



VIETNAM
AUSTRALIA
Vocational College

Slides 1.2: Tổng quan về HTML, Semantic Tags

Mentor: Nguyễn Bá Minh Đạo



Nội dung

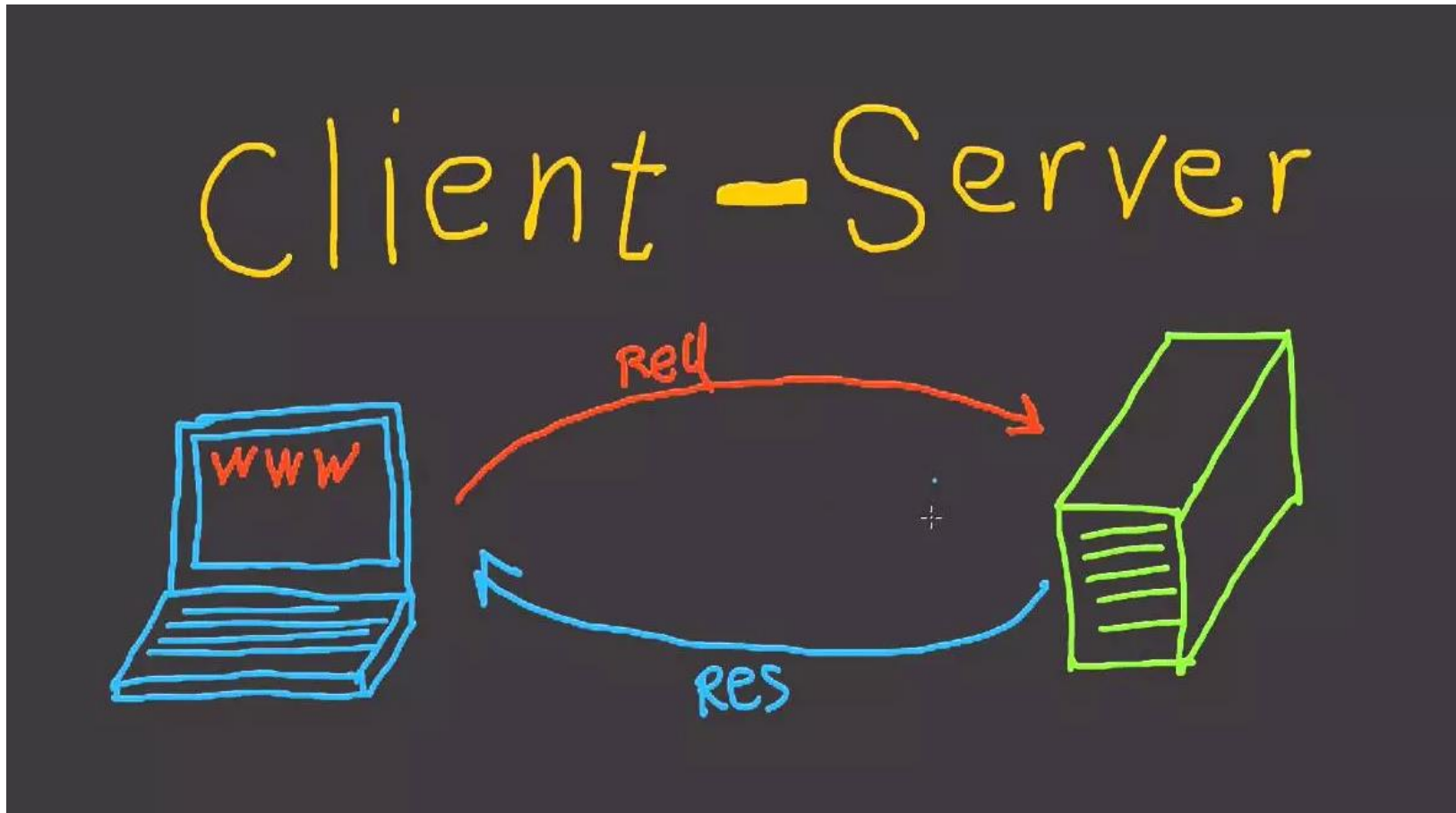
1. Mô hình Web Client – Web Server
2. Sự khác biệt giữa SSR & CSR
3. Các phần tử HTML của trang Web
4. Các thuộc tính của HTML trong Web
5. Các phần tử Semantic trong HTML5
6. Các thẻ hiển thị văn bản
7. Các loại liên kết, đường dẫn



Mô hình Web Client – Web Server

❑ Nhắc lại mô hình Web Server - Web Client:

- Máy chủ web (**Web Server**) – máy khách (**Web Client**)

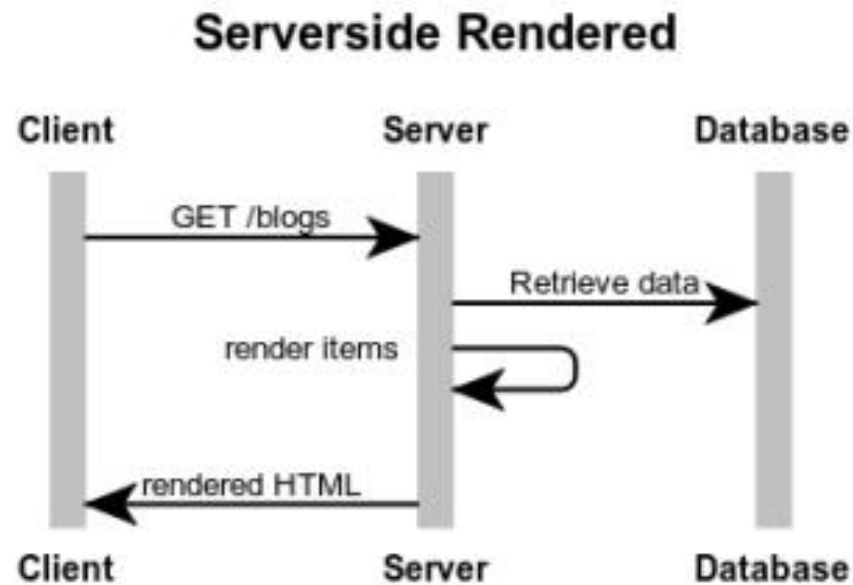




Sự khác biệt giữa SSR & CSR

❑ Cơ chế server-side rendering (SSR):

- Cơ chế Server-Side Rendering đã được dùng từ thuở xa xưa.
- Gọi nó là **server-side rendering** là vì **phần lớn logic** sẽ **được xử lý ở server**:
 - ◆ Khi người dùng vào một trang web, trình duyệt sẽ gửi GET request tới web server
 - ◆ Web server sẽ nhận request, đọc dữ liệu từ database.
 - ◆ Web server sẽ **render HTML**, trả về cho browser để hiển thị cho người dùng

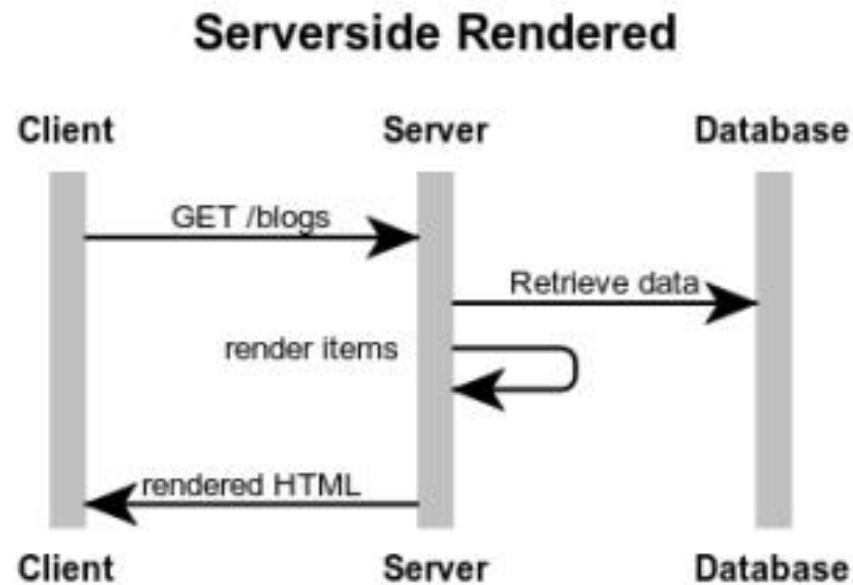




Sự khác biệt giữa SSR & CSR

❑ Cơ chế server-side rendering (SSR):

- Một số tính chất của cơ chế Server-Side Rendering:
 - ♦ Logic từ **đơn giản** (validation, đọc dữ liệu) cho **đến phức tạp** (phân quyền, thanh toán) **đều nằm ở phía server**
 - ♦ Logic để **routing** – chuyển trang **nằm ở server**.
 - ♦ Logic để **render** – hiển thị trang web cũng **nằm ở server** nốt





Sự khác biệt giữa SSR & CSR

❑ Cơ chế server-side rendering (SSR):

➤ Ưu điểm của cơ chế Server-Side Rendering:

◆ Ra đời đã lâu, **server side rendering** đã **được sử dụng, cải tiến rất nhiều** (Caching, CDN). Những **ưu điểm** của cơ chế này:

▪ **Initial load nhanh, dễ optimize**, vì toàn bộ dữ liệu đã được xử lý ở server. Client **chỉ** việc **hiển thị**.

▪ Các web framework từ xưa đến nay **đều hỗ trợ cơ chế này**





Sự khác biệt giữa SSR & CSR

❑ Cơ chế server-side rendering (SSR):

➤ Ưu điểm của cơ chế Server-Side Rendering:

- **Dễ hiểu** và **dễ code hơn**.

Developer chỉ cần code 1 project web là được, không cần phải tách ra front-end và back-end

- **SEO tốt** vì khi bot của Google, Bing vào web sẽ thấy toàn bộ dữ liệu dưới dạng HTML.

- **Chạy** được **trên phần lớn mọi trình duyệt**, kể cả disable JavaScript vẫn chạy tốt



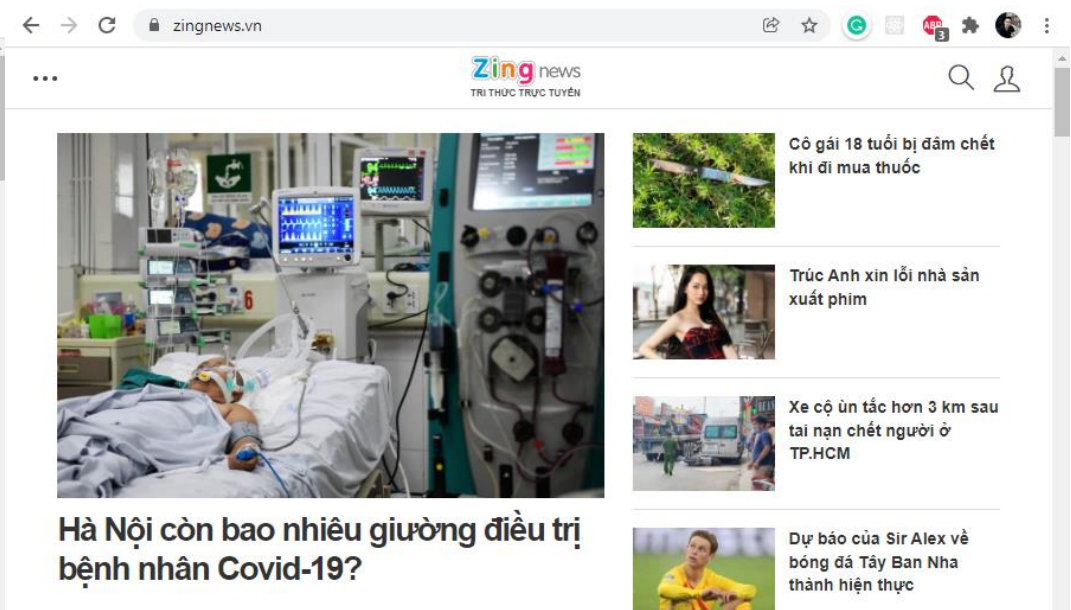
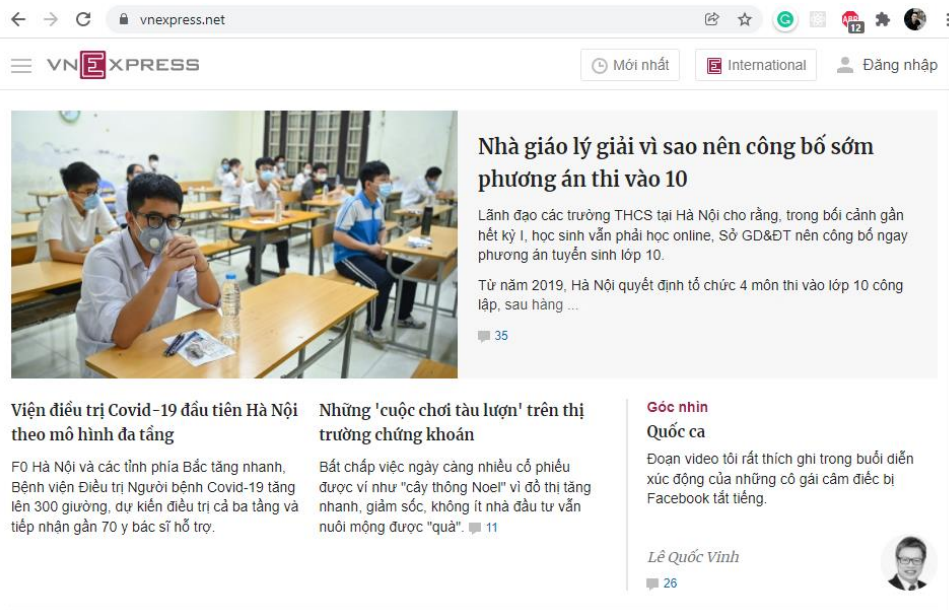


Sự khác biệt giữa SSR & CSR

❑ Cơ chế server-side rendering (SSR):

➤ Nhược điểm của cơ chế Server-Side Rendering:

- **Mỗi lần** người dùng **chuyển trang** là **site phải load lại nhiều lần**, gây **khó chịu**
- **Nặng server** vì **server phải xử lý nhiều logic và dữ liệu**. Có thể sử dụng caching để giảm tải.



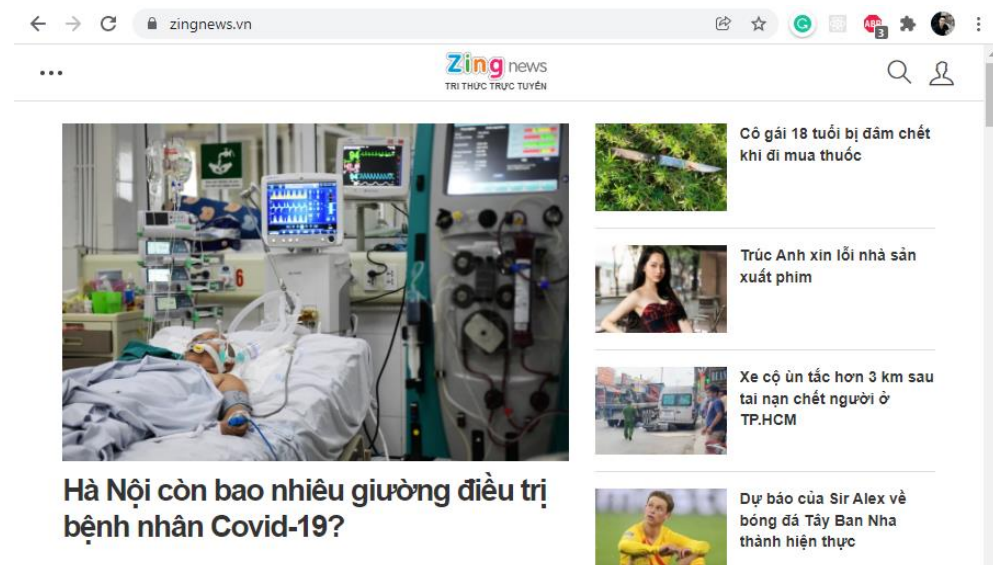
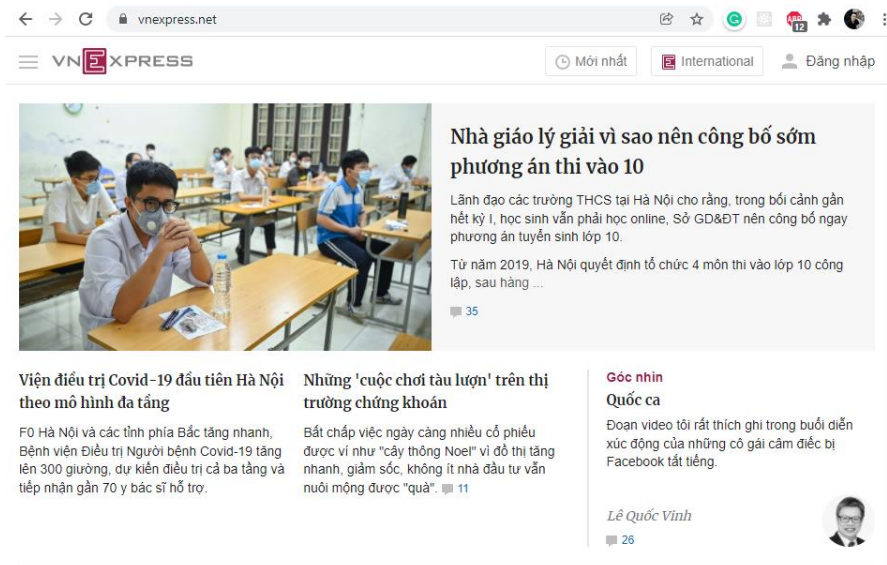


Sự khác biệt giữa SSR & CSR

❑ Cơ chế server-side rendering (SSR):

➤ Nhược điểm của cơ chế Server-Side Rendering:

- **Tốn băng thông** vì **server phải gửi nhiều dữ liệu thừa và trùng** (HTML, header, footer). Có thể sử dụng CDN để giảm tải.
- **Toàn bộ các trang web** được **build từ CMS** như Joomla, Wordpress và **các trang web bán hàng, web tin tức: thegioididong, vnexpress, zing new, ...đều dùng cơ chế này**





Sự khác biệt giữa SSR & CSR

❑ Cơ chế client-side rendering (CSR):

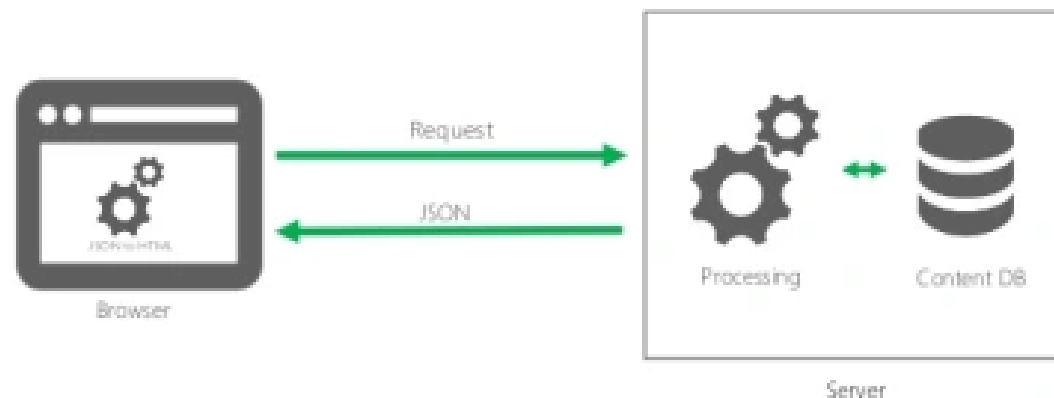
➤ Ở những năm 2010, với **sự phát triển của JavaScript** và **AJAX**, cơ chế client-side rendering **bắt đầu được sử dụng**.

➤ **Developer bắt đầu build ứng dụng dưới dạng SPA – Single Page Application.**

Ứng dụng nằm trong 1 page duy nhất nên được gọi là Single Page Application.

➤ **Client Side Rendering** tức là **việc render HTML, CSS** sẽ **được thực hiện ở client** (tức JavaScript ở trình duyệt)

Client Side Render (CSR)





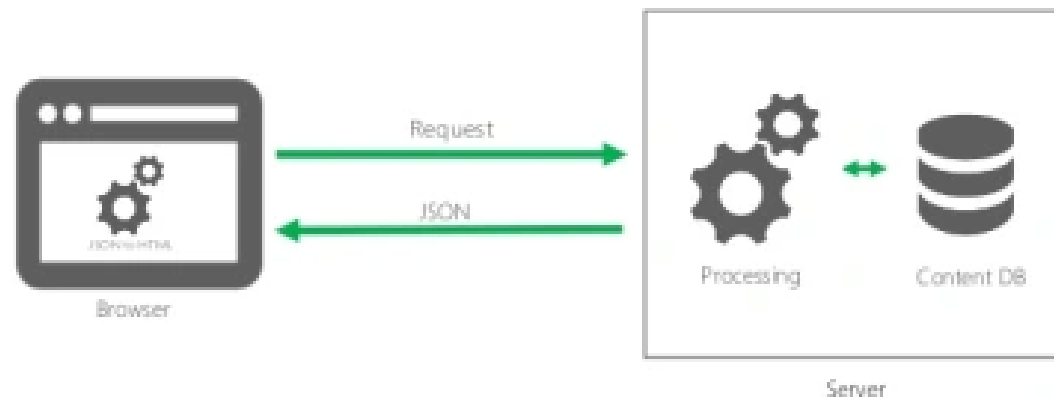
Sự khác biệt giữa SSR & CSR

❑ Cơ chế client-side rendering (CSR):

➤ Client Side Rendering (CSR) có những đặc điểm sau:

- ◆ Những **logic đơn giản** (validation, đọc dữ liệu, sorting, filtering) **nằm ở client side**.
- ◆ Những **logic để routing** (chuyển trang), **render** (hiển thị) **dữ liệu** thì 96.69% là **nằm ở client side**
- ◆ **Logic phức tạp** (thanh toán, phân quyền) hoặc **cần xử lý nhiều** (data processing, report) **vẫn nằm ở server side**.

Client Side Render (CSR)





Sự khác biệt giữa SSR & CSR

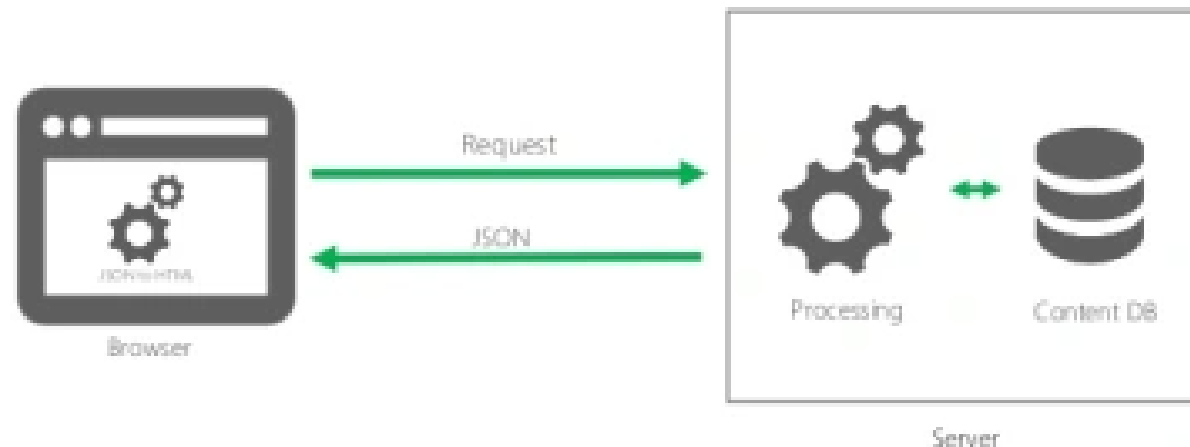
❑ Cơ chế client-side rendering (CSR):

➤ Ưu điểm của Client Side Rendering (CSR):

♦ Page **chỉ cần load một lần duy nhất**. Khi user chuyển trang hoặc thêm dữ liệu, **JavaScript sẽ lấy** và **gửi dữ liệu từ server qua AJAX**. User có thể thấy dữ liệu mới mà không cần chuyển trang.

♦ Chuyển logic sang client nên **giảm tải** được **một phần cho server**.

Client Side Render (CSR)



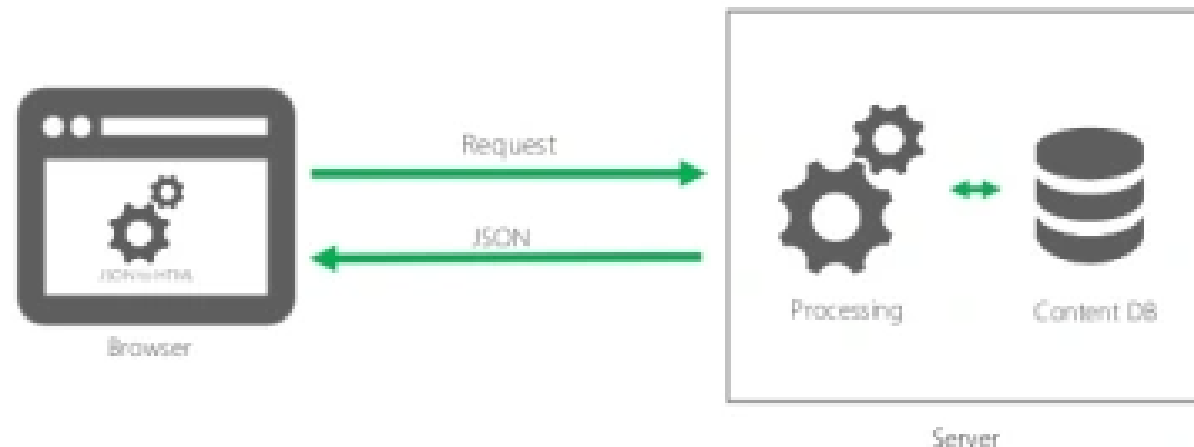


Sự khác biệt giữa SSR & CSR

❑ Cơ chế client-side rendering (CSR):

- Ưu điểm của Client Side Rendering (CSR):
 - ♦ **Giảm** được **băng thông** do **chỉ cần lấy JSON** và **dữ liệu cần thiết**, thay vì phải lấy toàn bộ trang.
 - ♦ **Với** các **ứng dụng** cần **tương tác nhiều**, **SPA hoạt động mượt mà hơn** vì code chạy trên browser, không cần load đi lại nhiều.

Client Side Render (CSR)





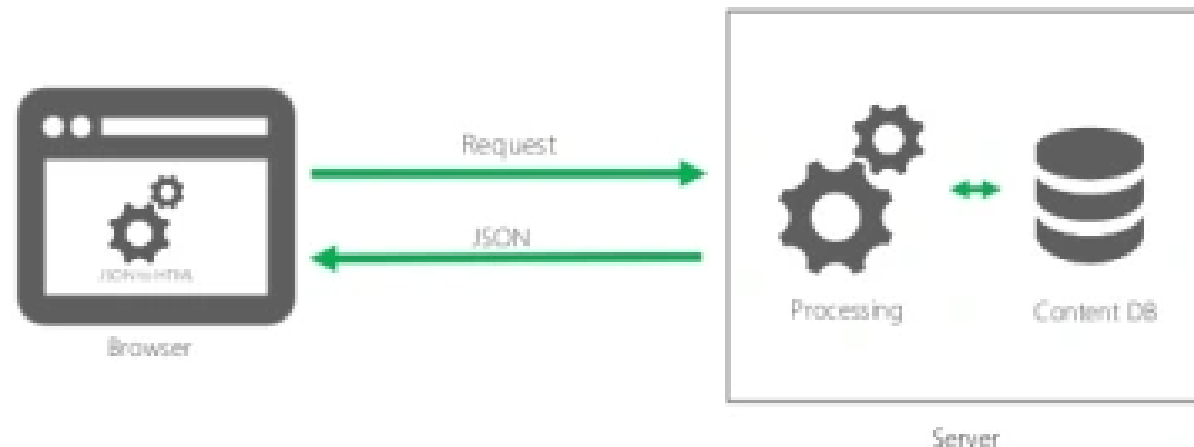
Sự khác biệt giữa SSR & CSR

❑ Cơ chế client-side rendering (CSR):

➤ Nhược điểm của Client Side Rendering (CSR):

- ♦ **Initial load** sẽ **chậm hơn** nếu **không biết optimize** vì browser phải tải toàn bộ JS về (khá nặng), parse, chạy JS, gọi API để lấy dữ liệu từ server (chậm), sau đó render dữ liệu.
- ♦ **Đòi hỏi project** phải **chia làm 2 phần riêng** là **back-end** (REST api) và **front-end** nên **khó code hơn**

Client Side Render (CSR)





Sự khác biệt giữa SSR & CSR

❑ Cơ chế client-side rendering (CSR):

➤ Nhược điểm của Client Side Rendering (CSR):

- ♦ Không chạy được nếu JavaScript bị disable, hoặc ở các trình duyệt cũ không nhận JavaScript ES6 (Có thể dùng transpiler và polyfill nhưng sẽ làm tăng kích cỡ file js)
- ♦ SEO không tốt bằng SSR (Do bot crawl không đọc được dữ liệu). Để giải quyết, ta phải kết hợp thêm SSR (Bot mới của Google đọc được client-side rendering).
- ♦ Nếu client sử dụng mobile, device yếu thì khi load sẽ bị chậm.



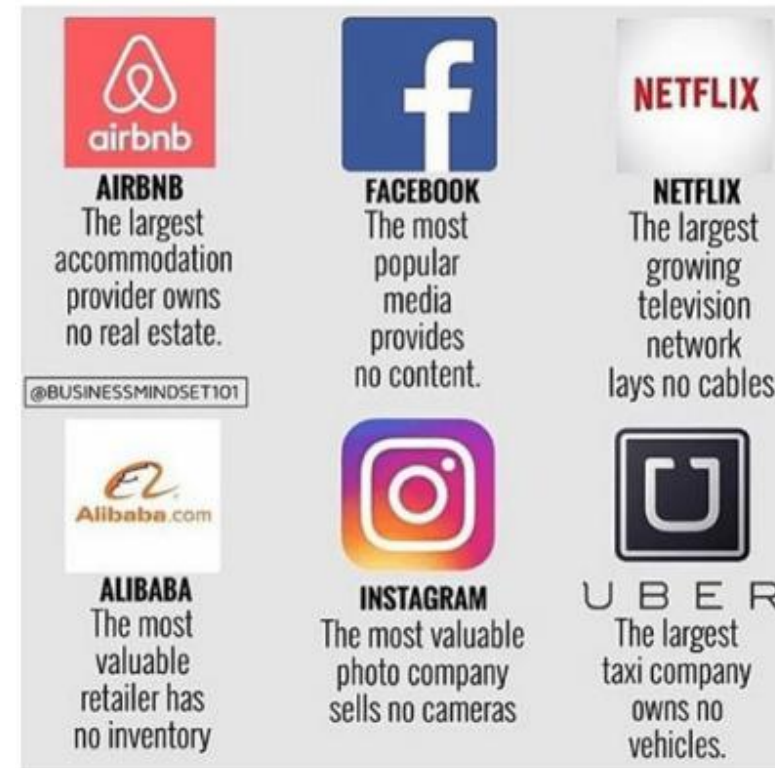


Sự khác biệt giữa SSR & CSR

❑ Các website sử dụng Client-Side Rendering:

➤ Vì Client-side rendering rất phù hợp cho những ứng dụng cần tương tác nhiều, hầu hết web của các công ty công nghệ, công ty startup đều dùng cơ chế này:

- ◆ Facebook (React)
- ◆ Instagram (React)
- ◆ Netflix (React)
- ◆ Dropbox (React)
- ◆ AirBnb (React)
- ◆ Trello (Angular)
- ◆ Paypal (React + Angular)
- ◆ Xiaomi (VueJS)
- ◆ Alibaba(VueJS)
- ◆ Gitlab (VueJS)



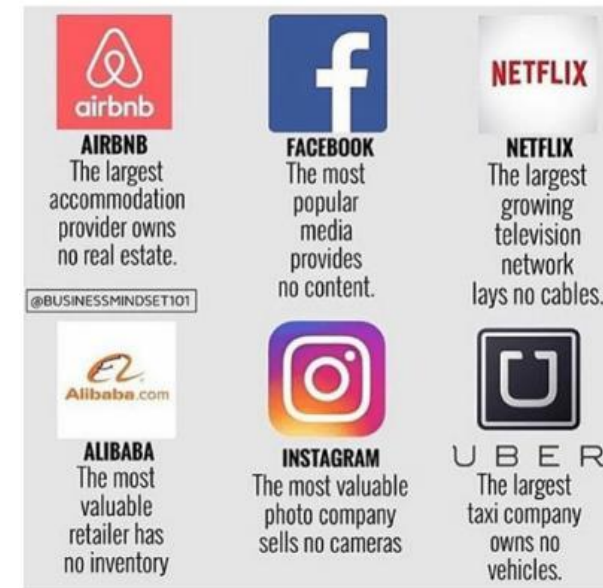
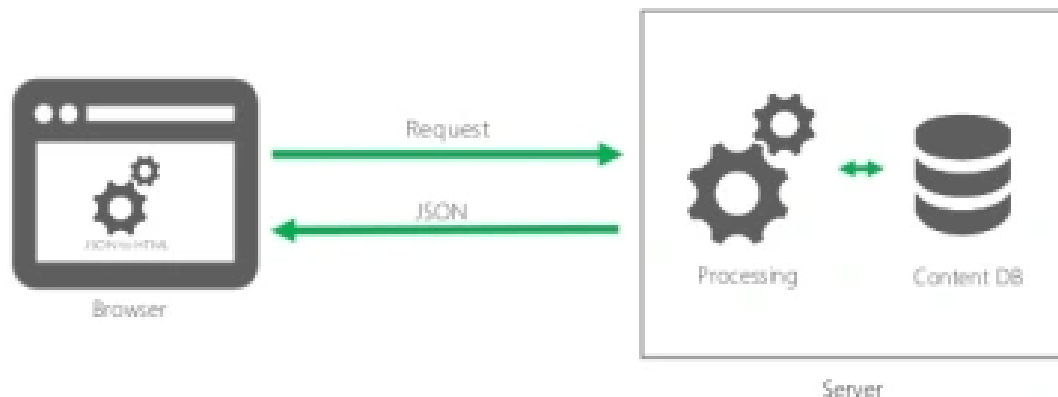


Sự khác biệt giữa SSR & CSR

❑ Các website sử dụng Client-Side Rendering:

- Nói chung, client side rendering còn khá mới mẻ nên vẫn còn một số vấn đề nhỏ chưa giải quyết tốt (tối ưu tốc độ initial load, giảm kích cỡ file JavaScript của app)
- Các vấn đề này vẫn đang được giải quyết bởi các kỹ sư tài năng ở các công ty hàng đầu như Facebook, Netflix, Instagram,...
- Môn thiết kế Web này sẽ chưa đụng tới dữ liệu, hoàn toàn là tạo ra các file web tĩnh.

Client Side Render (CSR)





Các phần tử HTML trong trang Web

❑ Giới thiệu HTML:

- ♦ Các **trang web** được **tạo ra bằng** cách sử dụng **Hypertext Markup Language** (HTML)
- ♦ **HTML** là ngôn ngữ đánh dấu siêu văn bản được dùng **để tạo ra các tài liệu web**.
- ♦ **HTML** bao gồm một **tập hợp các đánh dấu đặc biệt** gọi là **thẻ (tags)** để **xác định cấu trúc và bố cục** của **nội dung** trong một **trang web**.
- ♦ Một **trình duyệt** có thể **đọc các thẻ HTML** để xác định cách **hiển thị nội dung trang web trên màn hình**.
- ♦ Phiên bản mới nhất của **HTML** là **HTML5**.

Rough Timeline of Web Technologies

1991	HTML
1994	HTML 2
1996	CSS 1 + JavaScript
1997	HTML 4
1998	CSS 2
2000	XHTML 1
2002	Tableless Web Design
2005	AJAX
2009	HTML 5



Các phần tử HTML trong trang Web

❑ Các phần tử HTML:

- ♦ Một **trang web** là một **text file** có chứa 2 thứ: **nội dung** và **cặp thẻ HTML**.
 - **<tagname>** Nội dung thẻ **</tagname>**
- ♦ Một **phần tử HTML** **bao gồm tất cả mọi thứ từ thẻ mở tới thẻ đóng**.
 - **<h1>** Xin chào mọi người **</h1>**
- ♦ Một số **thẻ đặc biệt** là **thẻ đơn, không có thẻ đóng**.
 - Ví dụ: **
** hoặc **</br>** (xuống dòng), **<hr>** hoặc **</hr>** (đoạn kẻ ngang)



```
hello-html-tag.html X
html > hello-html-tag.html > h2
1 <h1>Xin chào VUSers</h1>
2 <h2>Tôi là HTML đây!</h2>
```



Các thuộc tính của HTML trong Web

❑ Các thuộc tính HTML:

♦ Một **thuộc tính HTML** dùng để nâng cấp phần tử HTML bằng cách thêm các đặc tính khác nhau cho thẻ HTML.

♦ Ví dụ: thẻ `<hr>` để khởi tạo một đường kẻ ngang trên web có một số thuộc tính sau:

▪ **align**: canh lề đường kẻ ngang.

▪ **width**: chiều rộng đường kẻ ngang.

▪ **size**: độ dày đường kẻ.

▪ **noshade**: không đổ bóng.

♦ Sử dụng:

`<hr noshade size="5px" align="center" width="50%">`

`<hr size="15px" align="right" width="80%">`

```
<> br-hr-tags.html X
html > <> br-hr-tags.html > hr
1 <h1>Xin chào VUSers</h1>
2 <hr noshade size="5px" align="center" width="50%">
3 <h2>Tôi là HTML đây!</h2>
4 <hr size="15px" align="right" width="80%">
```




Các phần tử Semantic trong HTML5

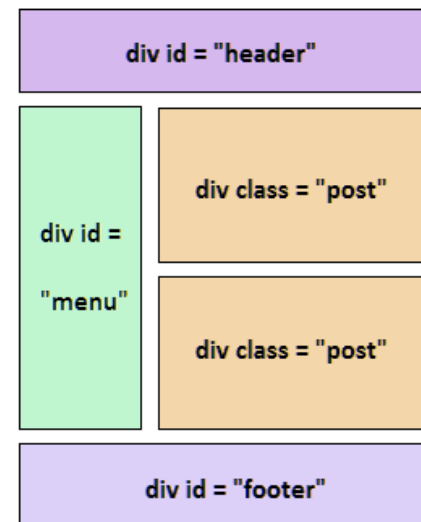
❑ Giới thiệu các phần tử Semantic:

♦ HTML 4.01 **giới thiệu phần tử div** (cặp thẻ `<div> </div>`) để **chia** một **trang web** ra **thành nhiều phần khác nhau**.

- Mỗi thẻ **div** sẽ được đặt id, class khác nhau để phân biệt với các thẻ **div** khác.
- Nhưng quá nhiều **div** sẽ **đễ gây nhầm lẫn** -> Làm sao khắc phục?
➔ Sử dụng các phần tử Semantic của HTML5.

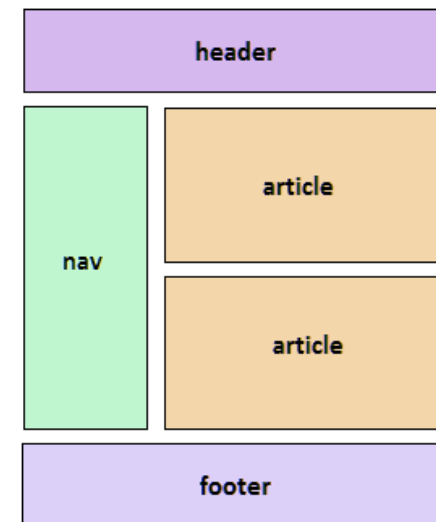


HTML4



VS

HTML5





Các phần tử Semantic trong HTML5

❑ Giới thiệu các phần tử Semantic:

♦ **HTML5** đã chuyển đổi và cải thiện sự phát triển trang web với việc **giới thiệu một số phần tử ngữ nghĩa mới với các tên được chuẩn hóa.**

Phần tử	Mô tả
<code><header>...</header></code>	<ul style="list-style-type: none">- Cho biết thông tin tiêu đề của trang web.- Nội dung tiêu đề thường bao gồm tên hoặc biểu trưng doanh nghiệp- Thường được đặt ngay sau thẻ mở <code><body></code>
<code><nav>...</nav></code>	<ul style="list-style-type: none">- Cho biết điểm bắt đầu và kết thúc của khu vực điều hướng trong trang web- Phần tử điều hướng chứa các siêu liên kết đến các trang khác trong một trang web- Thường được định vị ngay sau khi đóng thẻ <code></header></code>
<code><main>...</main></code>	<ul style="list-style-type: none">- Cho biết phần bắt đầu và phần cuối của phần nội dung chính của một trang web.- Chứa nội dung chính của trang web.- Chỉ một phần tử chính có thể xuất hiện trên một trang.



Các phần tử Semantic trong HTML5

❑ Giới thiệu các phần tử Semantics:

♦ **HTML5** đã chuyển đổi và cải thiện sự phát triển trang web với việc **giới thiệu một số phần tử ngữ nghĩa mới với các tên được chuẩn hóa.**

Phần tử	Mô tả
<footer>...</footer>	<ul style="list-style-type: none">- Cho biết phần đầu và phần cuối của phần chân trang của một trang web.- Chứa nội dung chân trang của trang web
<section>...</section>	<ul style="list-style-type: none">- Cho biết phần bắt đầu và phần cuối của một phần của trang web- Chứa một nhóm nội dung cụ thể trên trang web
<article>...</article>	<ul style="list-style-type: none">- Cho biết phần đầu và phần cuối của một phần của một trang web.- Chứa nội dung như diễn đàn hoặc bài đăng trên blog.
<aside>...</aside>	<ul style="list-style-type: none">- Cho biết phần bắt đầu và phần kết thúc của một phần trang web.- Chứa thông tin về nội dung lân cận của nội dung chính.- Thường được hiển thị dưới dạng thanh bên (trái hoặc phải)



Các phần tử Semantic trong HTML5

❑ Ghi chú trong trang web:

♦ Ghi chú (comment) dùng để cung cấp thêm thông tin bổ sung cho những khu vực trong code và chương trình sẽ không đọc dòng comment này của bạn.

♦ Cú pháp: `<!-- Đặt ghi chú của bạn ở đây -->`

```
<> index.html X
fitness > <> index.html > html > body
1  <!DOCTYPE html>
2
3  <!-- Tác giả: Tèo Nguyễn -->
4  <html lang="vi">
5
6  <head>
7      <title>VUS Fitness Club</title>
8      <meta charset="utf-8">
9  </head>
```



Các phần tử Semantic trong HTML5

❑ Thêm ký tự đặc biệt vào trang web:

♦ Khi thêm nội dung tĩnh vào trang web chúng ta có thể **thường cần dùng các biểu tượng**, ví dụ biểu tượng bản quyền ©,...Thay vì chèn trực tiếp các ký hiệu đặc biệt, ta có thể **chèn một ký hiệu đặc biệt** bằng cách **nhập Entity Name** hoặc **Entity Number**

Ký tự (Character)	Mô tả (Description)	Tên thực thể (Entity Name)	Số thực thể (Entity Number)
©	Ký hiệu bản quyền	©	©
®	Thương hiệu đã được đăng ký	®	®
€	Euro	€	€
&	Ký hiệu và	&	&
<	Ký hiệu nhỏ hơn	<	<
>	Ký hiệu lớn hơn	>	>
	Ký hiệu khoảng trắng	 	



Các thẻ hiển thị văn bản

❑ Một số thẻ hiển thị văn bản thường gặp:

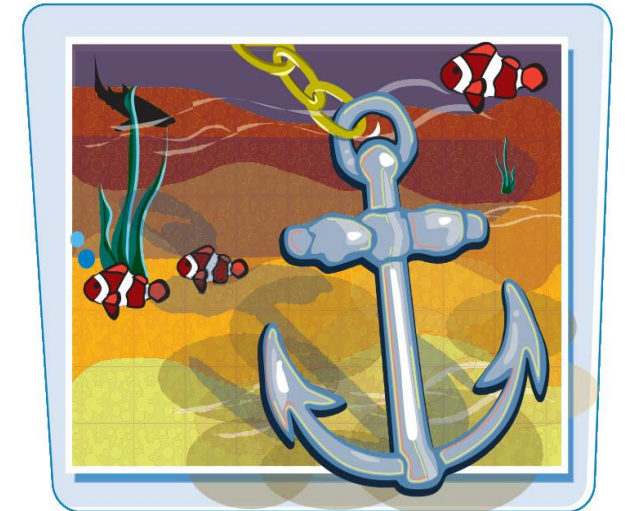
Thẻ	Công dụng	Ví dụ
<code><h1></code> , <code><h2></code> , ..., <code><h6></code>	Hiển thị tiêu đề trong trang Web	Tiêu đề đoạn văn bản, câu, từ cần in đậm để làm nổi bật
<code>
</code>	Xuống dòng trong trang Web	Cách dòng nhiều đoạn văn bản
<code><hr></code>	Khởi tạo một đường kẻ ngang	Đường kẻ ngăn cách các phần header, nav, main, footer
<code><p></code>	Hiển thị đoạn văn bản	Xin chào tất cả các bạn.
<code><i></code> , <code></code>	Tạo chữ nghiêng, chữ in đậm	<i>Hello World</i> , Hello World
<code><sub></code>	Tạo một ký tự nhỏ bên dưới	Công thức hóa học H ₂ O, CO ₂ ,...
<code><sup></code>	Tạo một ký tự nhỏ phía bên trên	Số mũ toán học $a^2 + b^2 = c^2$
* Lưu ý: Ta có thể lồng các thẻ inline: <code></code> , <code><i></code> ,... vào trong các thẻ block: <code><p></code> , <code><h1></code> ,... để hiển thị các đoạn văn bản phù hợp trong trang Web.		



Các loại liên kết, đường dẫn

❑ Thẻ <a> trong HTML

- ◆ <a> - anchor: thẻ tạo các siêu liên kết nối các trang web.
- ◆ Cấu trúc thẻ <a> như sau:
` Linked content `
- ◆ **URL**: đường dẫn truyền vào để đi tới các trang web khác.
- ◆ Thuộc tính target của thẻ <a>:
 - **_blank**: tải trang web vào cửa sổ mới.
 - **_self**: tải trang web vào chính cửa sổ mặc định.



- ◆ Có 4 loại liên kết chính trong HTML:
 - **External link**: liên kết ngoại
 - **Internal link**: liên kết nội
 - **Email link**: liên kết email
 - **Telephone link**: liên kết telephone





Các loại liên kết, đường dẫn

❑ Có 4 loại liên kết chính trong HTML :

♦ **Liên kết ngoại (External Link):** dùng để **tạo liên kết kết nối các trang Web nằm ngoài trang Web hiện tại**. Có 2 cách dùng liên kết ngoại thông qua 2 loại đường dẫn: **đường dẫn tuyệt đối, đường dẫn tương đối**. Vd:



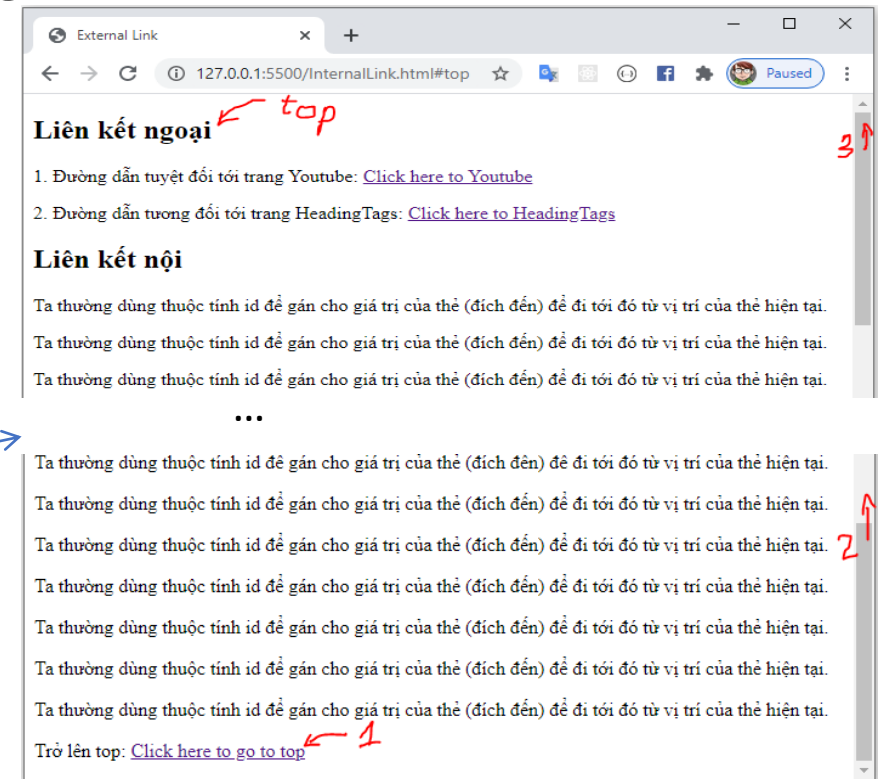
Các loại liên kết, đường dẫn

❑ Có 4 loại liên kết chính trong HTML:

◆ **Liên kết nội (Internal Link):** dùng để tạo liên kết kết nối các đoạn dữ liệu trang Web nằm ngay trên trang Web hiện tại. Ta thường dùng thuộc tính **id** của các thẻ làm giá trị của thuộc tính **href** để đi tới vị trí dữ liệu có gắn thuộc tính **id** như ví dụ sau:

```
InternalLink.html X
InternalLink.html > html > body > p
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>External Link</title>
5 </head>
6 <body>
7 <h2 id="top">Liên kết ngoại</h2>
8
9 <!-- Liên kết ngoại - Đường dẫn tuyệt đối -->
10 <p>1. Đường dẫn tuyệt đối tới trang Youtube:
11 <a href="https://www.youtube.com">Click here to Youtube</a>
12 </p>
13
14 <!-- Liên kết ngoại - Đường dẫn tương đối -->
15 <p>2. Đường dẫn tương đối tới trang HeadingTags:
16 <a href="../HeadingTags.html">Click here to HeadingTags</a>
17 </p>
18
19 <!-- Liên kết nội - Đường dẫn là giá trị thuộc tính id -->
20 <h2>Liên kết nội</h2>
21 <p>Ta thường dùng thuộc tính id để gắn cho giá trị của thẻ (đích đến)
22 để đi tới đó từ vị trí của thẻ hiện tại. </p>
```

```
InternalLink.html X
InternalLink.html > html > body > p
64 để đi tới đó từ vị trí của thẻ hiện tại. </p>
65 <p>Ta thường dùng thuộc tính id để gắn cho giá trị của thẻ (đích đến)
66 để đi tới đó từ vị trí của thẻ hiện tại. </p>
67 <p>Ta thường dùng thuộc tính id để gắn cho giá trị của thẻ (đích đến)
68 để đi tới đó từ vị trí của thẻ hiện tại. </p>
69
70 <p>Trở lên top:
71 <a href="#top">Click here to go to top</a>
72 </p>
73 </body>
74 </html>
```





Các loại liên kết, đường dẫn

❑ Có 4 loại liên kết chính trong HTML :

- ◆ Liên kết email: dùng để **tạo liên kết tới giao diện ứng dụng gửi Mail.**
- ◆ Liên kết telephone: dùng để **tạo liên kết tới giao diện ứng dụng gọi điện thoại.**

```
<h2>Ready to get started? Contact us today.</h2>
<p>We are located at:<br> 402 NTMK<br> District 3, HCM city</p>

<p>Phone Number<br>
|   <a class="mobile" href="tel:8145556789">(814) 555-9608</a>(814) 555-6789</p>
|
<p>Email Address<br>
|   <a href="mailto:vusfitness@club.com">vusfitness@club.com</a></p>
|   <br><br>
```



Các loại liên kết, đường dẫn

❑ Có 2 loại đường dẫn trong HTML :

- ♦ Đường dẫn **tuyệt đối**: là vị trí tuyệt đối so với mạng Internet (gồm một chuỗi đầy đủ bao gồm *http://, tên miền của trang web, đường dẫn đến tập tin*)
- ♦ Đường dẫn **tương đối**: là vị trí tương đối so với trang web hiện tại đang chứa liên kết (thông thường là *đường dẫn đến tập tin*)

Ký hiệu	Ý nghĩa
./	Thư mục hiện tại của website sử dụng link (mặc định)
../	Quay về thư mục cha (đi ngược lại 1 cấp thư mục)



Tổng kết nội dung bài học

- ☐ Mô hình Web Client – Web Server
- ☐ Sự khác biệt giữa SSR & CSR
- ☐ Các phần tử HTML của trang Web
- ☐ Các thuộc tính của HTML trong Web
- ☐ Các phần tử Semantic trong HTML5
- ☐ Các thẻ hiển thị văn bản
- ☐ Các loại liên kết, đường dẫn

Let's
Recap

