;</code>	Thuộc tính <code>border-collapse</code> xác định đường viền của <code>table</code> có tách biệt ra hay không.
<code>border-spacing</code>	<code>border-spacing: 10px;</code>	Xác định khoảng cách giữa các đường viền của các cột lân cận.
<code>bottom</code>	<code>bottom: 10px;</code>	Xác định vị trí dưới cùng của thành phần được định vị trí.
<code>caption-side</code>	<code>caption-side: bottom;</code>	Xác định vị trí một chú thích của <code>table</code> .

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

clear	clear: both;	Xác định 2 bên của phần tử (left, right), nơi mà phần tử float không được cho phép.
clip	clip: rect(0,0,50px,10px);	Xác định đoạn cho thành phần khi sử dụng thuộc tính position có giá trị "absolute".
color	color: #ff0000;	Xác định màu sắc cho text.
content	content: "."	Sử dụng kèm với bộ chọn ":before", ":after" để chèn nội dung được tạo.
counter-increment	counter-increment: section;	Gia tăng một hoặc nhiều counter (sắp xếp có thứ tự, có hiển thị số)
counter-reset	counter-reset: subsection;	Tạo hoặc reset một hoặc nhiều counter.
cursor	cursor: pointer;	Xác định kiểu con trỏ chuột được hiển thị.

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

direction	<code>direction: ltr;</code>	Xác định hướng cho văn bản.
display	<code>display: inline;</code>	Xác định loại hiển thị của thành phần.
empty-cells	<code>empty-cells: hide;</code>	Xác định có hay không có đường viền và nền trong một cột rỗng của table
float	<code>float: left;</code>	Xác định có hay không một thành phần được float.
font	<code>font: 12px arial,sans-serif;</code>	Thiết lập font cho thành phần, bao gồm font chữ, độ rộng, ...
height	<code>height: 50px;</code>	Thiết lập chiều cao của thành phần.
left	<code>left: 10px;</code>	Xác định vị trí bên trái của thành phần định vị trí (như position)

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

<code>letter-spacing</code>	<code>letter-spacing: 2px;</code>	Tăng hoặc giảm khoảng cách giữa các ký tự trong đoạn text.
<code>line-height</code>	<code>line-height: 1.5;</code>	Thiết lập chiều cao giữa các dòng.
<code>list-style</code>	<code>list-style: decimal;</code>	Thiết lập kiểu cho một danh sách.
<code>margin</code>	<code>margin: 15px;</code>	Canh lề cho thành phần.
<code>max-height</code>	<code>max-height: 200px;</code>	Thiết lập chiều cao tối đa của thành phần.
<code>max-width</code>	<code>max-width: 900px;</code>	Thiết lập chiều rộng tối đa của thành phần.
<code>min-height</code>	<code>min-height: 100px;</code>	Thiết lập chiều cao tối thiểu của thành phần.

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

<code>overflow</code>	<code>overflow: scroll;</code>	Xác định điều gì sẽ xảy ra nếu một thành phần box tràn nội dung.
<code>padding</code>	<code>padding: 15px;</code>	Thiết lập các thuộc tính padding trong một khai báo.
<code>page-break-after</code>	<code>page-break-after: alway;</code>	Xác định các phân chia văn bản ngay sau thành phần.
<code>page-break-before</code>	<code>page-break-before: alway;</code>	Xác định các phân chia văn bản ngay trước thành phần.
<code>page-break-inside</code>	<code>page-break-inside: alway;</code>	Xác định các phân chia văn bản ngay bên trong thành phần.
<code>position</code>	<code>position: absolute;</code>	Xác định loại của phương pháp định vị trí cho thành phần.
<code>quotes</code>	<code>" " " "</code>	Thiết lập các loại dấu ngoặc bao ngoài khi nhúng một trích dẫn.

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

<code>right</code>	<code>right: 10px;</code>	Xác định vị trí bên phải của thành phần định vị trí (như position)
<code>table-layout</code>	<code>table-layout: fixed;</code>	Thiết lập các thuật toán layout được sử dụng cho table.
<code>text-align</code>	<code>text-align: center;</code>	Sắp xếp các nội dung theo chiều ngang.
<code>text-decoration</code>	<code>text-decoration: underline;</code>	Xác định các trang trí thêm cho text.
<code>text-indent</code>	<code>text-indent: 10px;</code>	Ghi rõ thụt đầu dòng của dòng đầu tiên trong một khối văn bản.
<code>text-transform</code>	<code>text-transform: uppercase;</code>	Thiết lập các ký tự viết hoa cho văn bản.
<code>top</code>	<code>top: 10px;</code>	Xác định vị trí bên trên của thành phần định vị trí (như position)

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

<code>vertical-align</code>	<code>vertical-align: middle;</code>	Sắp xếp các nội dung theo chiều dọc.
<code>visibility</code>	<code>visibility: visible;</code>	Xác định thành phần có được nhìn thấy hay không.
<code>white-space</code>	<code>white-space: nowrap;</code>	Xác định khoảng trắng có bên trong thành phần được xử lý như thế nào.
<code>width</code>	<code>width: 800px;</code>	Thiết lập chiều rộng cho thành phần.
<code>word-spacing</code>	<code>word-spacing: 5px;</code>	Tăng hoặc giảm không gian giữa các từ trong đoạn văn bản.
<code>z-index</code>	<code>z-index: 100;</code>	Thiết lập thứ tự xếp chồng nhau của một thành phần vị trí.
<code>vertical-align</code>	<code>vertical-align: middle;</code>	Sắp xếp các nội dung theo chiều dọc.

1.2. Định dạng, trình bày trang web bằng CSS

- Các thuộc tính của CSS
 - Đường viền của phần tử

`border: [size] [type] [color];`

`border: 1px solid red;`

I have borders on all sides.

1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

– Đường viền của phần tử

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

1.2. Định dạng, trình bày trang web bằng CSS

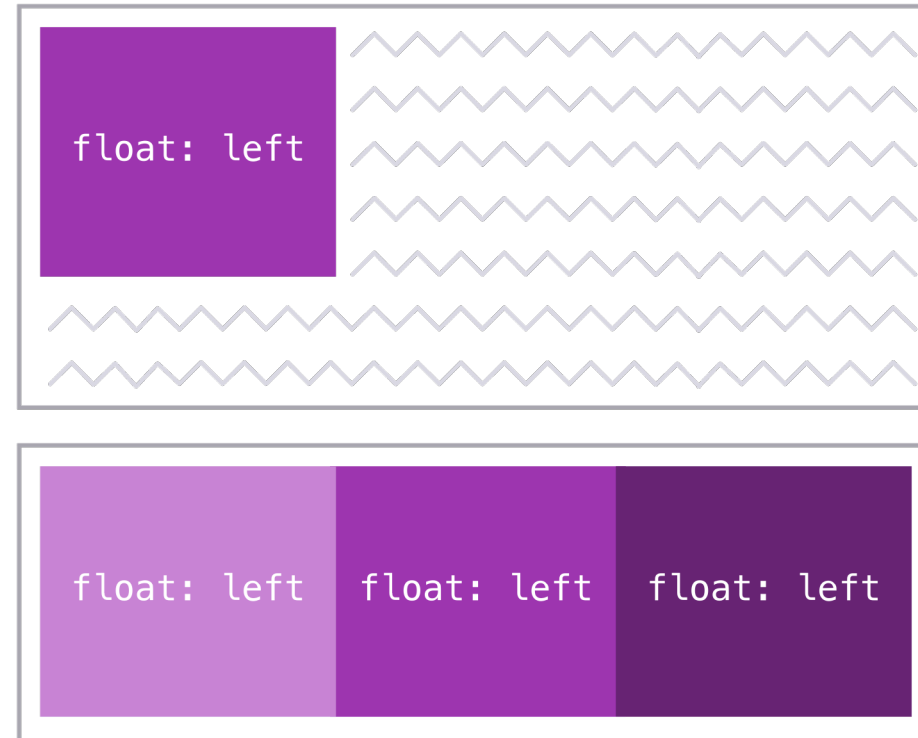
■ Các thuộc tính của CSS

– Float

- left – Thành phần nằm ở phía bên trái của vật chứa của nó
- Right - Thành phần nằm ở phía bên phải của vật chứa của nó
- none – Thành phần không nổi, đây là giá trị mặc định
- inherit - Thành phần kế thừa giá trị float của cha

1.2. Định dạng, trình bày trang web bằng CSS

- Các thuộc tính của CSS
 - Float

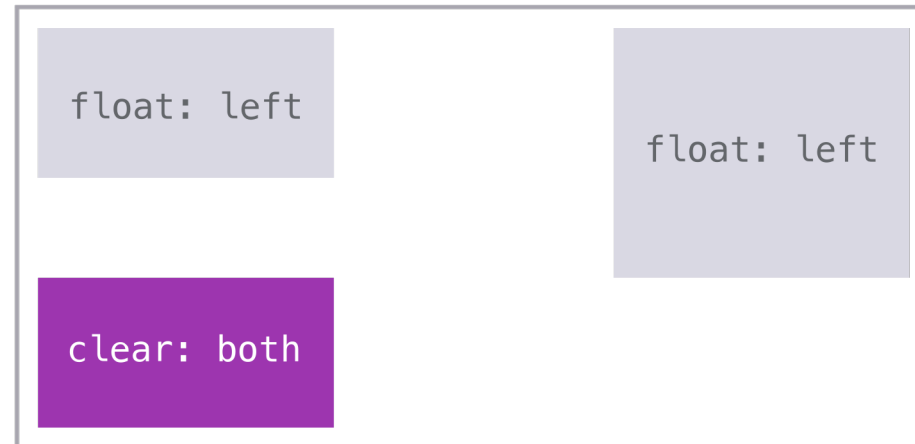


1.2. Định dạng, trình bày trang web bằng CSS

■ Các thuộc tính của CSS

– after

```
.row:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```



–clear

–left

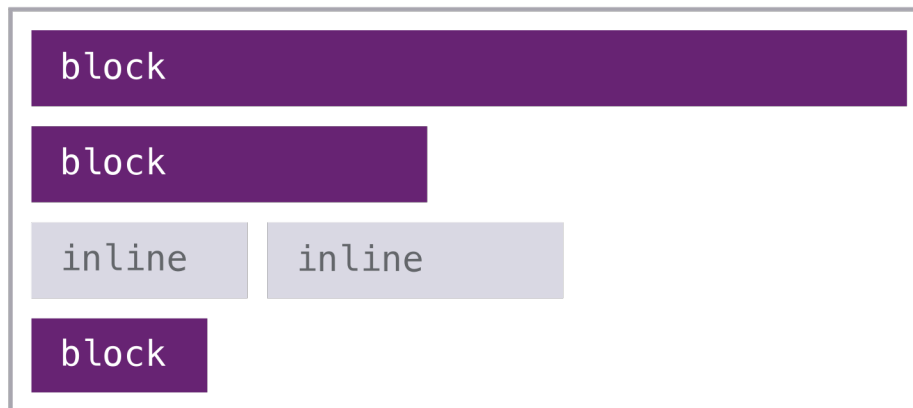
–Right

–none

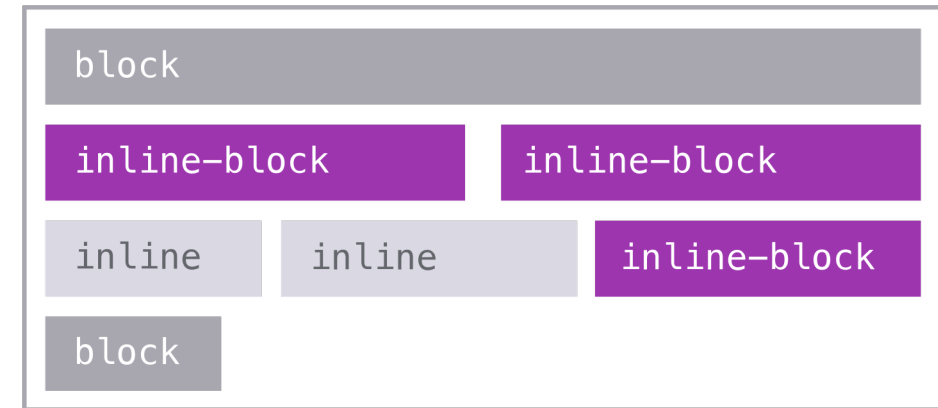
–Inherit

1.2. Định dạng, trình bày trang web bằng CSS

- Các thuộc tính của CSS
 - display: block, inline-block



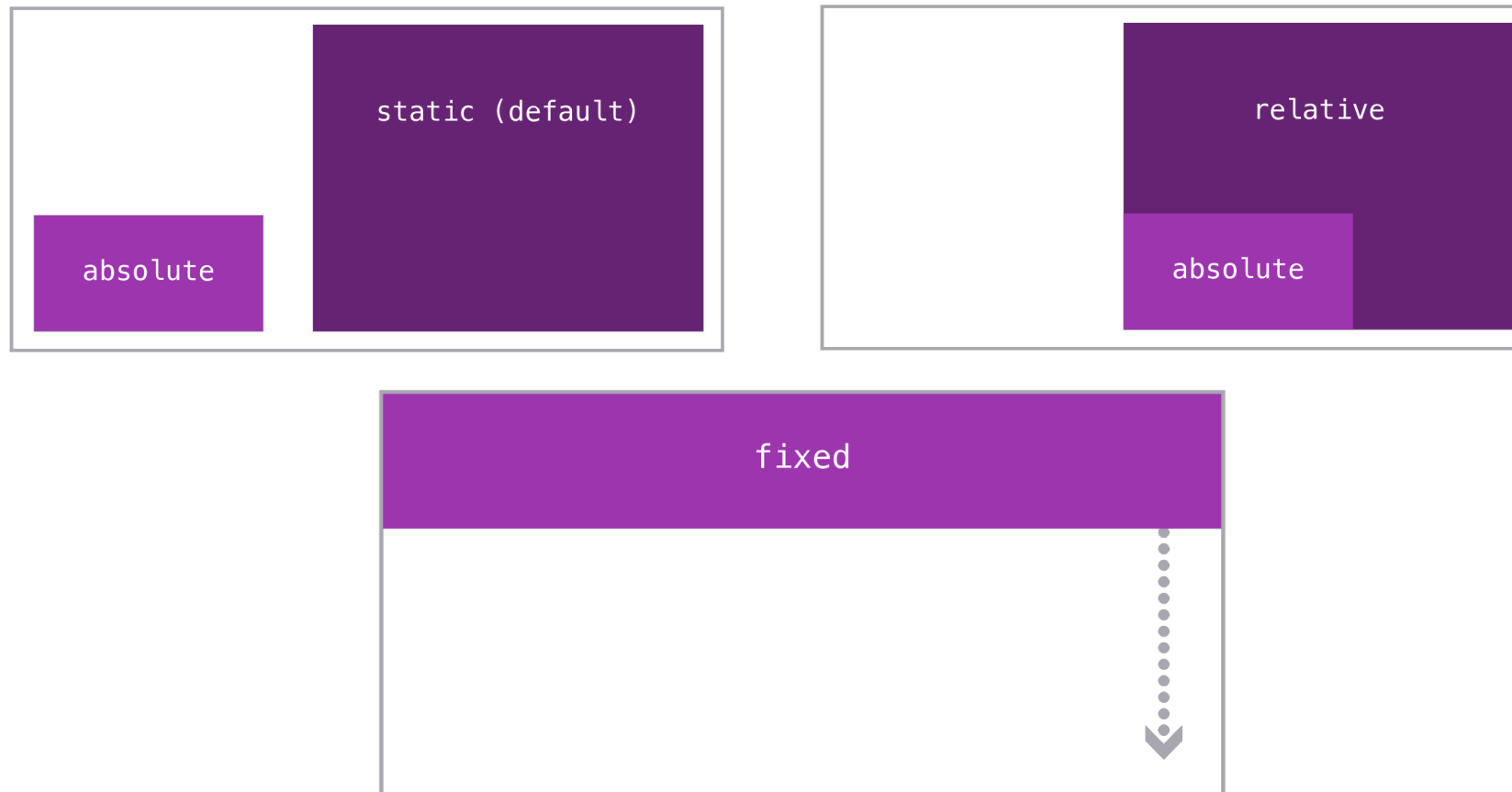
✓ padding ✓ margin ✓ width



✓ padding ✓ margin ✓ width

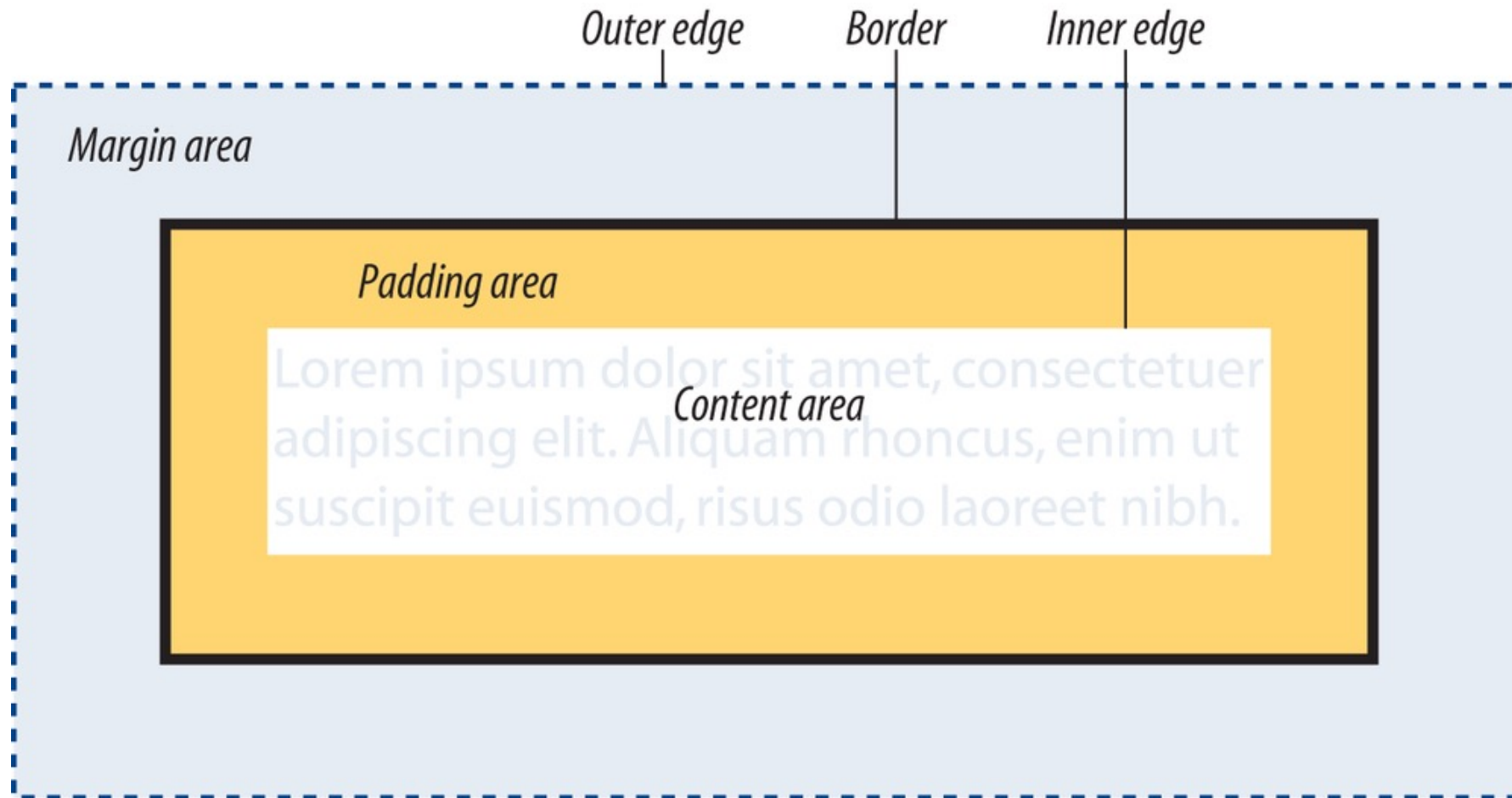
1.2. Định dạng, trình bày trang web bằng CSS

- Các thuộc tính của CSS
 - position: absolute, relative, fixed



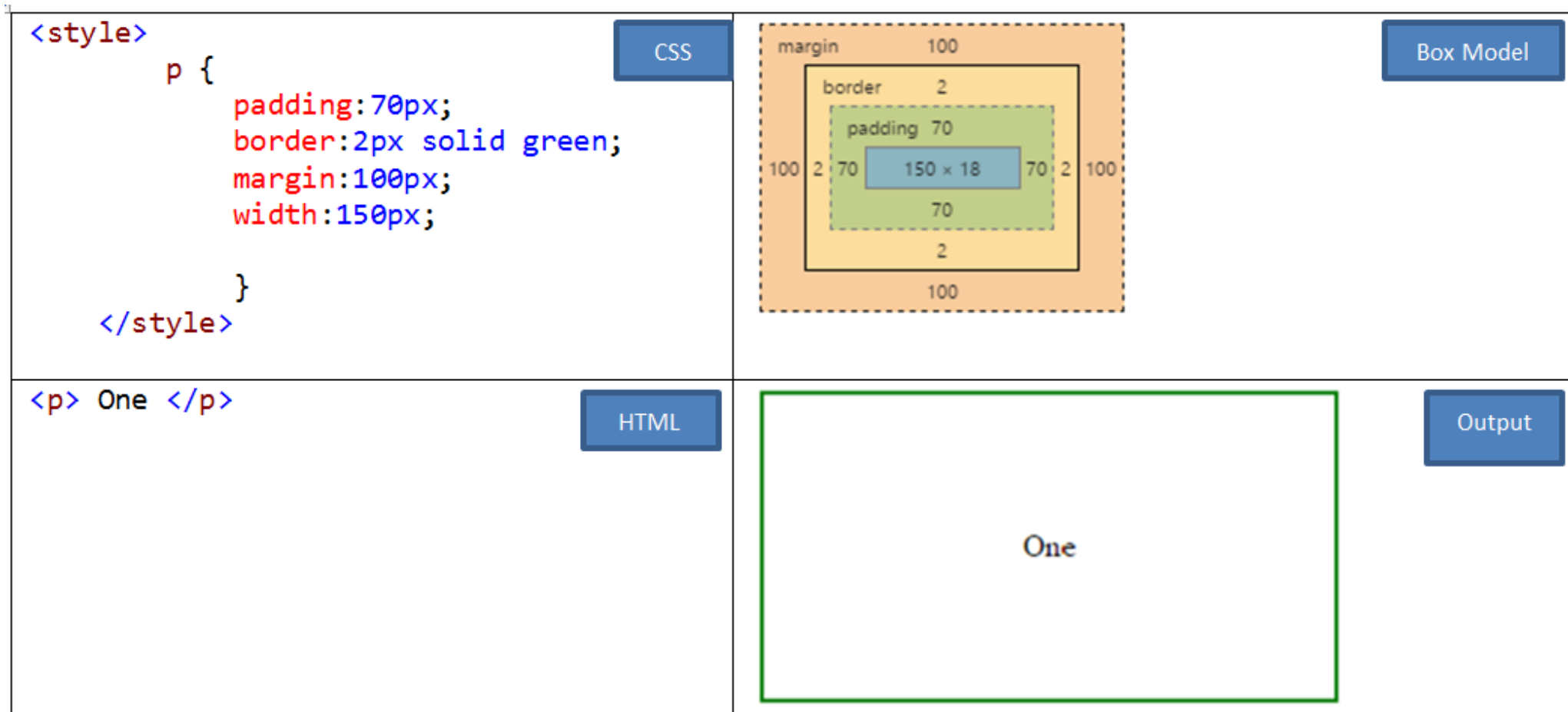
1.2. Định dạng, trình bày trang web bằng CSS

- CSS Box Model



1.2. Định dạng, trình bày trang web bằng CSS

■ CSS Box Model



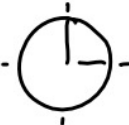
1.2. Định dạng, trình bày trang web bằng CSS

■ CSS Padding

there are 3 ways
to set padding

padding: 1em; ^{all sides}
padding: 1em 2em; ^{vertical horizontal}
padding: 1em 2em 3em 4em; ^{top right bottom left}

tricks to remember
the order

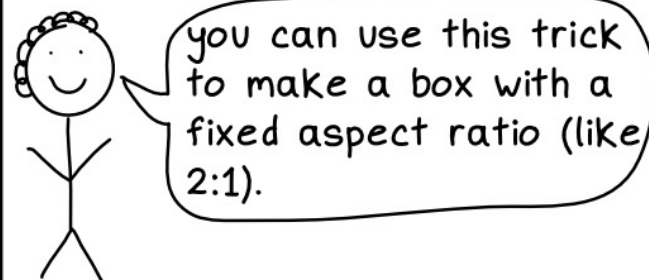
① **trouble**
top right bottom left
② it's clockwise 

you can also set
padding on just 1 side


examples:

padding-top: 1em;
padding-right: 10px;
padding-bottom: 3em;
padding-left: 4em;

padding-bottom: 20%; is a percentage
of the parent's width (not height!)
(same for top/bottom padding/margin)



```
.box {
  width: 100%;
  padding-bottom: 50%;
}
```

2:1 ratio →  this is the bottom padding

margin is set the
same way as padding

so is border-width: the
order is always
top, right, bottom, left

1.2. Định dạng, trình bày trang web bằng CSS

■ Các đơn vị trong CSS

– Đơn vị tuyệt đối:

- **px**: là đơn vị được sử dụng trên màn hình hiển thị, một px tương đương với một điểm ảnh
- và **pt**: đơn vị point, cứ 1 inch = 72pt

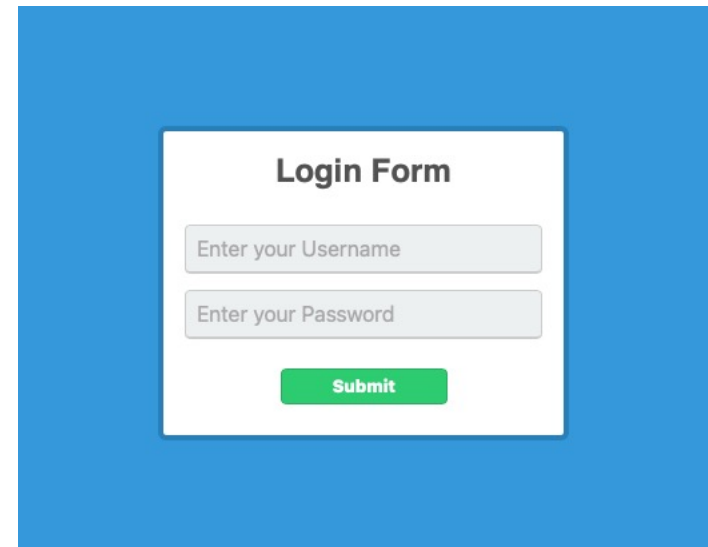
– Đơn vị tương đối:

- **%**: là đơn vị tham chiếu tỷ lệ so với một phần tử mẹ của nó dựa vào kích thước.
- **em**: là đơn vị tham chiếu tỷ lệ so với phần tử mẹ của nó dựa vào giá trị của thuộc tính font-size
- và **rem**

1.2. Định dạng, trình bày trang web bằng CSS

■ Tạo một trang login đơn giản

```
<body>
  <div id="divCenter">
    <form action="doLogin.aspx" method="post">
      <div class="box">
        <h1>Login Form</h1>
        <input type="text" name="sUsername" placeholder="Enter your Username" class="username">
        <input type="text" name="sPassword" placeholder="Enter your Password" class="username">
        <input type="submit" name="btnLogin" class="button">
      </div>
    </form>
  </div>
</body>
```

A visual representation of the login form described in the code. It features a white rectangular box centered on a solid blue background. Inside the box, the title "Login Form" is at the top. Below it are two text input fields: the first is labeled "Enter your Username" and the second is labeled "Enter your Password". At the bottom of the box is a green "Submit" button.

1.2. Định dạng, trình bày trang web bằng CSS

- Tạo một trang login đơn giản

```
.button{  
    background:#2ecc71;  
    width:125px;  
    padding-top:5px;  
    padding-bottom:5px;  
    color:white;  
    border-radius:4px;  
    border: #27ae60 1px solid;  
    margin-bottom: 20px;  
    margin-top: 20px;  
    float: left;  
    margin-left: 88px;  
    font-weight: 800;  
    font-size: 0.8em;  
}
```

1.3. Kỹ thuật dàn trang

- Một số bố cục trang web phổ biến
- Sắp xếp các lớp thành phần trong trang Web
- Giao diện tương thích đa thiết bị

1.3.1. Một số bố cục trang web phổ biến



Menu and content dynamic



Menu fixed, Content dynamic



Menu and content dynamic



3 columns, all dynamic



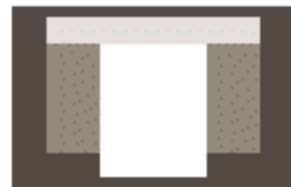
4 columns, all dynamic



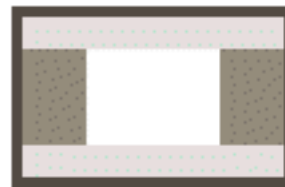
Menu floating



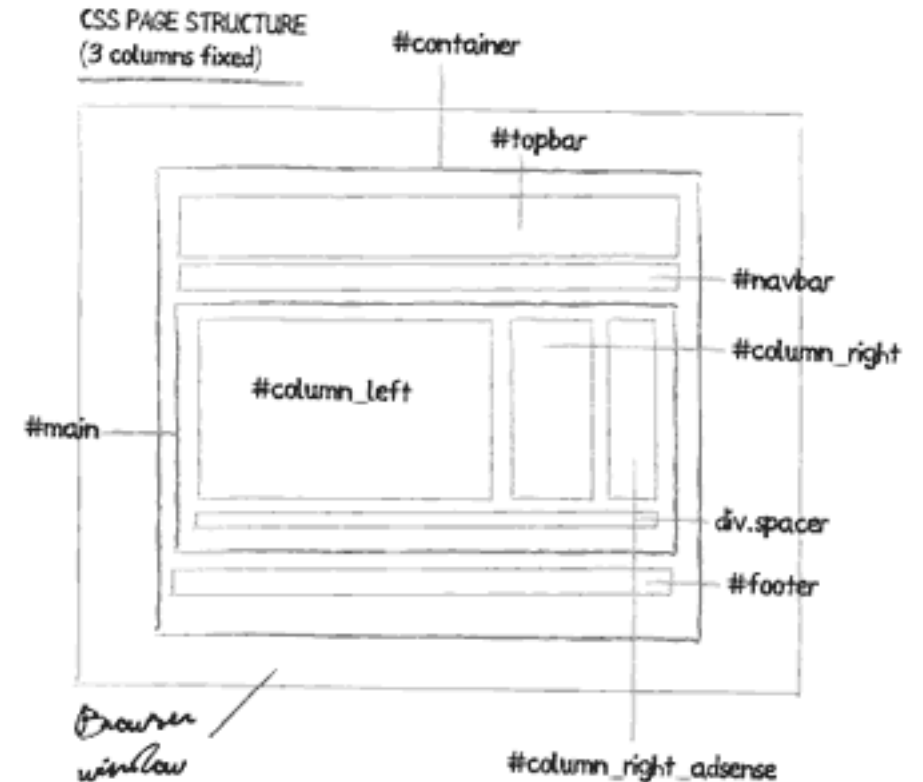
Menu fixed, content & header dynamic



3 columns fixed centered



dynamic with header and footer



1.3.1. Một số bố cục trang web phổ biến



1.3.2. Sắp xếp các lớp thành phần trong trang Web

- CSS cho phép sắp xếp các lớp (layer) hiển thị chồng lấp lên nhau
- Thuộc tính z-index được sử dụng cùng với position để tạo các lớp hiển thị (layer)
- Phần tử (thành phần) nào có z-index lớn hơn sẽ được hiển thị ở lớp trên

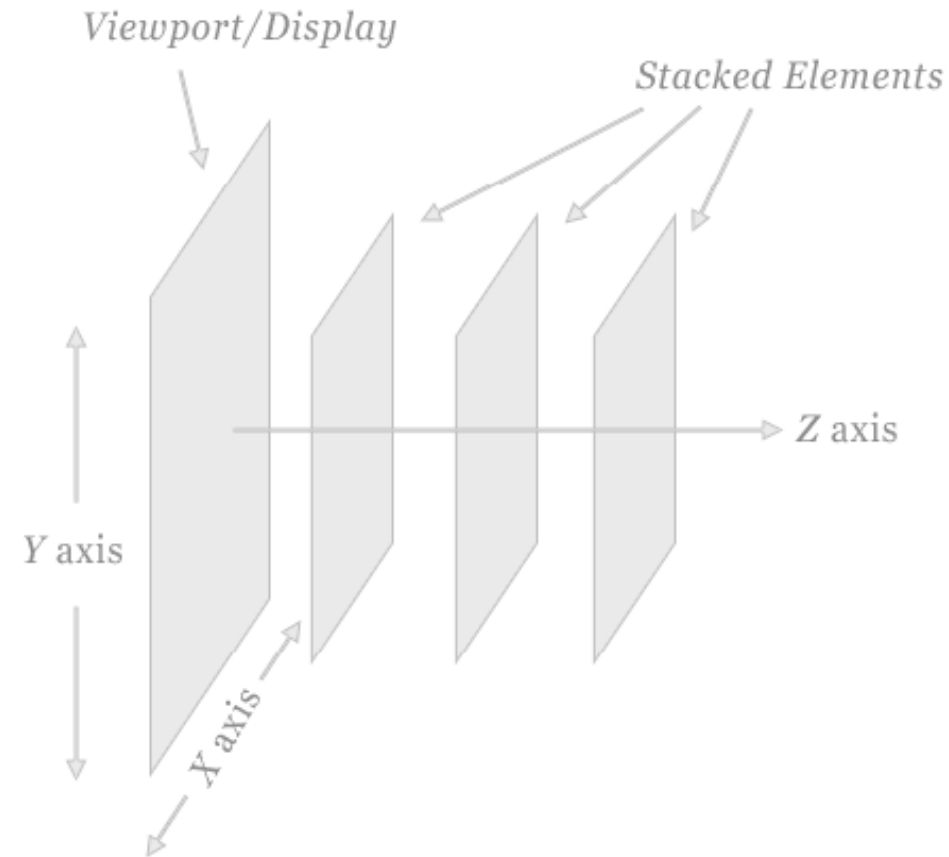
1.3.2. Sắp xếp các lớp thành phần trong trang Web

■ Cấu trúc

tag { z-index: giá trị; }

Thuộc tính	giá trị	Ví dụ	Mô tả
z-index	auto	z-index: auto;	Tự động sắp xếp thứ tự chồng nhau cho thành phần, đây là dạng mặc định.
	Giá trị	z-index: 10;	Sắp xếp thứ tự chồng nhau cho thành phần theo giá trị.
	inherit	z-index: inherit;	Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài).

1.3.2. Sắp xếp các lớp thành phần trong trang Web



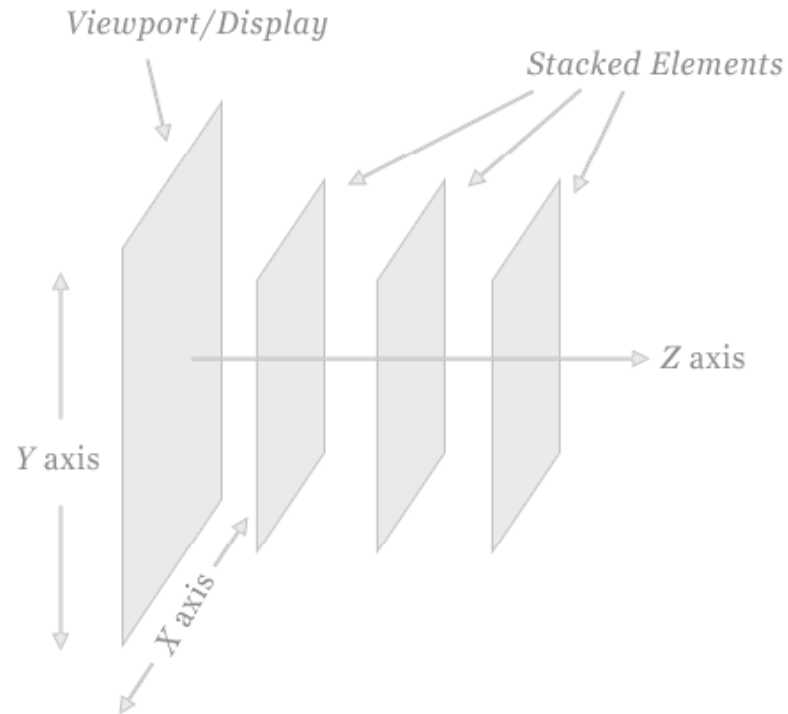
1.3.2. Sắp xếp các lớp thành phần trong trang Web

- Cấu trúc



1.3.2. Sắp xếp các lớp thành phần trong trang Web

- Cấu trúc



1.3.2. Sắp xếp các lớp thành phần trong trang Web

- Được sử dụng để tạo dropdown menu



```
.dropdown-content {  
  display: none;  
  position: absolute;  
  right: 0;  
  background-color: #e9d8f4;  
  min-width: 160px;  
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);  
  z-index: 1;  
}
```

```
.dropdown:hover .dropdown-content {  
  display: block;  
}
```

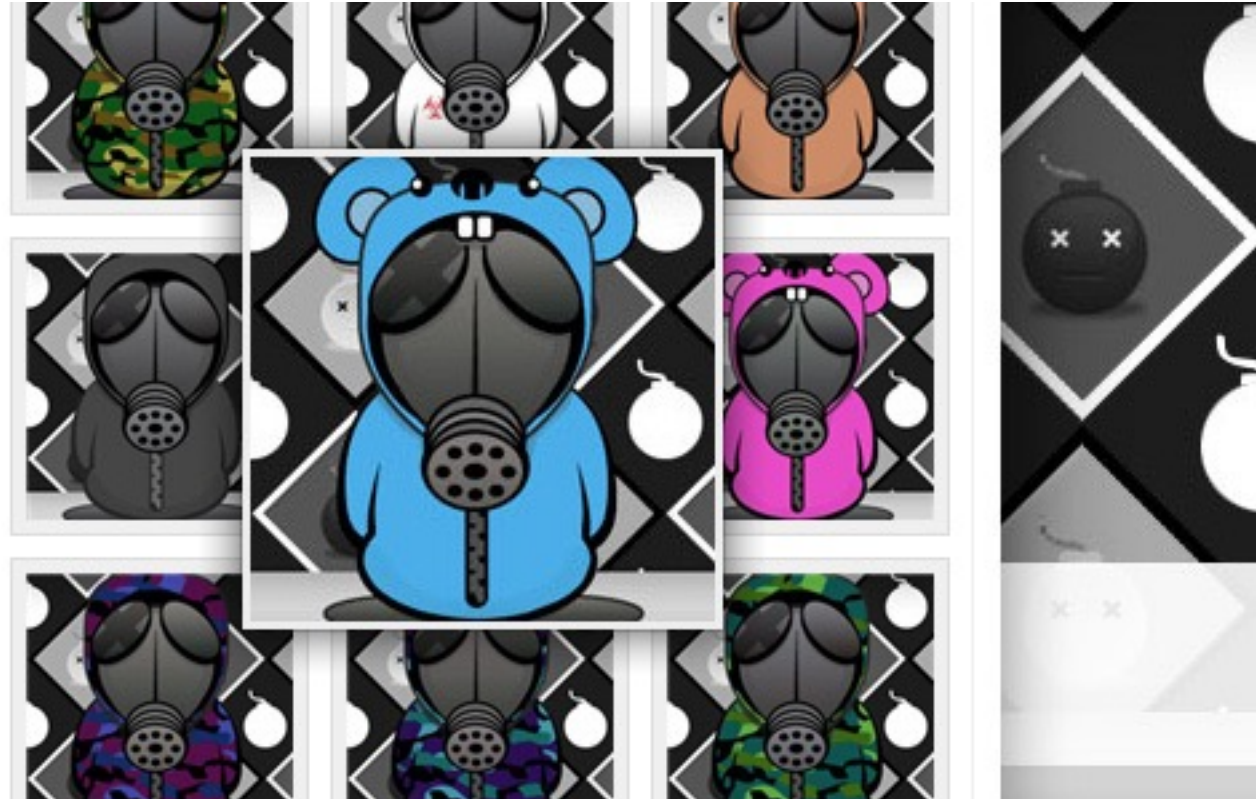
1.3.2. Sắp xếp các lớp thành phần trong trang Web

- Ứng dụng thực tế



1.3.2. Sắp xếp các lớp thành phần trong trang Web

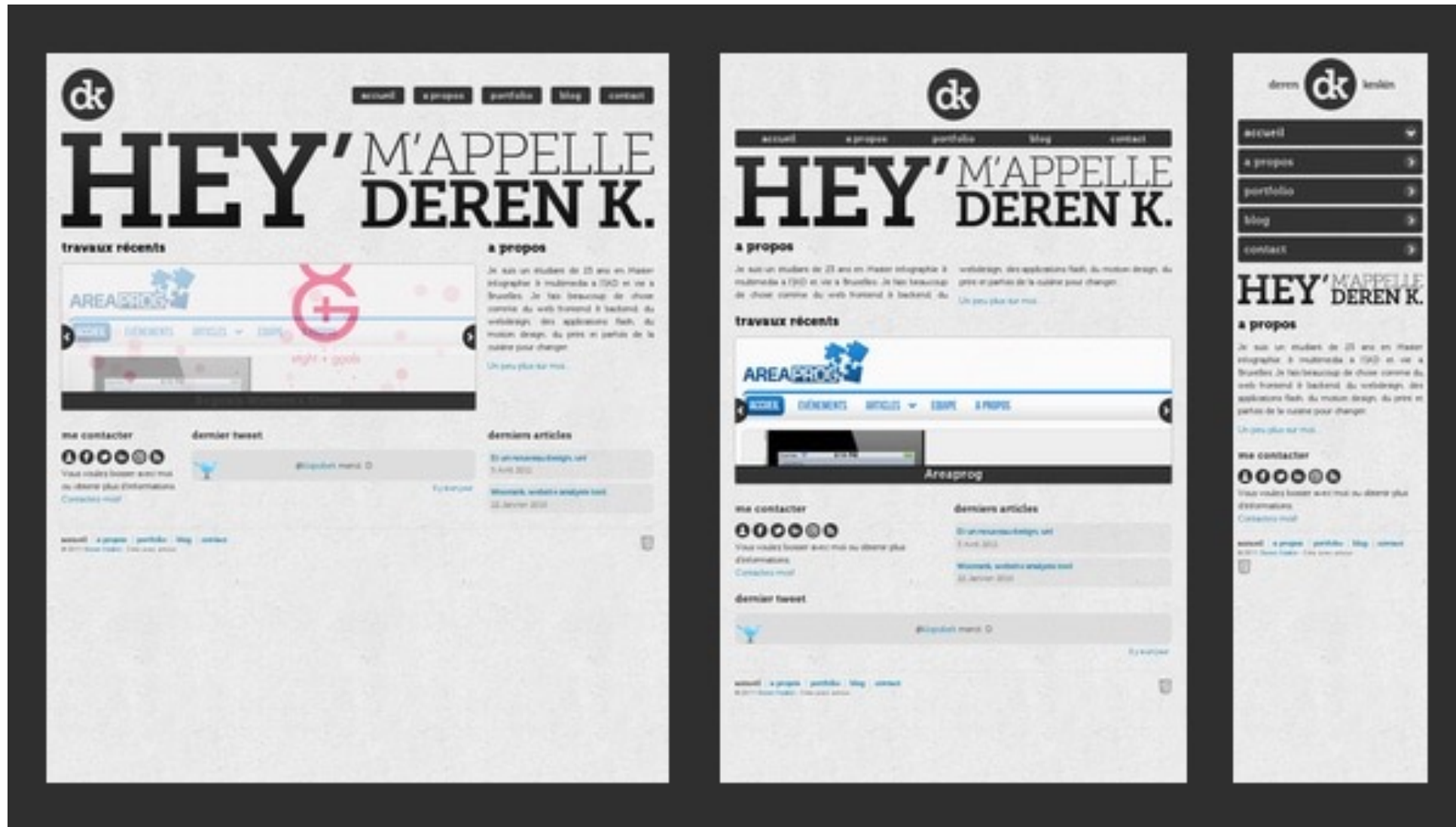
- Ứng dụng thực tế



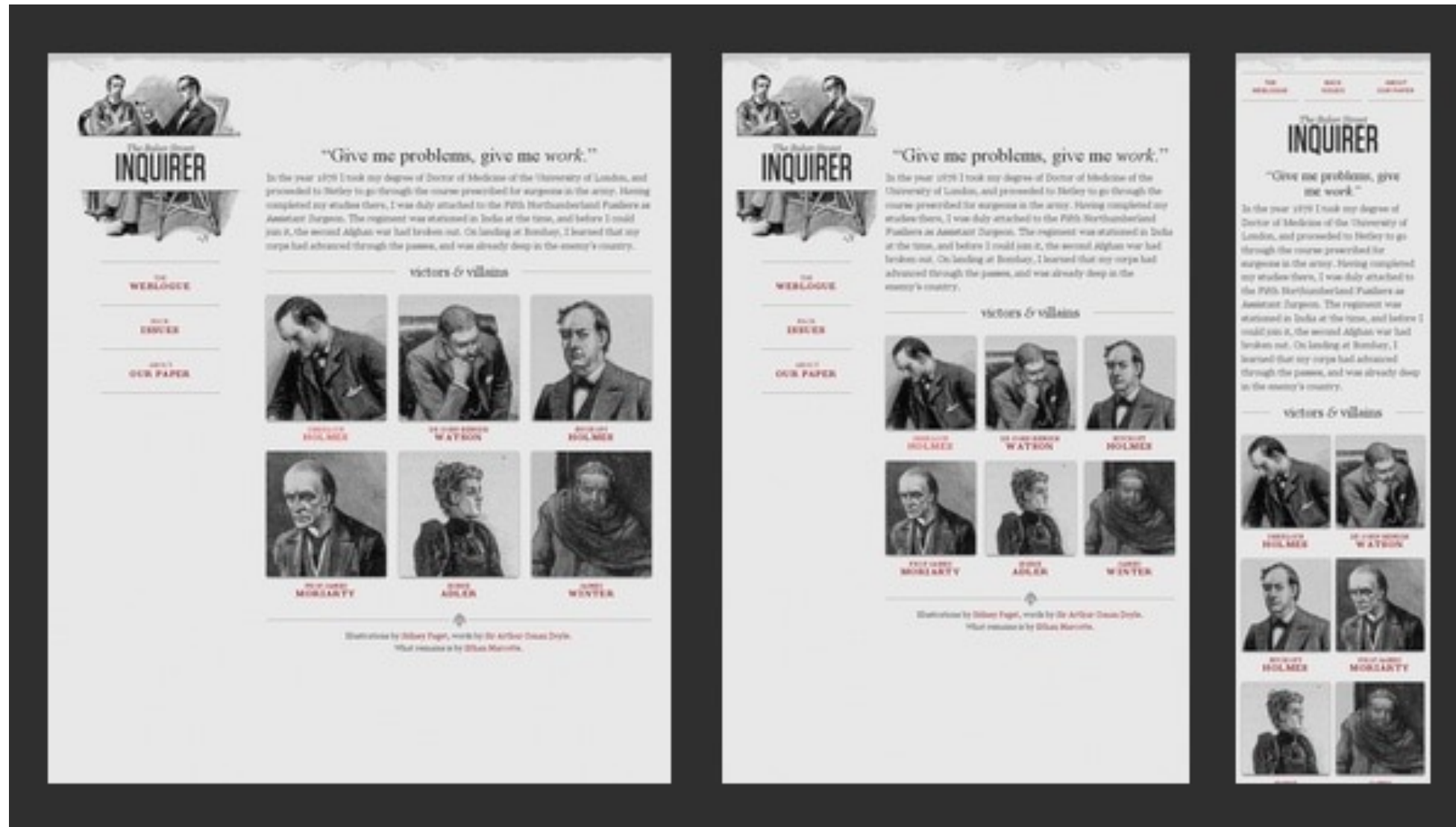
1.3.3. Giao diện tương thích đa thiết bị



1.3.3. Giao diện tương thích đa thiết bị



1.3.3. Giao diện tương thích đa thiết bị



1.3.3. Giao diện tương thích đa thiết bị

- Responsive Web Design (RWD)

- là phương pháp làm cho một thiết kế web có thể hiển thị đầy đủ chức năng trên màn hình nhỏ hay lớn, ở bất kì độ phân giải nào thông qua việc *sử dụng những đoạn code [CSS] phù hợp* trên thiết bị hiển thị.
- Dòng chảy hiển thị (the flow): người dùng sẽ lướt màn hình từ trên xuống dưới để xem nội dung hiển thị

1.3.3. Giao diện tương thích đa thiết bị

- Responsive Web Design (RWD)
 - Giúp tiết kiệm thời gian và chi phí khi không phải duy trì từng phiên bản Web khác nhau cho từng loại thiết bị
 - Giúp cải thiện Seo (Search Ranking) cho trang Web khi mà mọi URL để chỉ dẫn đến một địa chỉ URL.
 - Giúp dễ dàng bảo trì trang Web khi không phải thay đổi mã nguồn.

1.3.3. Giao diện tương thích đa thiết bị

- Quy tắc khi thực hiện RWD
 - Nội dung web phải luôn nằm trong giới hạn kích thước của chiều ngang màn hình,
 - 1. Không sử dụng các HTML element có chiều rộng cố định quá lớn
 - 2. Sử dụng CSS media queries để style cho từng thiết bị có chiều rộng khác nhau
 - 3. Sử dụng icon SVG thay cho icon hình ảnh thông thường (JPG, PNG,...)

1.3.3. Giao diện tương thích đa thiết bị

- Responsive Web Design (RWD)
 - Flexible grid based layout
 - Viewport
 - Grid View
 - Media Queries
 - Flexible media
 - FlexBox

1.3.3. Giao diện tương thích đa thiết bị

■ Viewport

- Là khung nhìn, là khu vực có thể nhìn thấy của người dùng về nội dung trong một trang web
- Viewport sẽ khác nhau với các thiết bị khác nhau, và sẽ nhỏ hơn trên điện thoại di động so với trên màn hình máy tính

1.3.3. Giao diện tương thích đa thiết bị

- Thiết lập Viewport

- Thuộc tính **width=device-width** đặt chiều rộng của trang web theo chiều rộng màn hình của thiết bị.
- Thuộc tính **initial-scale=1.0** thiết lập mức độ phóng ban đầu khi trang được trình duyệt tải lần đầu tiên, người dùng sẽ không thể zoom khi thuộc tính này có giá trị bằng 1.

1.3.3. Giao diện tương thích đa thiết bị

- Thiết lập Viewport

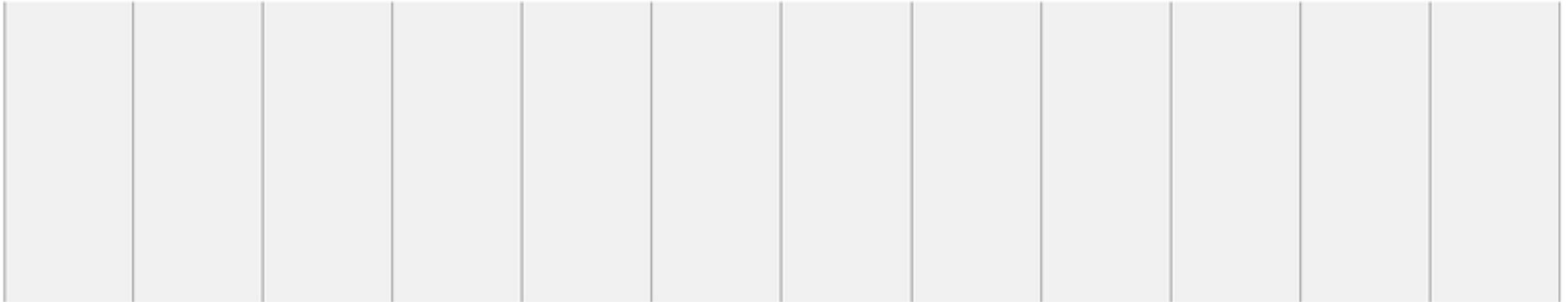
- `<meta name="viewport" content="width=device-width, initial-scale=1">`
- Thẻ `<meta>` viewport thiết lập cho trang web hiển thị tương ứng với kích thước của từng thiết bị khác nhau.

```
body {  
    background: #fff;  
    color: 333;  
}  
  
@media all and (max-width: 320px) {  
    body {  
        background: #e7e7e7;  
    }  
}
```

1.3.3. Giao diện tương thích đa thiết bị

- Grid-View

- Các trang web được dựa trên một Grid-View, có nghĩa là trang được chia thành các cột,



1.3.3. Giao diện tương thích đa thiết bị

- Grid-View

- 1 trang web bao gồm header, footer, 2 sidebar và phần content ở giữa, bố cục của trang web này được phân chia dựa trên 1 Grid-View.



1.3.3. Giao diện tương thích đa thiết bị

- Xây dựng một RWD Grid-View

This is a first css layout

Trang chủ FITHOU

CTMS

HOU

Fixed Full-height Side Nav

Try to scroll this area, and see how the sidenav sticks to the page

Notice that this div element has a left margin of 25%. This is because the side navigation is set to 25% width. If you remove the margin, the sidenav will overlay/sit on top of this div.

Also notice that we have set overflow:auto to sidenav. This will add a scrollbar when the sidenav is too long (for example if it has over 50 links inside of it).

Some text..

Some text..

Some text..

Some text..

Some text..

Some text..

Some text..

1.3.3. Giao diện tương thích đa thiết bị

- Grid-View

- Thiết lập box-sizing thành border-box để đảm bảo các thuộc tính padding và border bao gồm tổng chiều rộng và chiều cao của tất cả các phần tử

* { box-sizing: border-box; }

1.3.3. Giao diện tương thích đa thiết bị

■ Grid-View – Tạo các thành phần

```
<div class="row">
  <div class="header">
    <h1>This is a first css layout</h1>
  </div>
  <div class="naviBar col-3">
    <ul>
      /* Đặt các thẻ <a> */
    </ul>
  </div>
  <div class="content col-9">
    /* Đặt một số nội dung vào đây*/
  </div>
</div>
```

1.3.3. Giao diện tương thích đa thiết bị

■ Grid-View – Tạo các thành phần

```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
.row:after{
    clear: both;
    display: table;
    content: "";
}
```

```
[class*="col-"]{
    float: left;
    padding: 15px;
}
```

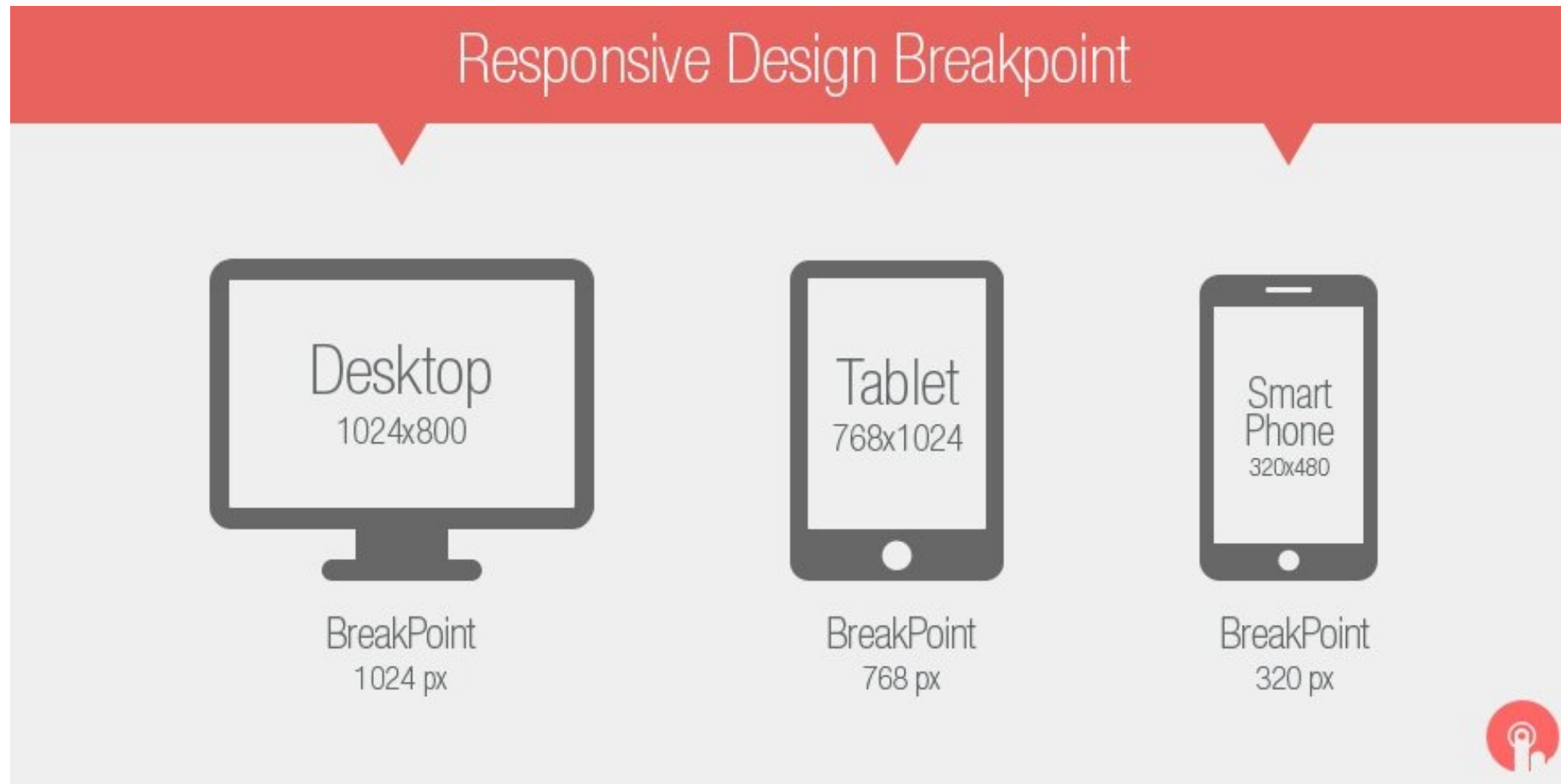
1.3.3. Giao diện tương thích đa thiết bị

- Các breakpoint (điểm dừng):
 - **Breakpoint** mang đến cho layout khả năng thay đổi tại điểm giới hạn
 - Phần lớn các thuộc tính của CSS có thể được thay đổi từ 1 breakpoint này tới một breakpoint khác trên màn hình hiển thị
 - Nơi mà bạn đặt breakpoint thường sẽ phụ thuộc vào nội dung;

```
@media only screen and (max-width: 768px) {  
  /* For mobile phones: */  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```


1.3.3. Giao diện tương thích đa thiết bị

- Các breakpoint (điểm dừng):



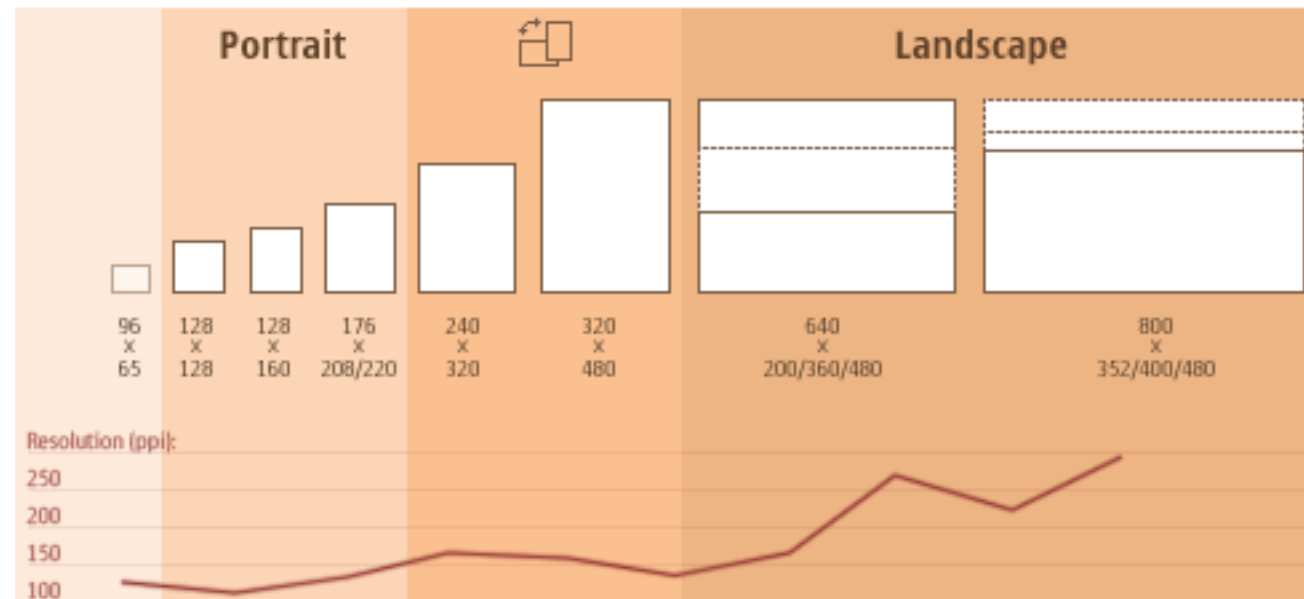
1.3.3. Giao diện tương thích đa thiết bị

- Các breakpoint (điểm dừng):



1.3.3. Giao diện tương thích đa thiết bị

- Các breakpoint (điểm dừng):



1.3.3. Giao diện tương thích đa thiết bị

- Các breakpoint (điểm dừng) dựa trên nội dung

```
/* Extra small devices (phones, 600px and down) */  
@media only screen and (max-width: 600px) {...}
```

```
/* Small devices (portrait tablets and large phones, 600px and up) */  
@media only screen and (min-width: 600px) {...}
```

```
/* Medium devices (landscape tablets, 768px and up) */  
@media only screen and (min-width: 768px) {...}
```

```
/* Large devices (laptops/desktops, 992px and up) */  
@media only screen and (min-width: 992px) {...}
```

```
/* Extra large devices (large laptops and desktops, 1200px and up) */  
@media only screen and (min-width: 1200px) {...}
```

1.3.3. Giao diện tương thích đa thiết bị

- Các breakpoint (điểm dừng) dựa trên thiết bị

```
/* ----- iPhone 6, 6S, 7 and 8 ----- */  
/* Portrait */  
@media only screen  
and (min-device-width: 375px)  
and (max-device-width: 667px)  
and (-webkit-min-device-pixel-ratio: 2)  
and (orientation: portrait) { }  
/* Landscape */  
@media only screen  
and (min-device-width: 375px)  
and (max-device-width: 667px)  
and (-webkit-min-device-pixel-ratio: 2)  
and (orientation: landscape) {  
}
```

1.3.3. Giao diện tương thích đa thiết bị

- Sử dụng các css khác nhau cho từng thiết bị

```
<link rel="stylesheet" media="(max-width: 640px)" href="max-640px.css">
<link rel="stylesheet" media="(min-width: 640px)" href="min-640px.css">
<link rel="stylesheet" media="(orientation: portrait)" href="portrait.css">
<link rel="stylesheet" media="(orientation: landscape)" href="landscape.css">
<style>
  @media (min-width: 500px) and (max-width: 600px) {
    h1 {
      color: fuchsia;
    }

    .desc:after {
      content: " In fact, it's between 500px and 600px wide.";
    }
  }
</style>
```

1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox

- là một kiểu dàn trang tự cân đối kích thước của các phần tử bên trong để hiển thị trên mọi thiết bị
- không cần sử dụng float
- chỉ cần thiết lập để các thành phần bên trong hiển thị theo chiều ngang hay dọc

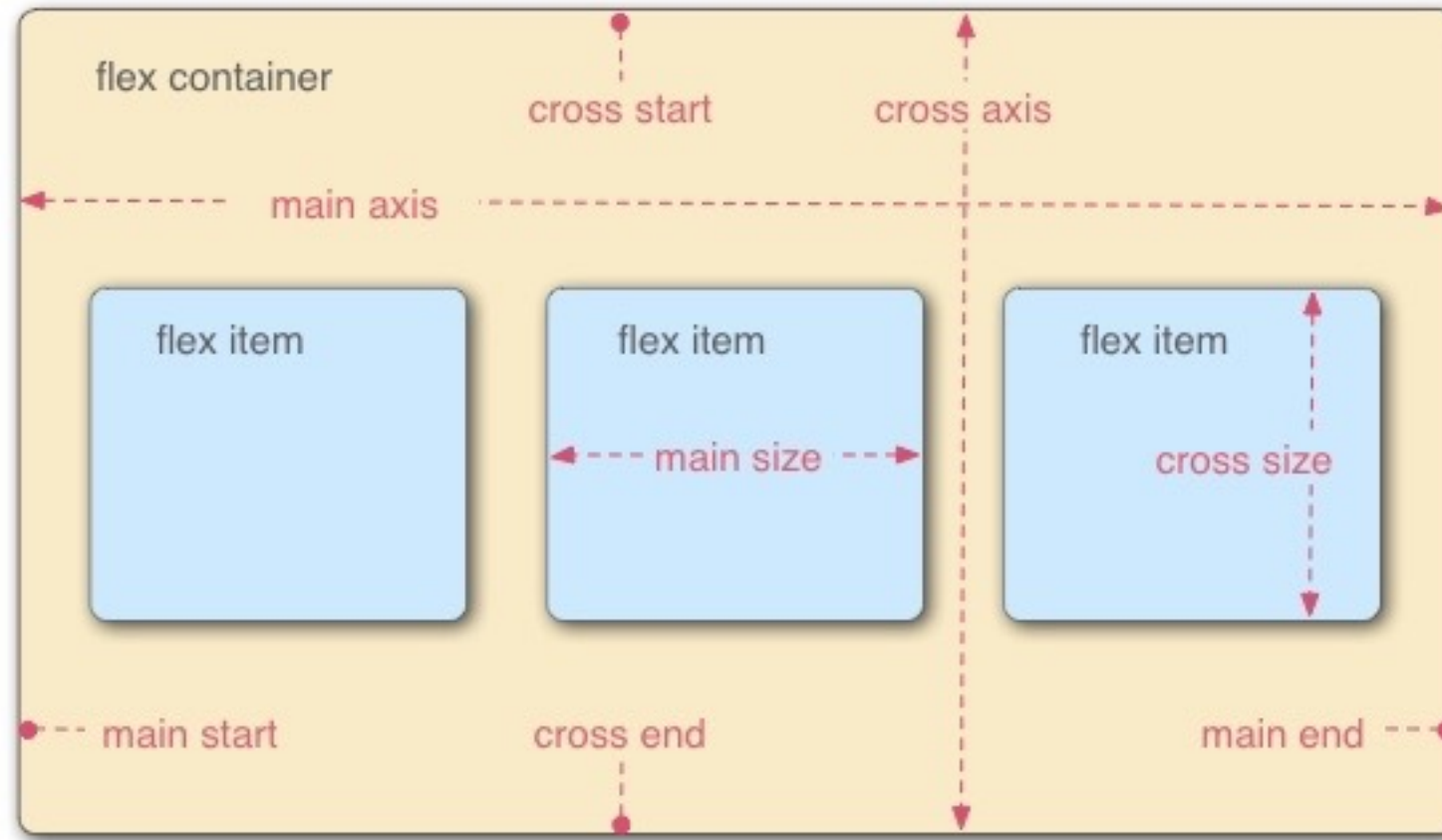
1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox

- Giao diện flexible box là cách thức sử dụng thuộc tính CSS là display
- Các thành phần block xếp theo chiều dọc, các thành phần inline theo chiều ngang (cho đến khi chúng tạo ra một dòng mới)
- Các thành phần trong flexbox có thể đặt theo chiều ngang hoặc chiều dọc
- Bất cứ thành phần nào được tham chiếu là một flexbox sẽ là một thành phần chứa (container)

1.3.3. Giao diện tương thích đa thiết bị

- Flexbox
 - Sơ đồ cấu trúc:



1.3.3. Giao diện tương thích đa thiết bị

- Flexbox – hai thành phần quan trọng nhất trong một bố cục flexbox là
 - container: là thành phần bao quanh các phần tử bên trong
 - item: các phần tử con của container, có thể thiết lập việc nó chiếm bao nhiêu cột trong container hoặc thứ tự hiển thị của nó

1.3.3. Giao diện tương thích đa thiết bị

- Flexbox – các tham số
 - main start, main end: điểm bắt đầu và kết thúc của container
 - main axis: trục chính để điều khiển hướng mà các item sẽ hiển thị
 - main size: kích của mỗi item dựa theo trục cross axis
 - cross size: là kích thước (chiều rộng hoặc cao) của mỗi item dựa theo trục cross axis

1.3.3. Giao diện tương thích đa thiết bị

- Flexbox

- justify-content:

flex-start



flex-end



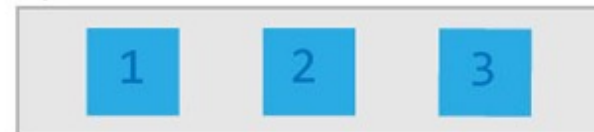
center



space-between



space-around



1.3.3. Giao diện tương thích đa thiết bị

- Flexbox
 - flex-direction: row (mặc định),
 - row-reverse (giống row nhưng ngược lại),
 - column (theo cột),
 - column-reverse,
 - initial (thiết lập mặc định),
 - inherit (kế thừa từ thành phần cha)

1.3.3. Giao diện tương thích đa thiết bị

- Flexbox

- *flex-grow flex-shrink flex basis* | auto | initial | inherit;

Giá trị	Mô tả
<i>flex-grow</i>	Giá trị xác định cách mà một đối tượng mở rộng ra vùng còn lại của các mục khác
<i>flex-shrink</i>	Giá trị xác định cách mà một đối tượng co lại ra vùng còn lại của các mục khác
<i>flex-basis</i>	Kích thước của một đối tượng. Các giá trị hợp lệ: "auto", "inherit", hoặc một giá trị như "%", "px", "em"
auto	Tự động
initial	Giá trị mặc định
none	
inherit	Kế thừa từ đối tượng cha của mục

1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox

```
<div class="container">
  <div class="item item1">1</div>
  <div class="item item2">2</div>
  <div class="item item3">3</div>
  <div class="item item4">4</div>
</div>
```

```
.container {
  background: red;
  max-width: 960px;
  max-height: 1000px;
  margin: 0 auto;
  padding: 5px;
}
.item {
  background: blue;
  margin: 5px;
  color: white;
  height: 50px;
  text-align: center;
  line-height: 50px;
}
```

1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox

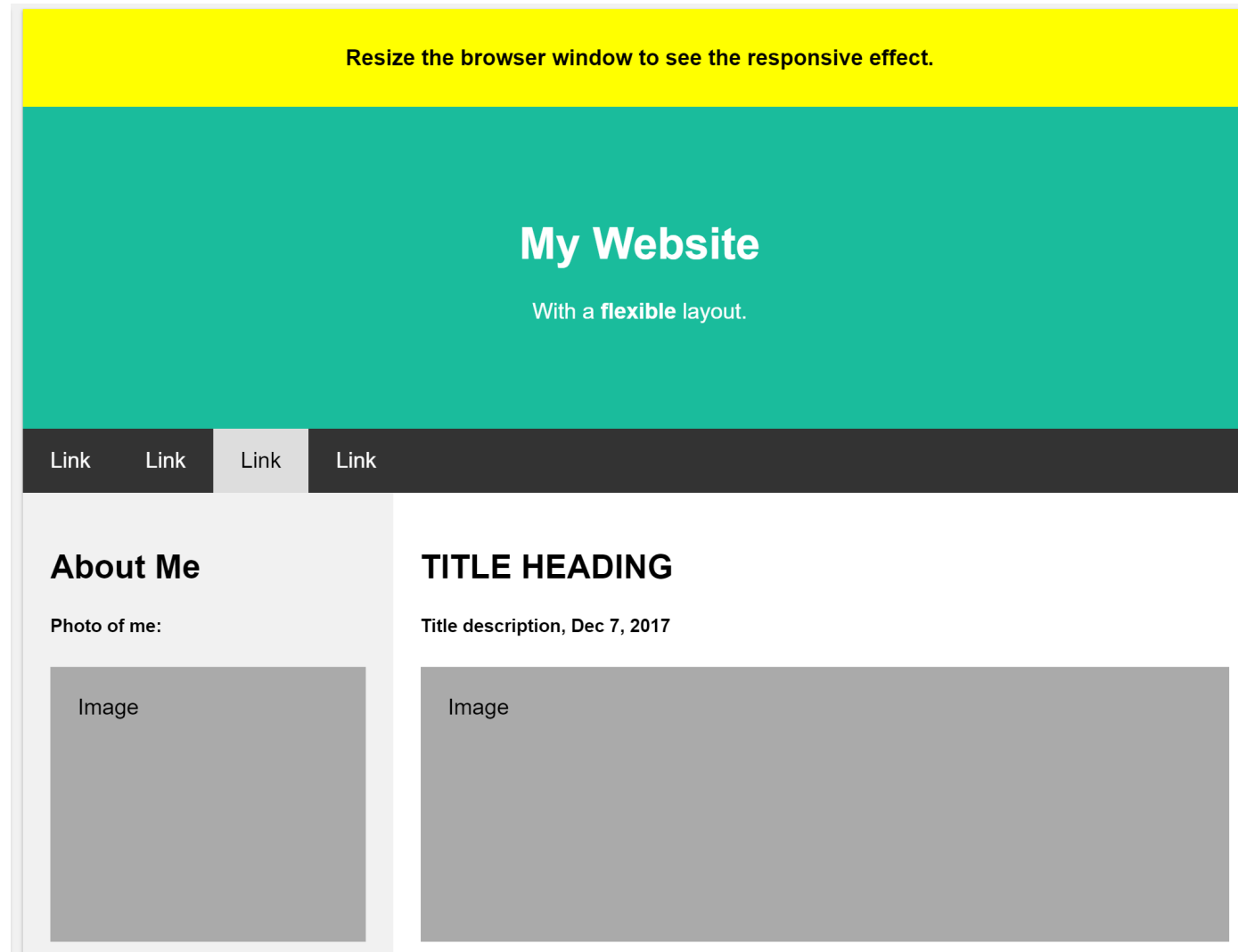
```
<div class="container">
  <div class="item item1">1</div>
  <div class="item item2">2</div>
  <div class="item item3">3</div>
  <div class="item item4">4</div>
</div>
```

```
/** Flex layout */
.container {
  display: flex;
}
```

<https://topdev.vn/blog/su-dung-bo-cuc-trang-flexbox-trong-css/>

1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox



1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox

<!-- Note -->

```
<div style="background:yellow;padding:5px">
```

```
  <h4 style="text-align:center">Resize the browser window to see the responsive effect.</h4>
```

```
</div>
```

```
  <div class="header">
```

```
    <h1>My Website</h1>
```

```
    <p>With a <b>flexible</b> layout.</p>
```

```
  </div>
```

```
<!-- Navigation Bar -->
```

```
<div class="navbar">
```

```
  <a href="#">Link</a>
```

```
  <a href="#">Link</a>
```

```
  <a href="#">Link</a>
```

```
  <a href="#">Link</a>
```

```
</div>
```

1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox

```
<div class="row">
  <div class="side">
    <h2>About Me</h2>
    <h5>Photo of me:</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
    <h3>More Text</h3>
    <p>Lorem ipsum dolor sit ame.</p>
    <div class="fakeimg" style="height:60px;">Image</div><br>
    <div class="fakeimg" style="height:60px;">Image</div><br>
    <div class="fakeimg" style="height:60px;">Image</div>
  </div>
  <div class="main">
    <h2>TITLE HEADING</h2>
    <h5>Title description, Dec 7, 2017</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text..</p>
    <p>A</p>
    <br>
    <h2>TITLE HEADING</h2>
    <h5>Title description, Sep 2, 2017</h5>
    <div class="fakeimg" style="height:200px;">Image</div>
    <p>Some text..</p>
    <p>A.</p>
  </div>
</div>
```

```
<!-- Footer -->
<div class="footer">
  <h2>Footer</h2>
</div>
```

1.3.3. Giao diện tương thích đa thiết bị

■ Flexbox

```
.navbar {  
  display: flex;  
  background-color: #333;  
}
```

```
.row {  
  display: flex;  
  flex-wrap: wrap;  
}
```

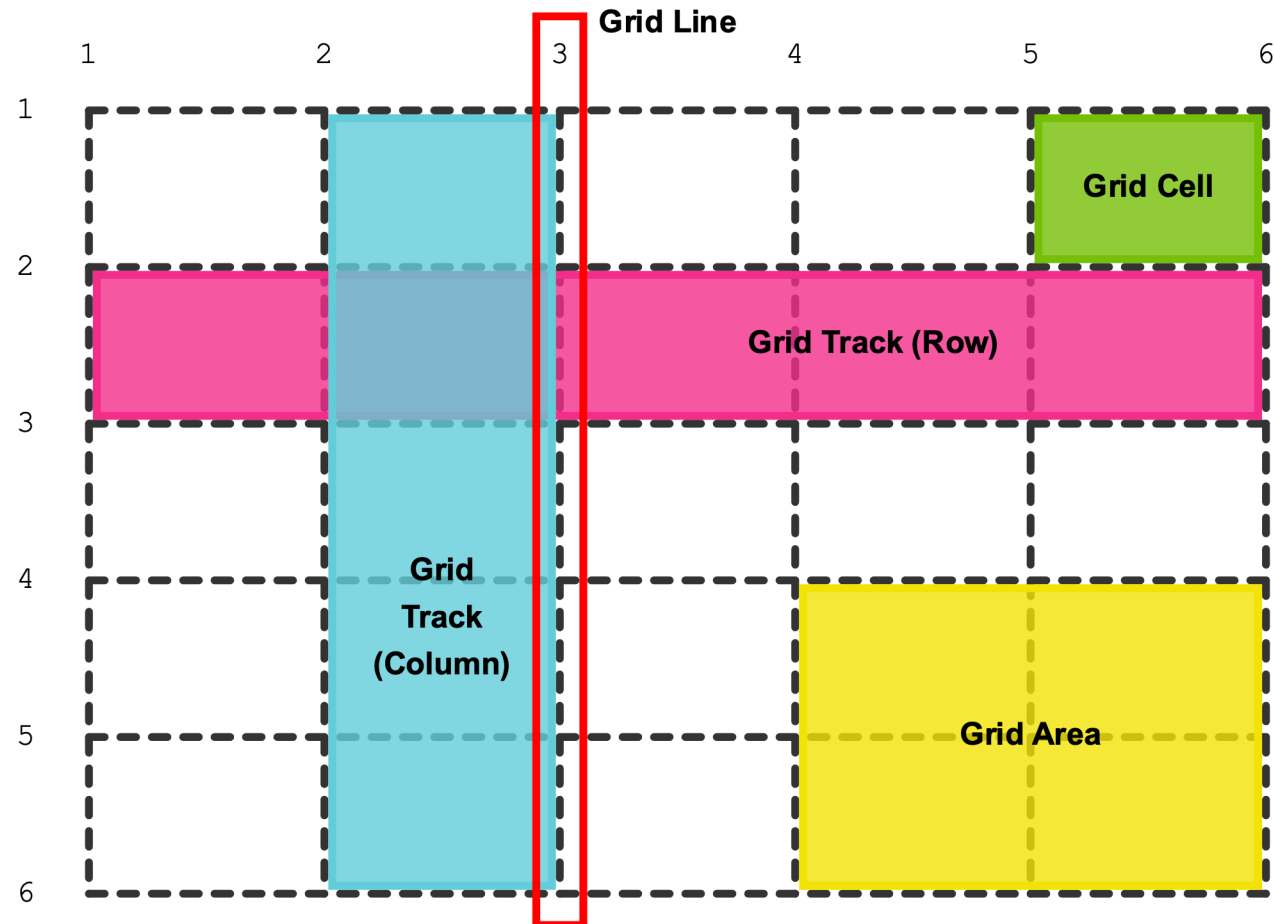
```
.side {  
  flex: 30%;  
  background-color: #f1f1f1;  
  padding: 20px;  
}
```

```
.main {  
  flex: 70%;  
  background-color: white;  
  padding: 20px;  
}
```

```
@media screen and (max-width: 700px) {  
  .row, .navbar {  
    flex-direction: column;  
  }  
}
```

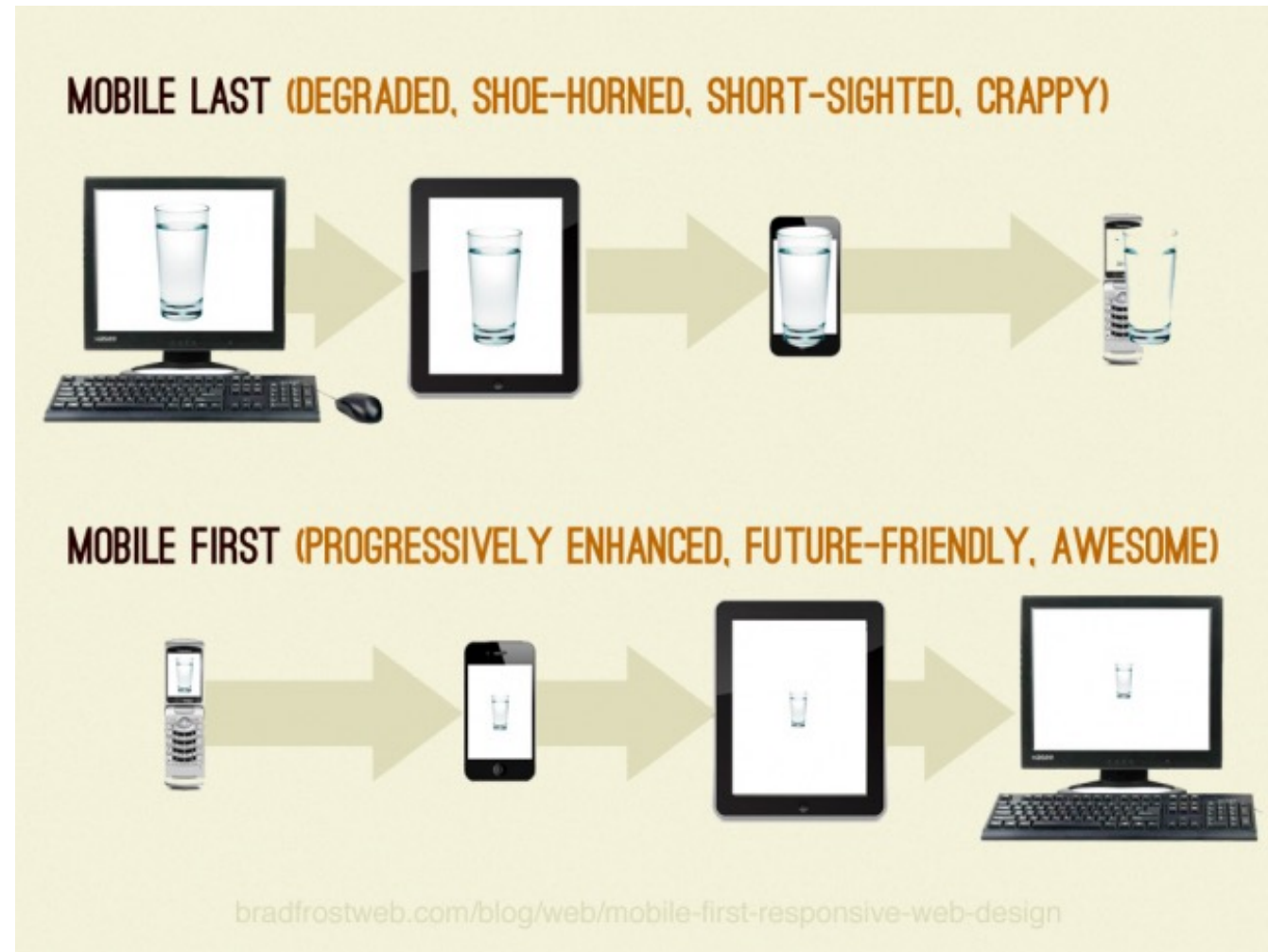
1.3.3. Giao diện tương thích đa thiết bị

- CSS Grid Layout

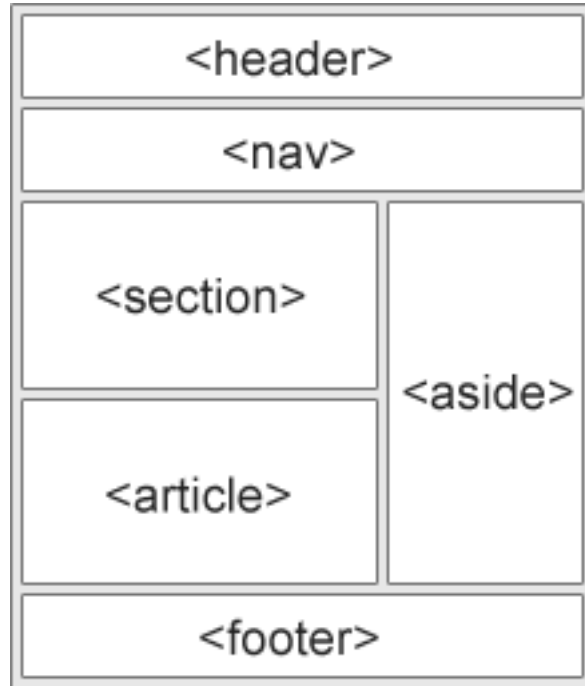


1.3.3. Giao diện tương thích đa thiết bị

- Khái niệm Mobile-First



1.3.4. HTML5 - Layout



- <header> - Định nghĩa header cho một tài liệu hoặc một vùng (section)
- <nav> - định nghĩa vùng chứa các navi links
- <section> - định nghĩa một section trong Doc
- <article> - định nghĩa một vùng độc lập chứa nội dung
- <aside> - định nghĩa một vùng nội dung (như side bar)
- <footer> - định nghĩa vùng footer
- <details> - định vùng thông tin bổ sung
- <summary> - định nghĩa tiêu đề cho thành phần <details>

