

# HTML Cơ bản

# Nội dung

1. HTML là gì
2. Môi trường phát triển HTML
3. Tự tạo file HTML đơn giản
4. Ngôn ngữ thẻ - các thẻ cơ bản

HTML là gì?

# HTML là gì ?

- HTML viết tắt của : **H**yper**T**ext **M**arkup **L**anguage
- Là ngôn ngữ đánh dấu
- Dùng để thể hiện nội dung hiển thị trên website ( là chương trình chạy trên trình duyệt - browser)
- Sử dụng các thẻ và các phân tử HTML, ta có thể:
  - Điều khiển hình thức và nội dung của trang
  - Xuất bản các tài liệu trực tuyến và truy xuất thông tin trực tuyến bằng cách chèn các liên kết vào tài liệu HTML
  - Tạo các biểu mẫu (form) để thu thập thông tin về người dùng, quản lý giao dịch,...

# HTML là gì



# HTML là gì

facebook

Email hoặc điện thoạiMật khẩu

Đăng nhập

Quên tài khoản?

Đăng ký

Luôn miễn phí.

HọTên

Số di động hoặc email

Mật khẩu mới

Ngày sinh

29Tháng 61993

Tại sao tôi cần cung cấp ngày sinh của mình?

Nữ

Nam

Billing cách nhập vào Đăng ký, bạn đồng ý với Điều khoản, Chính sách dữ liệu và Chính sách cookie của chúng tôi. Bạn có thể nhận được thông báo của chúng tôi qua SMS và hủy nhận bất kỳ lúc nào.

Đăng ký

Tạo Trang dành cho người nổi tiếng, nhãn hiệu hoặc doanh nghiệp.

```
1 <!DOCTYPE html>
2 <html lang="vi" id="facebook" class="no_js">
3 <head><meta charset="utf-8" /><meta name="referrer" content="origin-when-crossorigin" id="meta_referrer" />
4 <script>window._start=new Date();</script><script>script.env.flush(a)(function(b){for(var c in a)b[c]=a[c]});window.requireLazy?
5 window.requireLazy(["Env"],b):(window.Env=window.Env||
6 {}),b(window.Env).env.flush({ajaxpipe_token:"AXI3I204Zfeyj1Ct","timeslice_heartbeat_config":
7 {"pollIntervalMs":33,"idleCapThresholdMs":60,"ignoredTimesliceNames":["requestAnimationFrame",true,"Event listener handler
8 mouseover",true,"Event listener handler mouseover",true,"Event listener handler mouseout",true,"Event listener handler
9 scroll",true},"isHeartbeatEnabled",true,"isArtilleryOn",false},"shouldLogCounters",true,"timeslice_categories":
10 {"react_render",true,"reflow",true},"sample_continuation_stacktraces",true,"dom_mutation_flag",true,"kshh":"0 sj" e" xm" s-
11 ofdu" ghdoer-ggc" eurf-3gc" eurf;|enbtdou;fdumddourcx0_id-2YLMtugSgptdru;ggnunuxqd;rdoe-dunjdjnk-bunjdjnk0-gdublrdbadobv-
12 0 sj" e" z-qg xm" r-0t0t0ba qhof-0mbo"q xm";,"stack_trace_limit":30,"deferred_stack_trace_rate":1000,"timeslicebufferSize":5000});
13 </script><style><script>_DEV_ =0;CavalryLogger=window.CavalryLogger||function(a)
14 (this.lid=a,this.transition=1,this.metric_collected=1,this.is_detailed_profiler=1,this.instrumentation_started=1,this.pagelet_m
15 etrics={},this.events={},this.ongoing_watch={},this.values={t_start:window._start},this.piggy_values={},this.bootloader_metrics=
16 {}),this.resource_to_pagelet_mapping=
17 {}),this.e2eLogged=1,this.initializeInstrumentation&this.initializeInstrumentation(),CavalryLogger.prototype.setIsDetailedProfile
18 r=function(a){this.is_detailed_profiler=a;return this},CavalryLogger.prototype.setTTIEvent=function(a){this.tti_event=a;return
19 this},CavalryLogger.prototype.setValue=function(a,b,c,d){d&#x2192;this.piggy_values[this.values];(typeof d[a])=="undefined"?[c]&#x2192;
20 d[a]=b;return this},CavalryLogger.prototype.getLastTTIValue=function()
21 (return this.lastTTIValue),CavalryLogger.prototype.setTimestamp=CavalryLogger.prototype.setTimestamp|function(a,b,c,d){this.mark(a);var
22 e=this.values,t_start|[this.values.t_start=ed&#x2192;CavalryLogger.now|];this.setValue(a,b,c);this.tti_event&#x2192;this.tti_event&#x2192;
23 (this.lastTTIValue&#x2192;this.setTimestamp("t_tti",b));return this},CavalryLogger.prototype.mark=typeof
24 console=="object"&#x2192;console.timestamp(function(a){console.timestamp(a)});function()
25 {}),CavalryLogger.prototype.addPiggyback=function(a,b){this.piggy_values[a]=b;return this},CavalryLogger.instances=
26 {}),CavalryLogger.prototype.perfNubMarkup=CavalryLogger.disableArtilleryOnTTIOfLogging=1,CavalryLogger.getInstance&#x2192;
27 function(a){(typeof a=="undefined"&#x2192;(a=CavalryLogger).prototype.instances[a])||(CavalryLogger.instances[a]=new
28 CavalryLogger(a));return CavalryLogger.instances[a]},CavalryLogger.setPageID=function(a){if(CavalryLogger.id==0){var
29 b=CavalryLogger.now();CavalryLogger.instances[a]=CavalryLogger.instances[a];CavalryLogger.instances[a]=a;delete
30 CavalryLogger.instances[0];CavalryLogger.id=a,CavalryLogger.setPerfNubMarkup=function(a,b)
31 (CavalryLogger.perfNubMarkup=a),CavalryLogger.now=function()
32 (return window.performance&#x2192;performance.timing&#x2192;performance.timing.navigationStart&#x2192;performance.now?
33 performance.now|performance.timing&#x2192;performance.timing.navigationStart&#x2192;new Date().getTime(),CavalryLogger.prototype.measureResources=function()
34 {}),CavalryLogger.prototype.profileEarlyResources=function()
35 {}),CavalryLogger.prototype.start_js=function()
36 {}),CavalryLogger.done_js=function()
37 {}),CavalryLogger.getTTIEvent("t_domcontent");CavalryLogger.prototype.measureResources=function(a,b)
38 (if(this.log_resources)return var
39 c="bootload/"+a.name;if(this.bootloader_metrics[c]==undefined|this.ongoing_watch[c]==undefined){return;var
40 d=CavalryLogger.now();this.ongoing_watch[c]=d;start_c="c in this.bootloader_metrics|[this.bootloader_metrics["start_c"]=d];b&#x2192;
41 ("tag_c" in this.bootloader_metrics)&#x2192;(this.bootloader_metrics["tag_c"]=b);if(a.type=="js")
42 (c="js_exec/"+a.name,this.ongoing_watch[c]=d),CavalryLogger.prototype.stopWatch=function(a){if(this.ongoing_watch[a]){var
43 d=t_resource.download|[d.t_resource.download=0],d.resources.download=1,d["tag_"+a]="_EF_&#x2192;
44 (d.resources.download=0),d.t_resource.download=c,d.resources.download=1,d["tag_"+a]="_EF_&#x2192;
45 (d.pagelet_cssload_early_resources=b);delete this.ongoing_watch[a];return
46 this},CavalryLogger.getBootloaderMetricsFromAllLoggers=function()
47 (var a=
48 {}),Object.values(window.CavalryLogger.instances).forEach(function(b)
49 (b.bootloader_metrics&#x2192;Object.assign(a,b.bootloader_metrics));return a},CavalryLogger.start_js=function(a){for(var
50 b=0;b<a.length;b++)CavalryLogger.getInstance().stopWatch("js_exec/"+a[b]),CavalryLogger.done_js=function(a){for(var
51 a){for(var b=0;b<a.length;b++)this.measureResources({name:a[b][0],type:a[b]
52 [1]});js="";_EF_);CavalryLogger.getInstance().log_resource=true;CavalryLogger.getInstance().setIsDetailedProfiler(true);window
53 .CavalryLogger&#x2192;CavalryLogger.getInstance().setTimestamp("t_start");</script><noscript><meta http-equiv="refresh" content="0";
```

# HTML là gì

- HTML phiên bản đầu tiên chính thức ra đời là bản HTML 4.01 năm 1999
- Phiên bản tiếp theo XHTML
- Phiên bản hiện nay HTML5
- Website giới thiệu HTML và cú pháp cơ bản

<https://www.w3schools.com/>



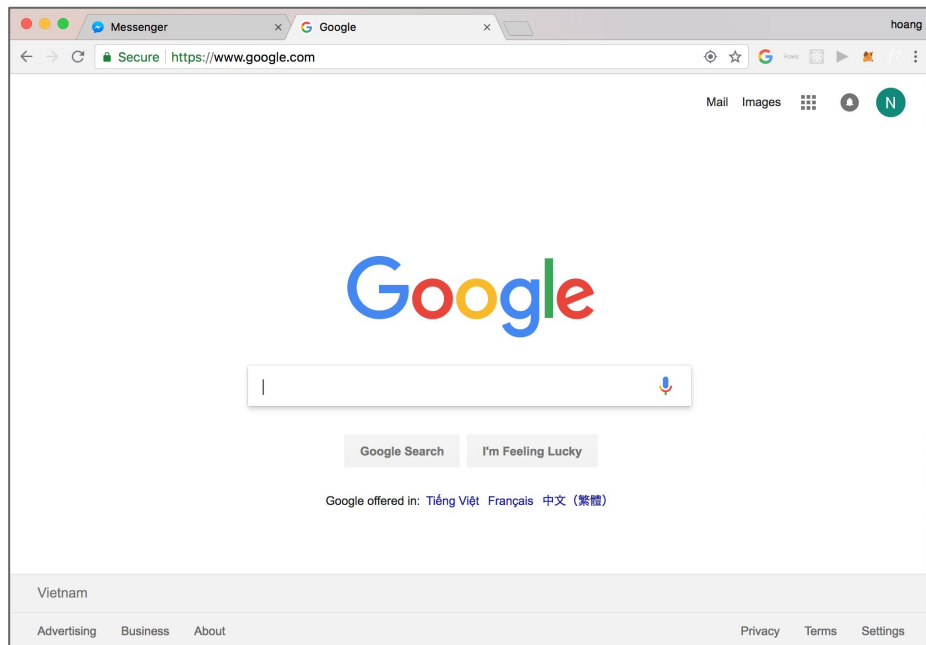
Môi trường phát triển HTML



# Môi trường phát triển

## Trình duyệt - Browser

- Là môi trường để chạy các file HTML
- Tất cả các trình duyệt hiện tại đều hỗ trợ hiển thị nội dung HTML
- Tuy nhiên mỗi trình duyệt khác nhau có thể có cách hiển thị khác nhau, không hoàn toàn giống nhau 1 cách tuyệt đối



# Môi trường phát triển

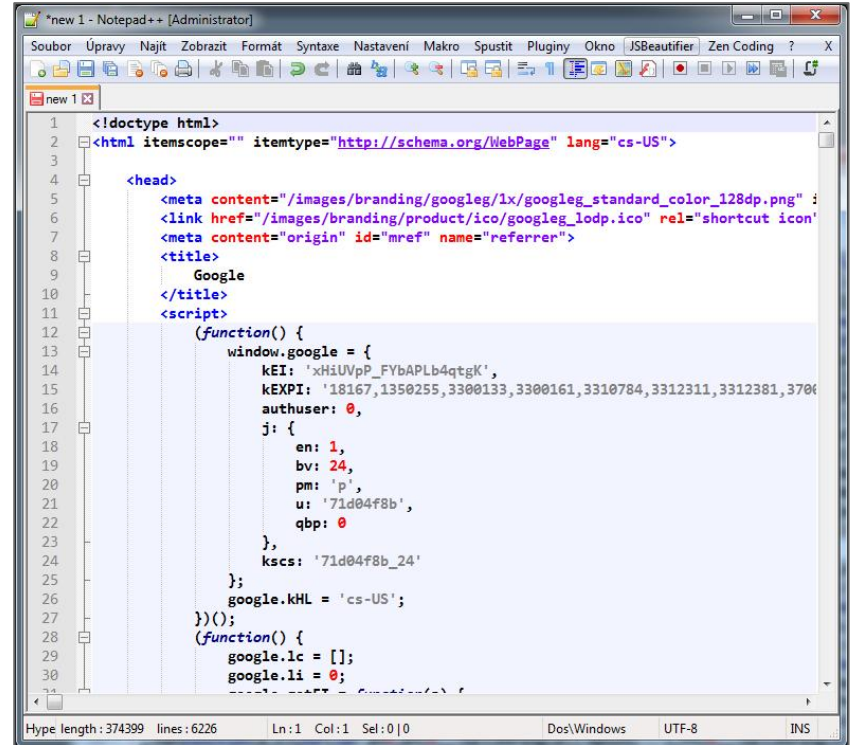
## Các trình duyệt phổ biến



# Môi trường phát triển

## Trình soạn thảo - IDE

- Là ứng dụng để soạn thảo mã HTML
- Có thể sử dụng bất cứ chương trình soạn thảo nào
- Ví dụ: Notepad, Notepad++, Sublime Text, Netbean,...
- Nên sử dụng phần mềm chuyên dụng để được hỗ trợ tối đa trong việc viết các mã HTML.



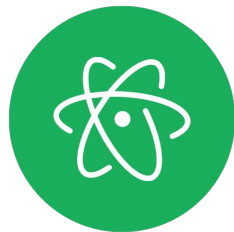
The screenshot shows a Notepad++ window titled '\*new 1 - Notepad++ [Administrator]'. The menu bar includes 'Soubor', 'Úpravy', 'Najít', 'Zobrazit', 'Formát', 'Syntaxe', 'Nastavení', 'Makro', 'Spustit', 'Pluginy', 'Okno', 'JSBeautifier', 'Zen Coding', '?', and 'X'. The toolbar contains various icons for file operations and editing. The main text area displays HTML code for a Google search page. The code includes a DOCTYPE declaration, an HTML tag with attributes, a head section with meta tags for branding and a link to the Google logo, a title 'Google', and a script block containing a function to initialize the Google search interface. The status bar at the bottom shows 'Byte length: 374399', 'lines: 6226', 'Ln: 1', 'Col: 1', 'Sel: 0 | 0', 'Dos/Windows', 'UTF-8', and 'INS'.

```
1 <!doctype html>
2 <html itemscope="" itemtype="http://schema.org/WebPage" lang="cs-US">
3
4 <head>
5   <meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" :
6   <link href="/images/branding/product/ico/googleg_lodp.ico" rel="shortcut icon"
7   <meta content="origin id="mref" name="referrer">
8   <title>
9     Google
10  </title>
11  <script>
12    (function() {
13      window.google = {
14        KEI: 'xHiUVpP_FYbAPLb4qtgK',
15        kEXPI: '18167,1350255,3300133,3300161,3310784,3312311,3312381,370
16        authuser: 0,
17        j: {
18          en: 1,
19          bv: 24,
20          pm: 'p',
21          u: '71d04f8b',
22          qbp: 0
23        },
24        kscs: '71d04f8b_24'
25      };
26      google.kHL = 'cs-US';
27    })();
28    (function() {
29      google.lc = [];
30      google.li = 0;
```

# Môi trường phát triển

## Một số Trình soạn thảo HTML chuyên dụng

- Netbean
- Notepad++
- Sublime Text
- Visual Studio Code
- Atom
- Webstorm

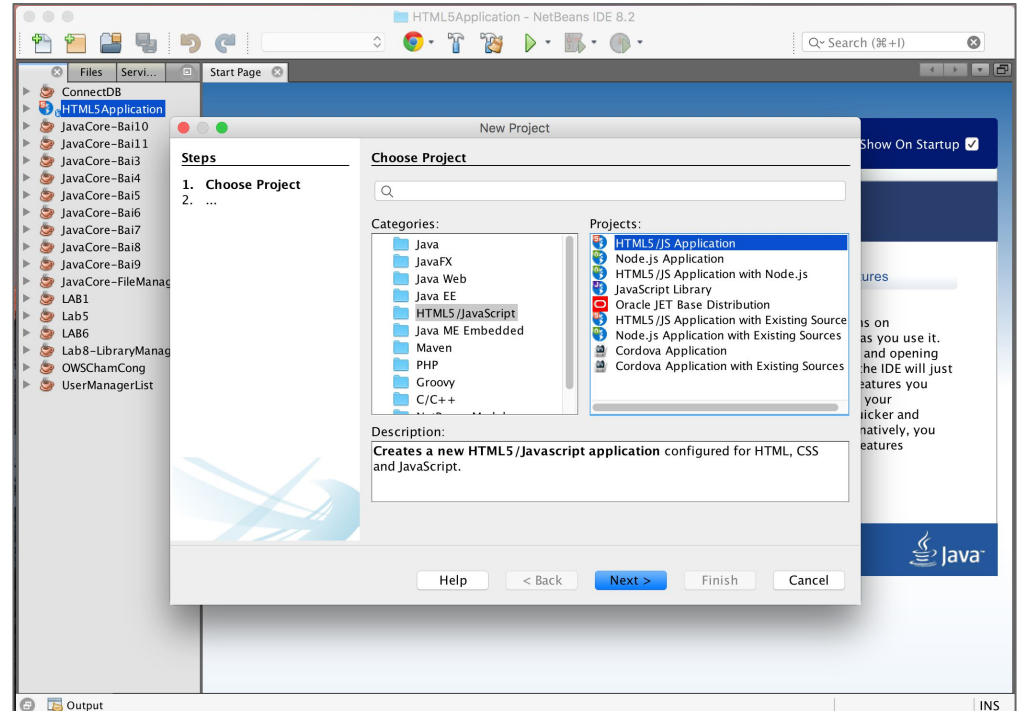


Tạo HTML đơn giản

# Tạo HTML đơn giản

## Mở trình soạn thảo - IDE

- Tạo project HTML
- Tạo file mới lưu với tên ``index.html``
- Gõ lại nội dung như sau

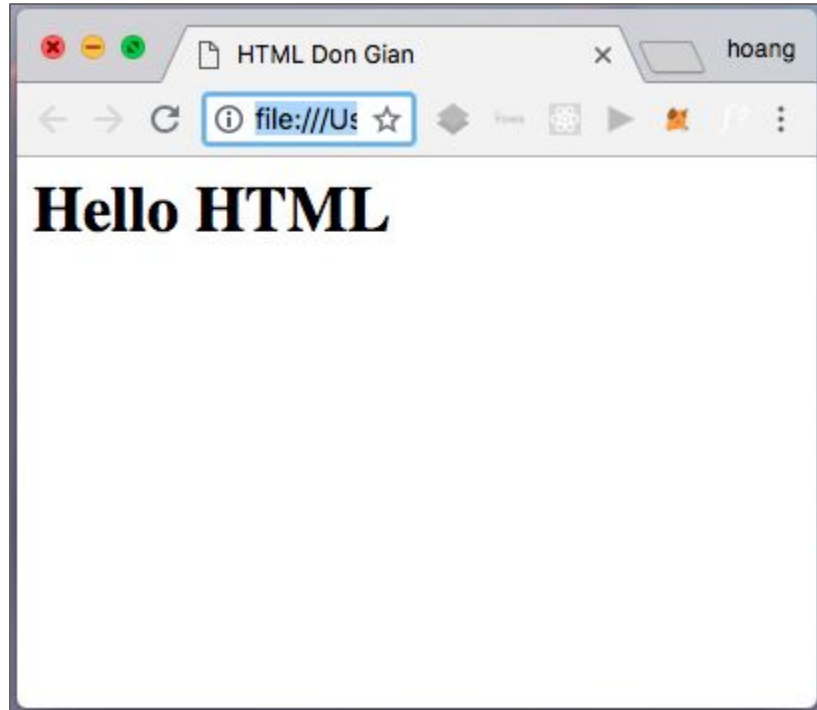


# Tạo HTML đơn giản

```
<html>  
  <head>  
    <title>HTML Don Gian</title>  
  </head>  
  <body>  
    <h1> Hello HTML </h1>  
  </body>  
</html>
```

# Tạo HTML đơn giản

**Mở file vừa tạo bằng trình duyệt**





# Tạo HTML đơn giản

## Phân tích cấu tạo 1 file `index.html`

- Nội dung được tạo bởi các thẻ
- Bắt đầu bằng cặp thẻ `<html> ... </html>`

```
<html>  
...  
</html>
```

# Tạo HTML đơn giản

## Phân tích cấu tạo 1 file `index.html`

- Nội dung chứa trong 2 thẻ chính  
`<head>` và thẻ `<body>`
- Thẻ `<head>` định nghĩa nội dung cho phần đầu - tiêu đề của trang
- Thẻ `<body>` định nghĩa nội dung cho phần thân của trang

```
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

# Ngôn ngữ thẻ và Các thẻ cơ bản trong HTML

# Ngôn ngữ thẻ

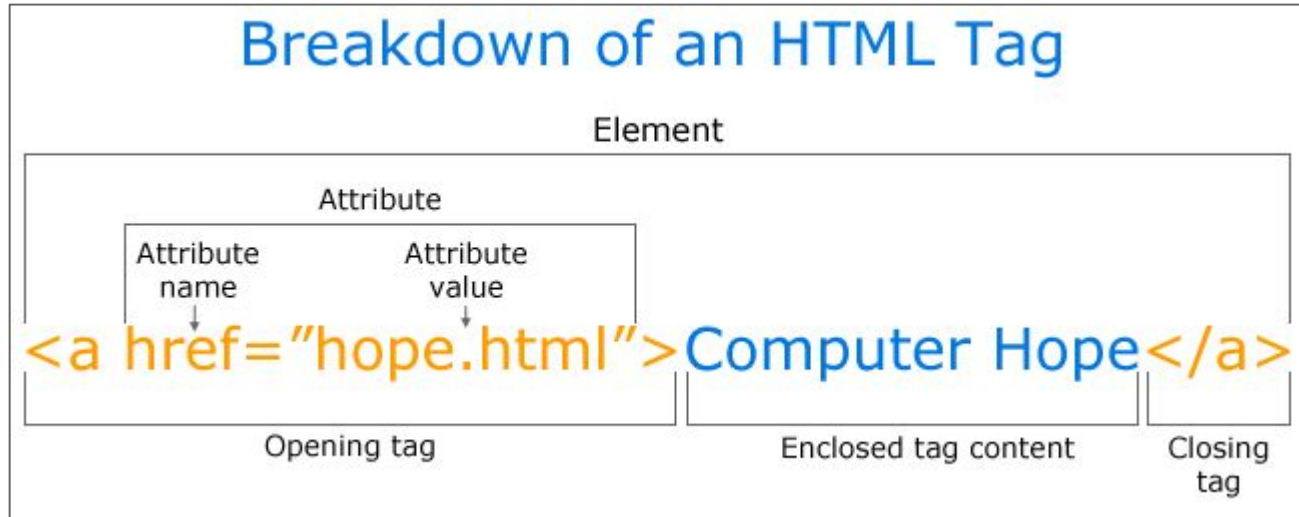
## Ngôn ngữ thẻ - Ngôn ngữ đánh dấu

- Là 1 dạng ngôn ngữ được sử dụng phổ biến
- Ngoài HTML còn có: XML, SGML, LaTeX,...
- Sử dụng các thẻ (**Tag**) để đánh dấu nội dung trong văn bản

# Ngôn ngữ thẻ

## Cấu tạo của 1 thẻ (Tag)

- PAIR



# Ngôn ngữ thẻ

## Cấu tạo của 1 thẻ (Tag)

- UNPAIR

```

```

# Các thẻ cơ bản

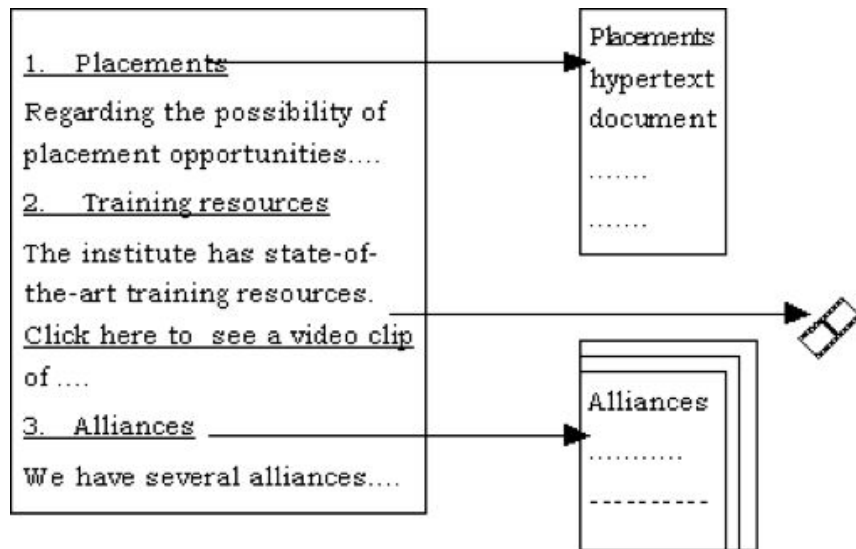
## **Các thẻ cơ bản trong HTML**

- Thẻ liên kết
- Thẻ tiêu đề
- Thẻ cấu trúc - khối
- Thẻ định dạng kí tự
- Danh sách
- Thẻ Đường kẻ ngang
- Thẻ Font
- Thẻ Hình ảnh
- Thẻ bảng

# Các thẻ cơ bản - Thẻ liên kết

## Các loại siêu liên kết

- **Liên kết trong** là liên kết đến các thành phần trong cùng tài liệu hoặc cùng một website
- **Liên kết ngoài** là liên kết đến các trang trên các website khác hoặc máy chủ khác.





# Các thẻ cơ bản - Thẻ liên kết

- Để tạo liên kết chúng ta cần phải xác định hai thành phần
  - Địa chỉ đầy đủ hoặc URL của file được kết nối
  - Điểm neo (anchor) cung cấp cho liên kết. Điểm này có thể là một dòng văn bản hoặc thậm chí là một ảnh.
- Các kiểu URL
  - URL tuyệt đối (vd: **<http://demo.website.com/index.html>**)
  - URL tương đối (vd: **</image/demo.png>**)

# Các thẻ cơ bản - Thẻ liên kết

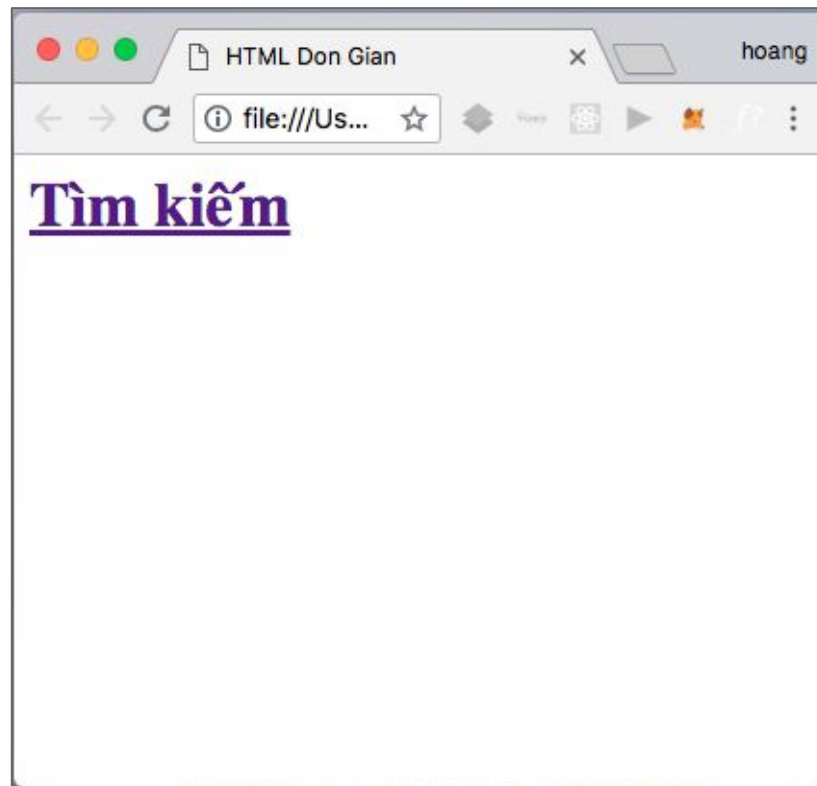
- Trong HTML ta có thể sử dụng thẻ `<a>` để tạo siêu liên kết
- Cú pháp sử dụng thẻ

```
<a href=" url "> anchor </a>
```

- Ví dụ

```
<a href="http://www.google.com"> Tìm kiếm </a>
```

# Các thẻ cơ bản - Thẻ liên kết



# Các thẻ cơ bản - Thẻ tiêu đề

- Để tạo các tiêu đề (header) với các kích cỡ khác nhau
- HTML có các thẻ tiêu đề từ **<h1>** - **<h6>**
- Cú pháp sử dụng thẻ và ví dụ

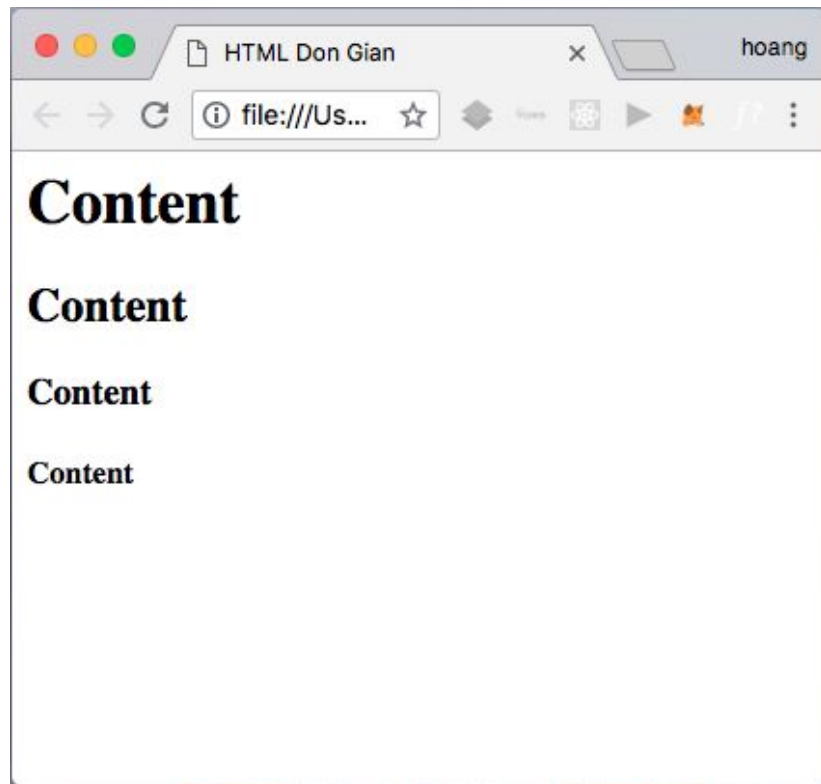
```
<h1> Content </h1>
```

```
<h2> Content </h2>
```

```
<h3> Content </h3>
```

```
<h4> Content </h4>
```

# Các thẻ cơ bản - Thẻ tiêu đề



# Các thẻ cơ bản - Thẻ cấu trúc

- Thẻ trích dẫn <blockquote>
- Thẻ dữ nguyên định dạng văn bản <pre>
- Thẻ đoạn <p>

# Các thẻ cơ bản - Thẻ cấu trúc

**<p>**The blockquote element is used to format the content in blocks of text.**</p>**

**<blockquote>**

Humpty Dumpty sat on a wall

Humpty Dumpty had a great fall

All the King's horses

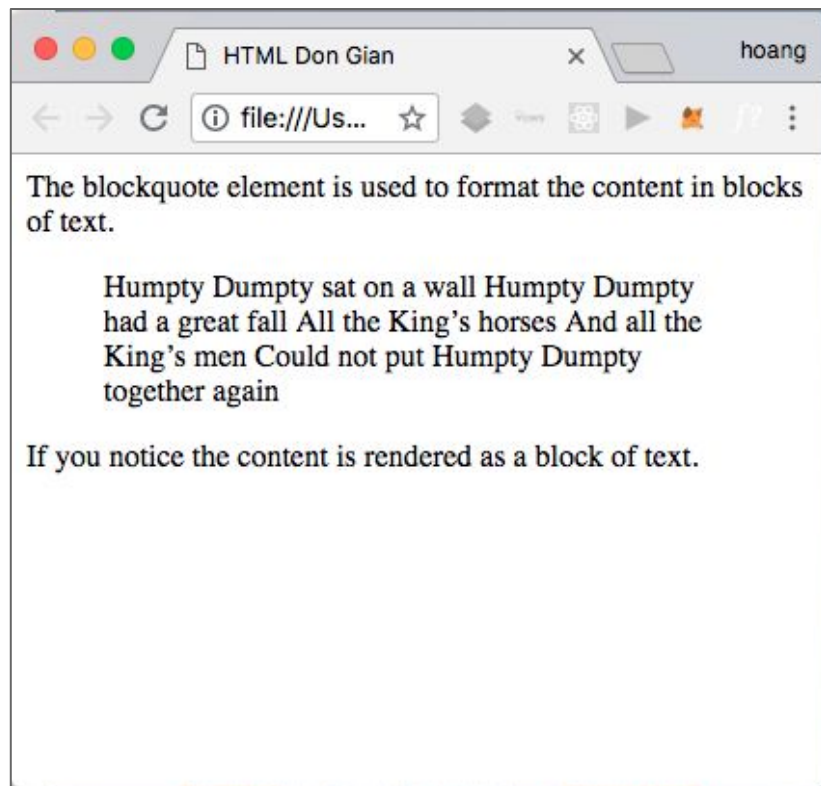
And all the King's men

Could not put Humpty Dumpty together again

**</blockquote>**

**<p>**If you notice the content is rendered as a block of text.**</p>**

# Các thẻ cơ bản - Thẻ tiêu đề





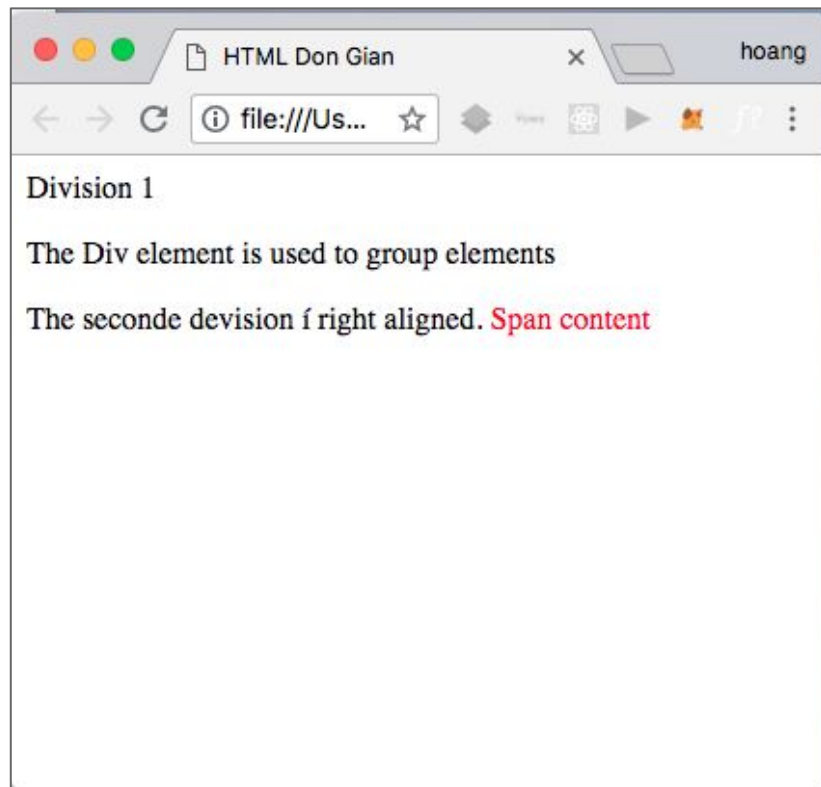
# Các thẻ cơ bản - Thẻ khối

- Thẻ khối trên dòng (in-line) **<span>**
- Thẻ khối **<div>**

# Các thẻ cơ bản - Thẻ khối

```
<div>
  Division 1
  <p>The Div element is used to group elements</p>
</div>
<p> The seconde devision í right aligned.
<span style="color: red">Span content</span>
</p>
```

# Các thẻ cơ bản - Thẻ khối



# Các thẻ cơ bản - Thẻ định dạng kí tự

- Thẻ in đậm **<b>**
- Thẻ chỉ số trên **<sup>**
- Thẻ chỉ số dưới **<sub>**

# Các thẻ cơ bản - Thẻ định dạng kí tự

```
<b>This is good fun</b>
```

```
<p>
```

```
    pi * r <sup>2</sup>
```

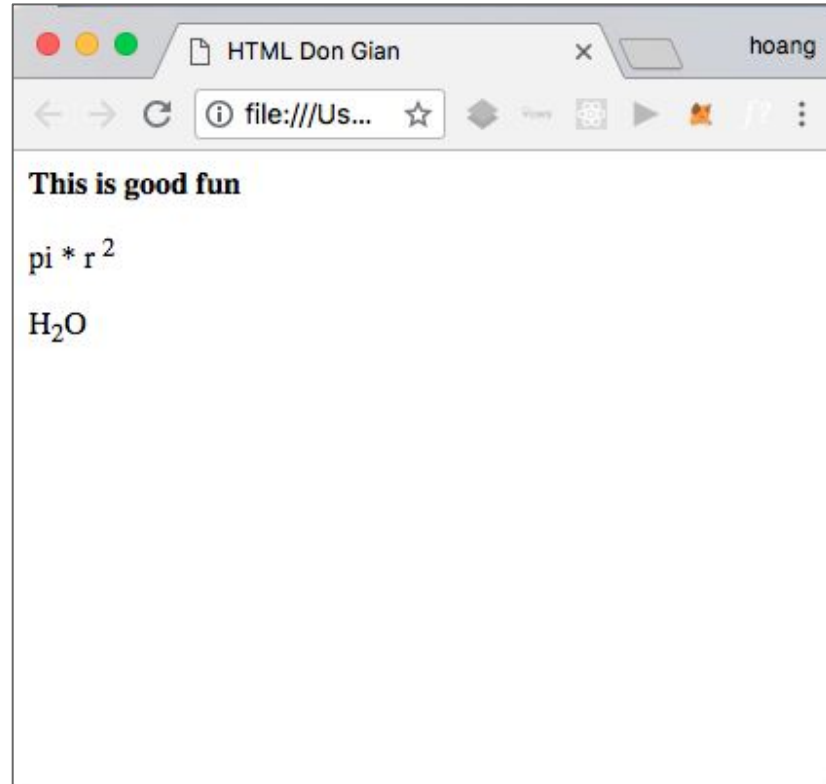
```
</p>
```

```
<p>
```

```
    H<sub>2</sub>O
```

```
</p>
```

# Các thẻ cơ bản - Thẻ định dạng kí tự



# Các thẻ cơ bản - Thẻ danh sách

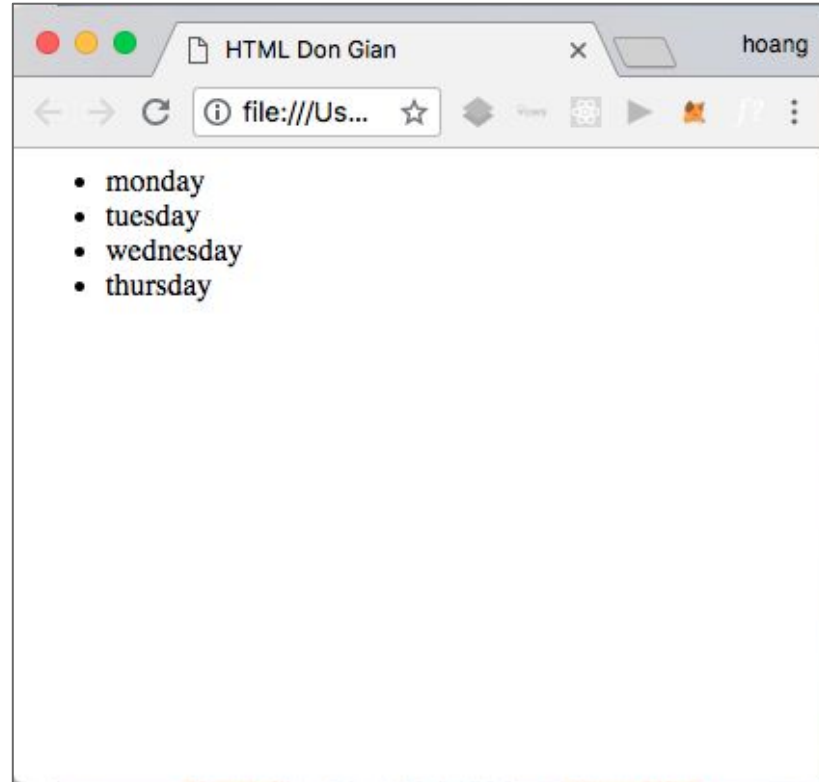
- Danh sách không thứ tự `<ul>` `<li>`
- Danh sách có thứ tự `<ol>` `<li>`
- Danh sách định nghĩa `<dl>` `<dt>` `<dd>`

# Các thẻ cơ bản - Thẻ danh sách

```
<ul>  
  <li> monday </li>  
  <li> tuesday </li>  
  <li> wednesday </li>  
  <li> thursday </li>  
</ul>
```



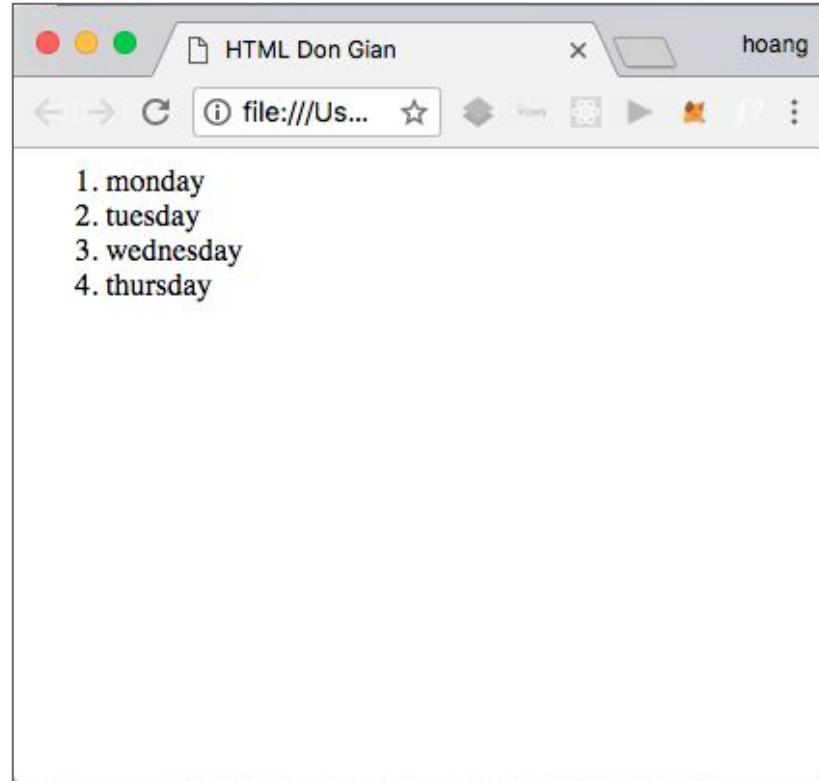
# Các thẻ cơ bản - Thẻ danh sách



# Các thẻ cơ bản - Thẻ danh sách

```
<ol>  
  <li> monday </li>  
  <li> tuesday </li>  
  <li> wednesday </li>  
  <li> thursday </li>  
</ol>
```

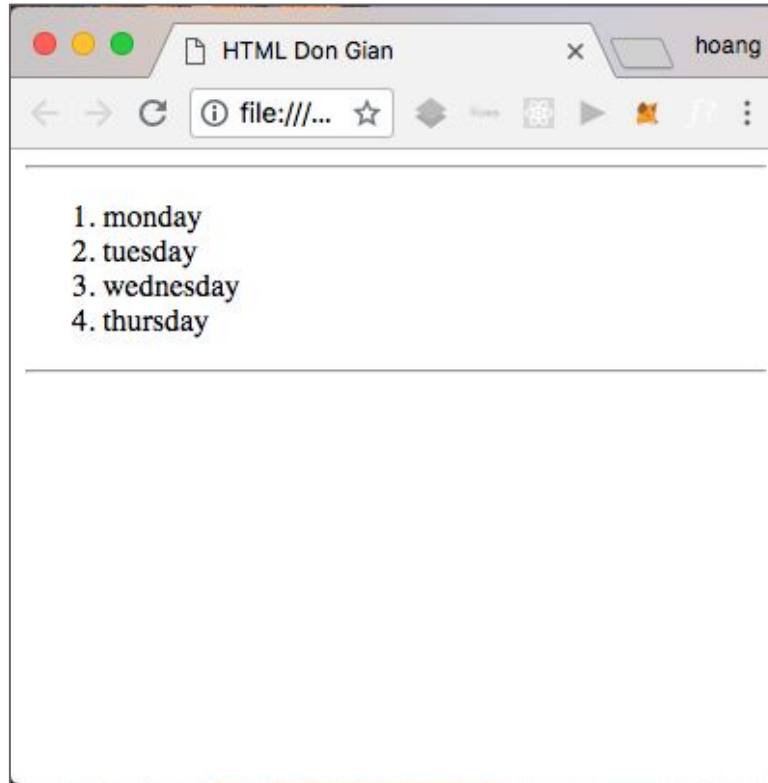
# Các thẻ cơ bản - Thẻ danh sách



# Các thẻ cơ bản - Thẻ đường kẻ ngang

```
<hr/>  
<ol>  
  <li> monday </li>  
  <li> tuesday </li>  
  <li> wednesday </li>  
  <li> thursday </li>  
</ol>
```

# Các thẻ cơ bản - Thẻ đường kẻ ngang



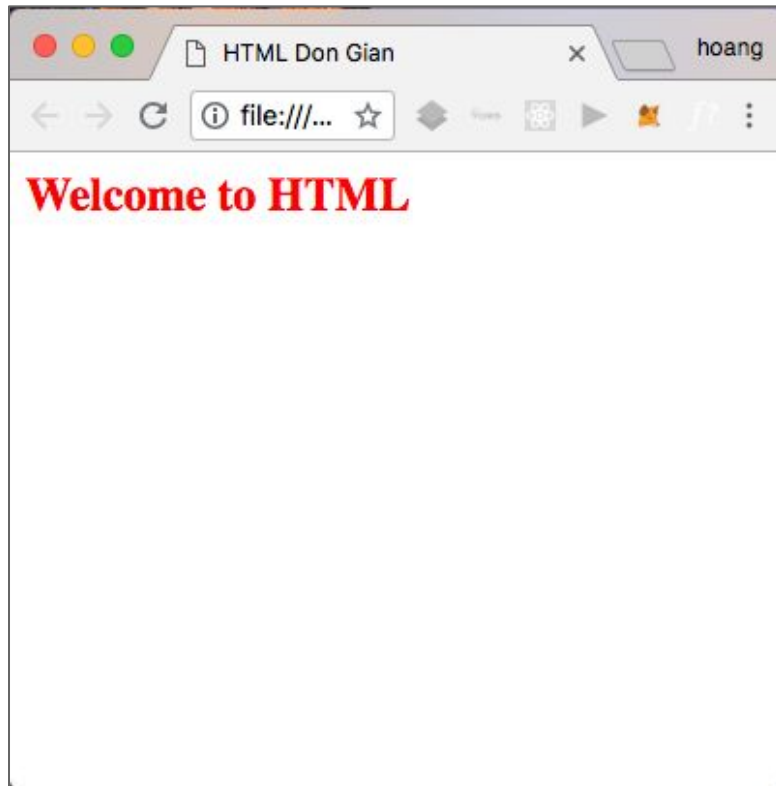
# Các thẻ cơ bản - Thẻ font

- Định dạng hiển thị của văn bản
- `<font> ... </font>`
  - `color`
  - `size`
  - `face`

# Các thẻ cơ bản - Thẻ font

```
<h2>  
  <font color="red"> Welcome to HTML </font>  
</h2>
```

# Các thẻ cơ bản - Thẻ font





# Các thẻ cơ bản - Thẻ hình ảnh

- Dùng để hiển thị hình ảnh trong văn bản
- `<img />`
  - `src`
  - `alt`

# Các thẻ cơ bản - Thẻ font

```

```

# Các thẻ cơ bản - Thẻ bảng

- Dùng để tạo bảng trong văn bản
- `<table>`
- `<tr>` : Định nghĩa dòng
- `<td>` : Định nghĩa cột
- Cú pháp

# Các thẻ cơ bản - Thẻ bảng

```
<table border="1">
  <tr>
    <td>Name</td>
    <td>Email</td>
    <td>Phone</td>
  </tr>
  <tr>
    <td>Nghiem Hoang</td>
    <td>hoangnm@gmail.com</td>
    <td>091234567</td>
  </tr>
</table>
```

# Các thẻ cơ bản - Thẻ bảng

