



# REACT NATIVE

CYBERSOFT.EDU.VN



# REACT NATIVE ? TẠI SAO NÊN HỌC?

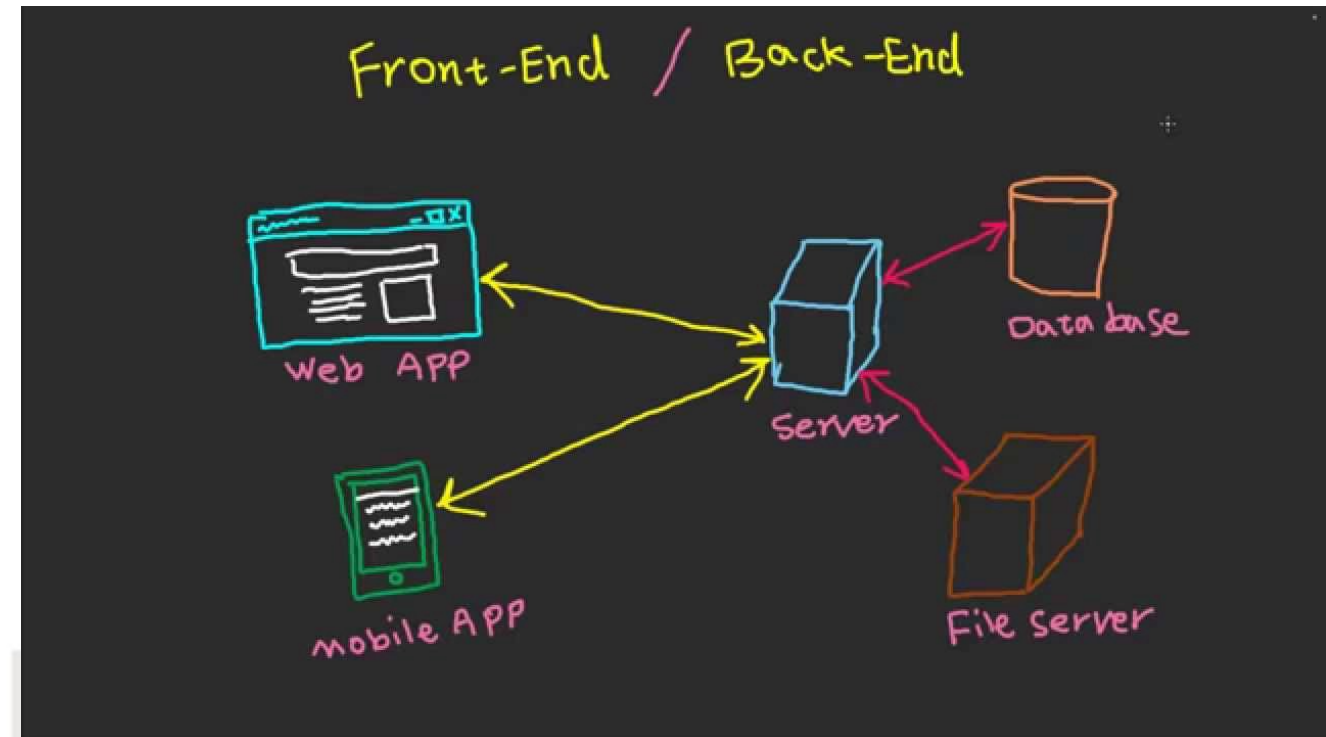
## ❑ React native là gì ?

- React native là một công nghệ được Facebook phát triển vào năm 26-03-2015 dùng để phát triển ứng dụng nền tảng **Mobile** cụ thể là android hoặc ios (UWP trong tương lai phát triển mạnh nền tảng ứng dụng của microsoft chạy trên thiết bị di động cụ thể win10 và 10 mobile).

## ❑ Tại sao nên dùng react native để phát triển ứng dụng di động (mobile App) ?

- Hiện tại trên thế giới có khá nhiều hệ điều hành chạy trên các thiết bị di động điển hình như **IOS** (Iphone) hay các điện thoại **Android** (Samsung, Google, Sony, HTC, Bphone ...) hay một số điện thoại Nokia chạy hệ điều hành **Winphone**.
- Lập trình viên phát triển ứng dụng (app) thì cần phải thông thạo các ngôn ngữ khác nhau trên các hệ điều hành khác nhau. Ví dụ:
  - ♦ Viết ứng dụng trên IOS: C, C++, Objective-C, Swift ...
  - ♦ Viết ứng dụng trên Android: Java, C, C++, XML, Hợp ngữ, Python, Câu lệnh shell, Go, Make, D ...
  - ♦ Winphone: C, C++, C# ...

➡ Tuy nhiên với tốc độ và nhu cầu phát triển ứng dụng xã hội hiện tại thì để 1 lập trình viên phát triển được 1 sản phẩm (1 ứng dụng) trên nhiều hệ điều hành khác nhau rất chi là tốn nhiều thời gian, và tốn khá nhiều chi phí. Ví dụ: 1 dev ios, 1 dev android, 1 dev winphone ... Vì vậy react native ra đời với 1 đích tối ưu hoá thời gian làm sản phẩm cũng như tối ưu hoá chi phí cho sản phẩm chỉ cần người dùng thông thạo XML ( đừng sợ giống html web thôi mà ^^!), css, javascript (react càng tốt !). Nghe có vẻ gần giống lập trình web nhỉ. Đúng react native tương đồng so với web khá nhiều nên sẽ rất dễ học. Khi viết ứng dụng chúng ta có thể build ra nhiều app dành cho nhiều nền tảng (platform mobile) khác nhau: IOS, Android, ...



# REACT NATIVE CÓ THỰC SỰ TỐT ?



## ❑ ƯU ĐIỂM ?

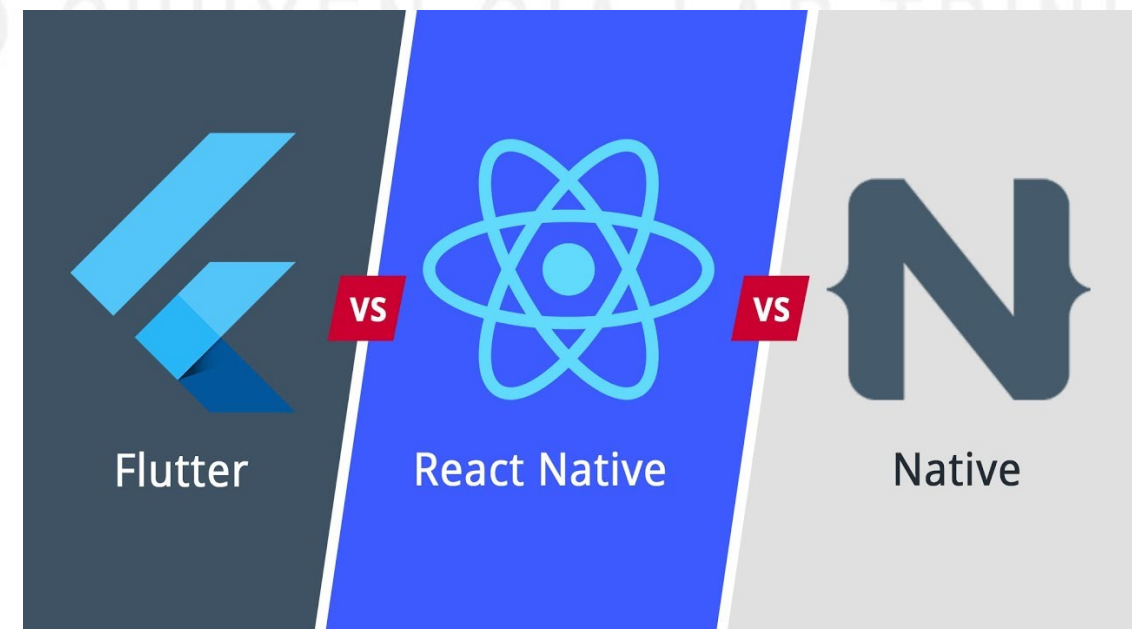
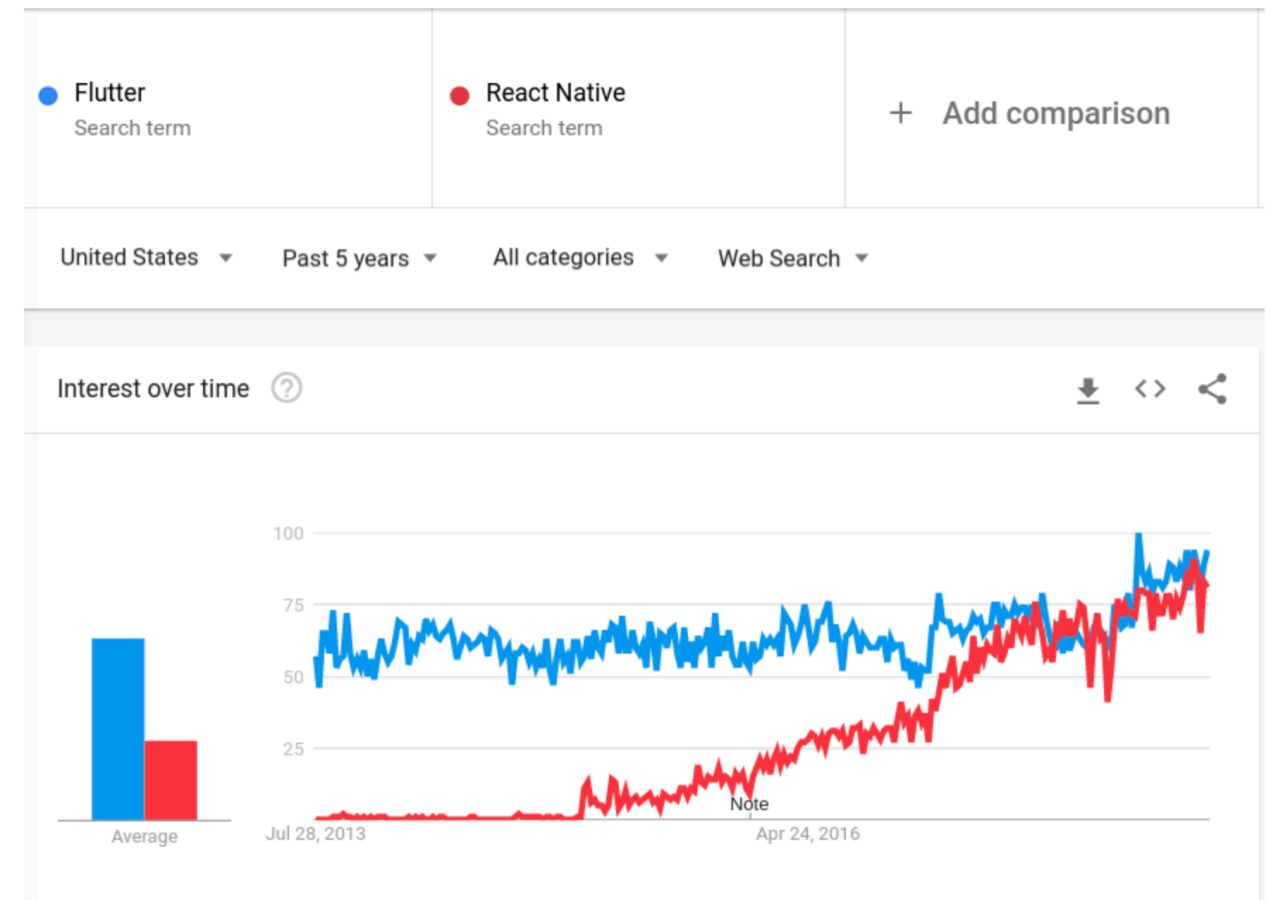
- Gần giống so với lập trình website (FE Web). Vì vậy nên dễ tiếp cận
- Tiết kiệm nhân sự thời gian build sản phẩm
- Cộng đồng lớn (FB chống lưng)
- Cơ hội tìm việc cao
- Hiệu năng tương đối ổn định

## ❑ NHƯỢC ĐIỂM ?

- Nói như trên không có nghĩa là react native thực sự hoàn hảo. Những ngôn ngữ trung gian luôn luôn không thể nào bằng ngôn ngữ gốc phát triển hệ điều hành của nó. Sẽ thiếu thốn thư viện nhiều khi cần can thiệp sâu vào tính năng nâng cao của hệ điều hành ( app chạy ngầm, app chạy theo lịch, deep link ...) vì đó là nền tảng riêng của từng hệ điều hành (platform specific) mà ReactNative không có sẵn (Đối với dự án lớn lâu dài cần can thiệp sâu để phát triển) - trong khoá học sẽ gặp và config khá vất vả.
- Điều đó dẫn đến hiệu năng sẽ chênh lệch 1 ít so với ngôn ngữ thuần của từng hệ điều hành (chậm hơn xíu)
- Bảo mật không cao do viết bằng js.
- Quản lý bộ nhớ không tốt. Nếu tập khách hàng Iphone X, 11 ... hay samsung note 10 +++++ ( thì cũng không ảnh hưởng lắm đâu nhỉ ^^!).
- Khả năng tùy biến chưa thực sự tốt do lý do đầu tiên

## ❑ Đối trọng Flutter (Google)

- Cũng giống như react native (của facebook). Một đối trọng khác của native mobile app là Flutter (của Google) cũng khá mạnh về lập trình mobile app đa nền tảng. Tuy nhiên vẫn khó có thể so sánh toàn diện cái nào hay hơn cái nào. Mỗi cái đều có thể mạnh riêng.
- Tuy nhiên học react native chúng ta dễ dàng mở rộng qua phần dev frontend web (React JS) vì kiến trúc của reactnative và reactjs giống nhau khá nhiều. Còn nếu flutter thì chúng ta cần phải học 1 ngôn ngữ mới của Google: DART, bloc pattern, DART Streaming (Google từng có tiền sử quảng con giữa chợ như angular js, kotlin ... :))) .Nói vậy chứ nó vẫn phát triển nhưng không trending lắm thôi !). Tuy nhiên về hiệu năng animation độ mượt v...v... thời điểm hiện tại Flutter nhỉnh hơn tí.
- Nói vậy không có nghĩa là React Native thua kém hoàn toàn so với Flutter vì hiện tại cộng đồng React Native khá lớn và FB dùng nó để phát triển Product của mình nên tương lai khả năng hoàn thiện chắc chắn hơn và tính ổn định cao. Còn Flutter dù đã release 1.0 chính thức tuy nhiên còn khá mới. Một số plugin rất quan trọng như Google Map vẫn còn đang phát triển, chưa ổn định (stable).





## ❑ Tại sao phải học HTML CSS khi học react native.

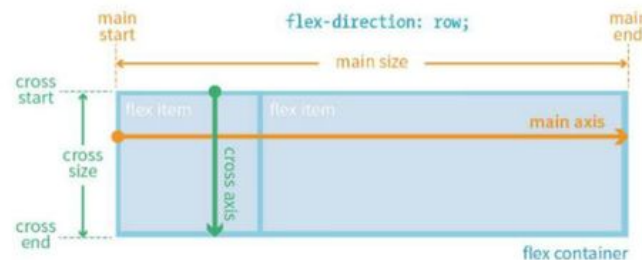
- Khi xây dựng ứng dụng bất kể là web hay là app mobile thì chúng ta đều cần phải học các lý thuyết căn bản như bố cục giao diện tinh chỉnh làm đẹp giao diện. Bởi 1 ứng dụng viết ra nếu không bố cục hợp lý, xấu cũng không ai sử dụng.
- Vậy tại sao không phải học thẳng vào dàn bố cục layout trên react native trước mà lại học web trước ? Đơn giản bởi vì web dễ tiếp cận hơn ít lỗi vặt hơn trong quá trình xây dựng giao diện và thực hiện xử lý. Ví dụ app thiếu đóng thẻ => Lỗi biên dịch. Web thiếu đóng thẻ vẫn chạy => kết quả sai kiểm tra rõ trên giao diện (Chạy nhưng hiển thị sai). Tóm lại ta tiếp cận với các lý thuyết layout trên nền tảng web trước qua React Native sẽ tiếp thu dễ hơn rất nhiều.

### GIAO DIỆN WEB

## CSS Layout - Flexbox

FOR LANGUAGES THAT READ HORIZONTALLY FROM LEFT TO RIGHT:

When `flex-direction` is set to `row`, the main axis is horizontal and the cross axis is vertical.



When `flex-direction` is set to `column`, the main axis is vertical and the cross axis is horizontal.

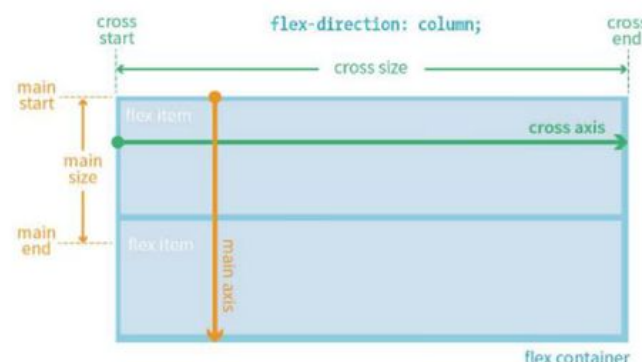


Figure 16-4. Niederst, J. (2018). Learning Web Design. O'Reilly Media, Inc.

### flex items

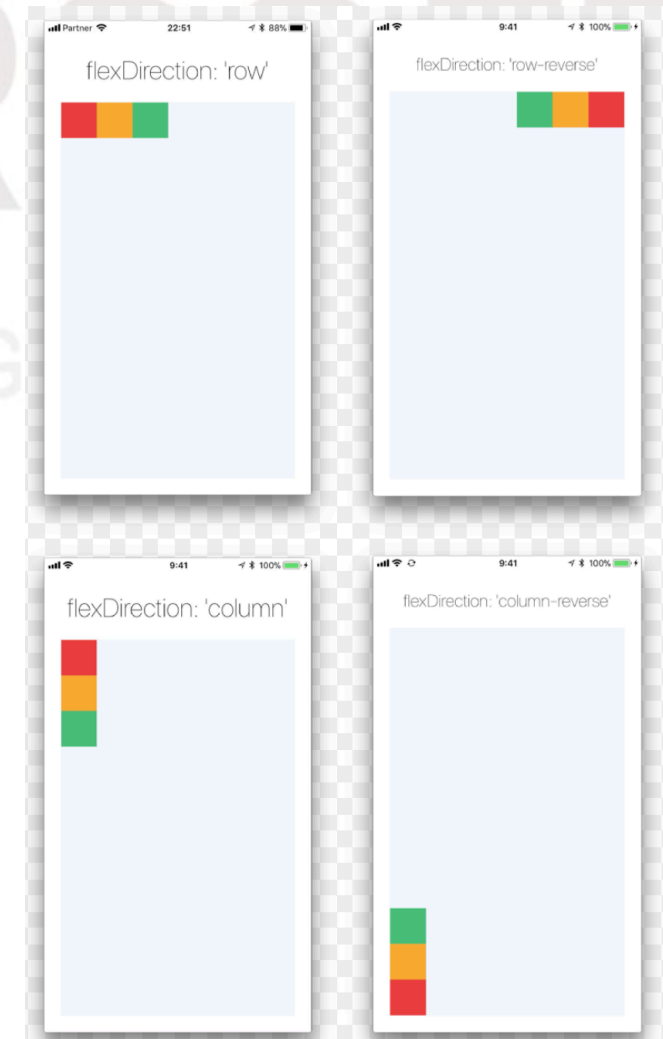
flex items line up along one axis

main axis -or- cross axis

The main axis is the flow direction you've specified for the flex container. The cross axis is perpendicular to the main axis

Note: axis direction is specific to the direction of the writing system in use.  
For example: In horizontally oriented languages – "row" would align items horizontally  
Vertically oriented languages – "row" would align items vertically

### GIAO DIỆN APP



\* **KHOÁ HỌC NÀY CHÚNG TA KHÔNG QUÁ CHUYÊN SÂU VỀ WEB NÊN TA CHỈ HỌC SƠ 1 SỐ KHÁI NIỆM CHIA BỐ CỤC TRANG WEB THÔNG QUA CÁC TAG HTML THÔNG DỤNG**

\* **CÀI ĐẶT CÔNG CỤ THỰC HIỆN:**

- ☐ Notepad++
- ☐ Sublime text
- ☐ **Visual Code** - <https://code.visualstudio.com/>
- ☐ Dreamweaver
- ☐ CotEditor (Macbook)
- ☐ Pingendo (support bootstrap)
- ☐ WebStorm

CYBERSOFT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

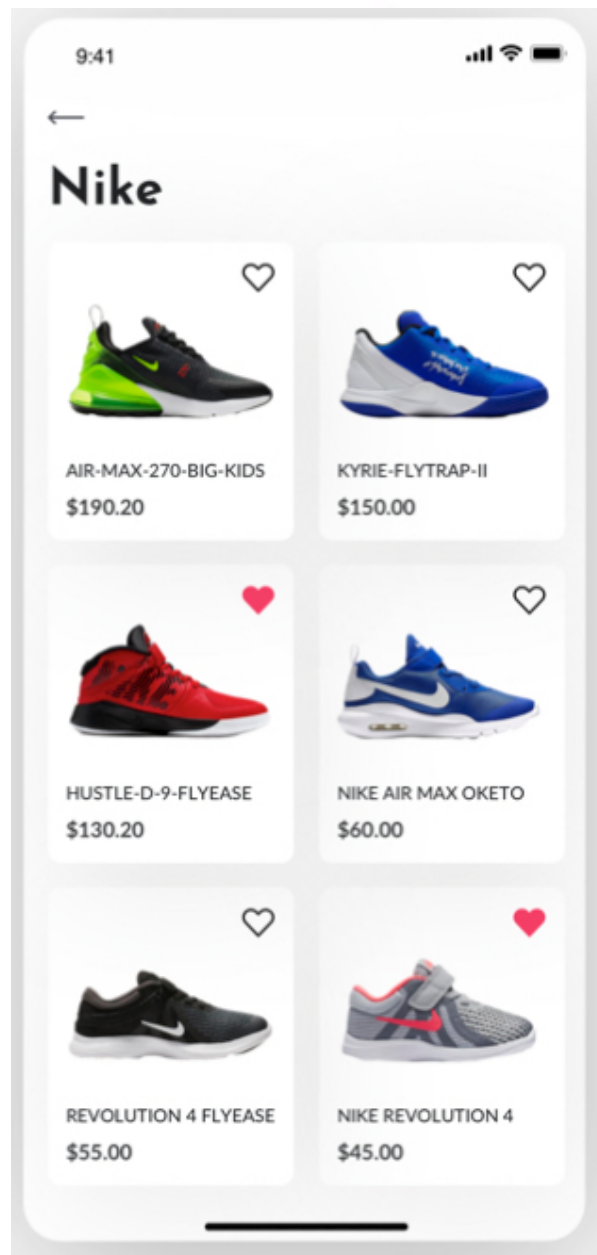
- **HTML (Hyper Text Markup Language)** – Ngôn ngữ đánh dấu siêu văn bản
- Là một ngôn ngữ được sử dụng rộng rãi nhất dùng để xây dựng một trang Web.
- Định dạng bố cục, giúp hiển thị nội dung của trang web lên browser (Trình duyệt web) thông qua các thẻ (tag).
- Trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để hiển thị nội dung của trang Web
- Chứa các siêu liên kết giúp ta kết nối các trang web lại với nhau.

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

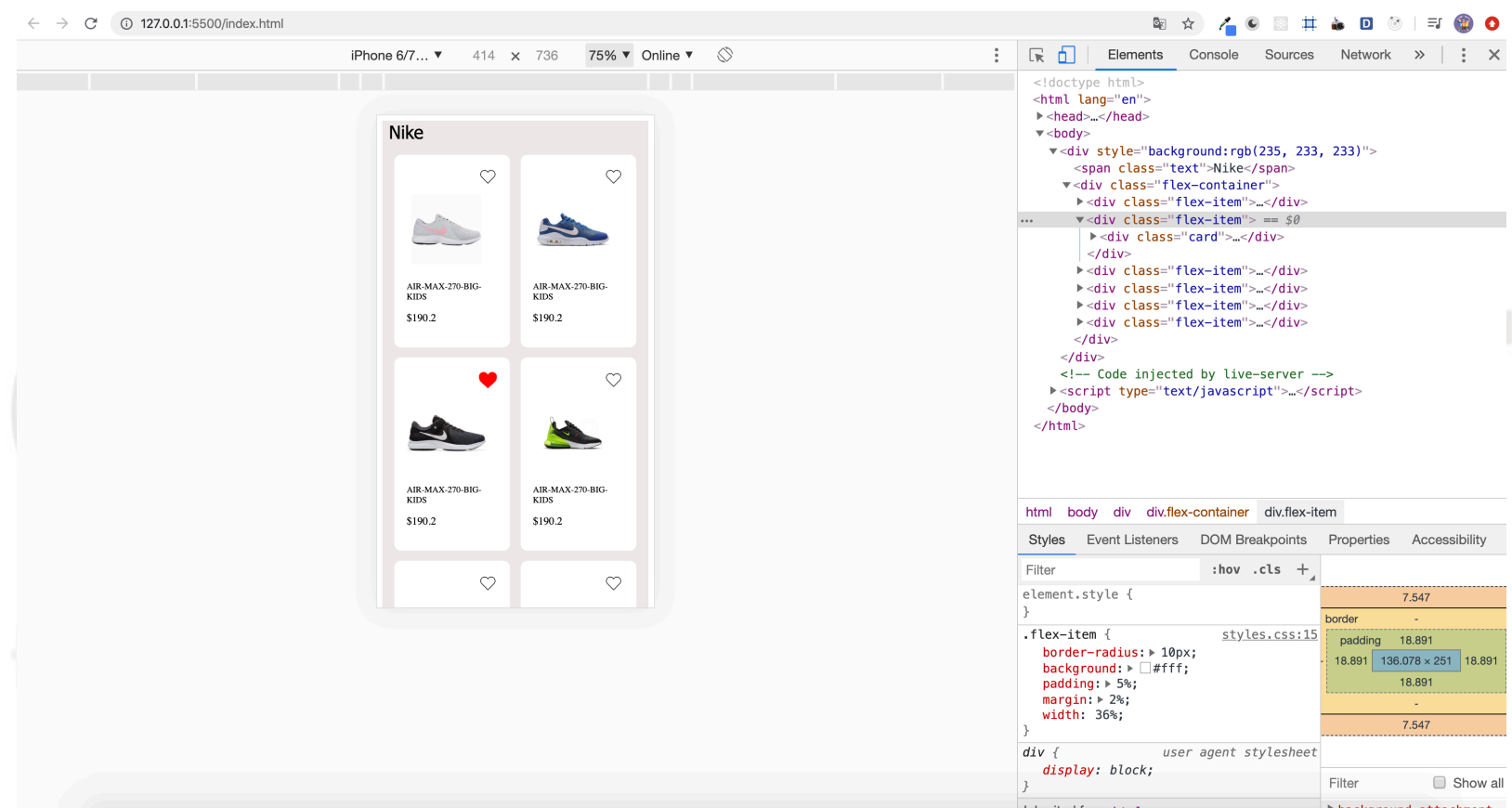
## ★ Nhiệm vụ buổi học

Hoàn thành các layout sau đây trên trình duyệt web

◆ Mobile



◆ Web





Cấu trúc thẻ (tag):

```
<tagname> content... </tagname>
```

- Các thẻ HTML thường đi theo 1 cặp cấu trúc như trên (ex: <p> </p>)
- Tên thẻ thường mang tính gợi nhớ và không phân biệt hoa thường  
Ex: p ~ paragraph (đoạn), b ~ bold (in đậm), i ~ italic (in nghiêng)
- Một số thẻ đặc biệt là thẻ đơn, không có thẻ đóng
  - <br> hoặc <br/> (xuống dòng)
  - <hr> hoặc <hr/> (đoạn kẻ ngang)

## ❑ Các thẻ thông dụng tạo bố cục nội dung cho web

- Thẻ <div>
- Thẻ <p>
- Thẻ <img />



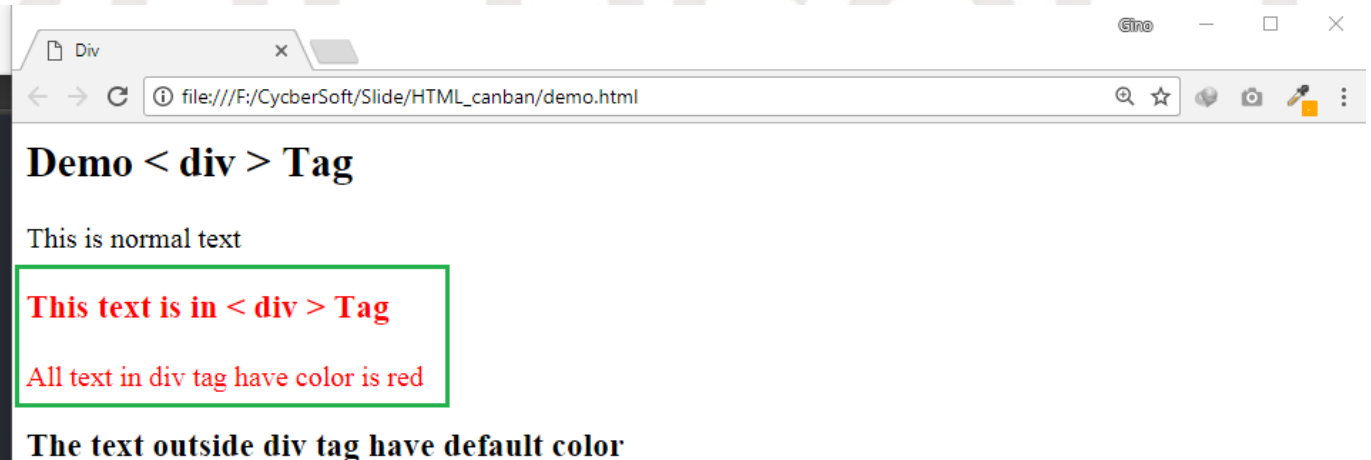
CYBERSOFT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

## ❑ Thẻ <div> - division

- Là một thẻ quan trọng của html
- Được dùng để gom nhóm các thành phần html thành một khu vực trên trang web.
- Thẻ <div> là một phần tử khối, cho nên trước và sau phần tử <div> sẽ bị ngắt xuống dòng
- Có thể định dạng thuộc tính cho tất cả thẻ con bên trong

```
F:\CyberSoft\Slide\HTML_canban\demo.html - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
demo.html
1 <html>
2 <head>
3 <meta charset="UTF-8">
4 <title>Div</title>
5 </head>
6
7 <body>
8
9 <h2>Demo < div > Tag </h2>
10 <p>This is normal text</p>
11
12 <div style="color: red">
13 <h3>This text is in < div > Tag</h3>
14 <p>All text in div tag have color is red</p>
15 </div>
16
17 <h3>The text outside div tag have default color</h3>
18 </body>
19 </html>
```



## ❑ Một số thuộc tính style mới

- Width: xét chiều rộng cho phần tử HTML
- Mặc định những phần tử block sẽ có width = 100% so với thành phần chứa nó.
- Thuộc tính width cho phép căn chỉnh chiều rộng theo ý muốn. Đơn vị :px, rem, em, %.

```
<div style="width:200px;"> content... </div>
```

- Height: xét chiều cao cho phần tử HTML
- Height của thẻ <div> mặc định sẽ bằng với chiều cao của các phần tử bên trong.
- Thuộc tính height cho phép căn chỉnh chiều rộng theo ý muốn. Đơn vị :px, rem, em.

```
<div style="height:200px;"> content... </div>
```

## ❑ Thẻ <p> trong HTML

- Sử dụng thẻ <p> để hiển thị đoạn văn bản trong HTML.

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
</head>
<body>
  <p>
    Cơn mưa ngang qua mang em đi xa, cơn mưa ngang qua khiến em nhạt
    nhòa.
    Chẳng một lời chào rồi từ biệt cho con tim anh        yếu mềm.
  </p>
</body>
</html>
```

Cơn mưa ngang qua mang em đi xa, cơn mưa ngang qua khiến em nhạt nhòa. Chẳng một lời chào rồi từ biệt cho con tim anh yếu mềm.



## Tag hình ảnh <img> : Khuyết thẻ đóng

```

```

### Thuộc tính:

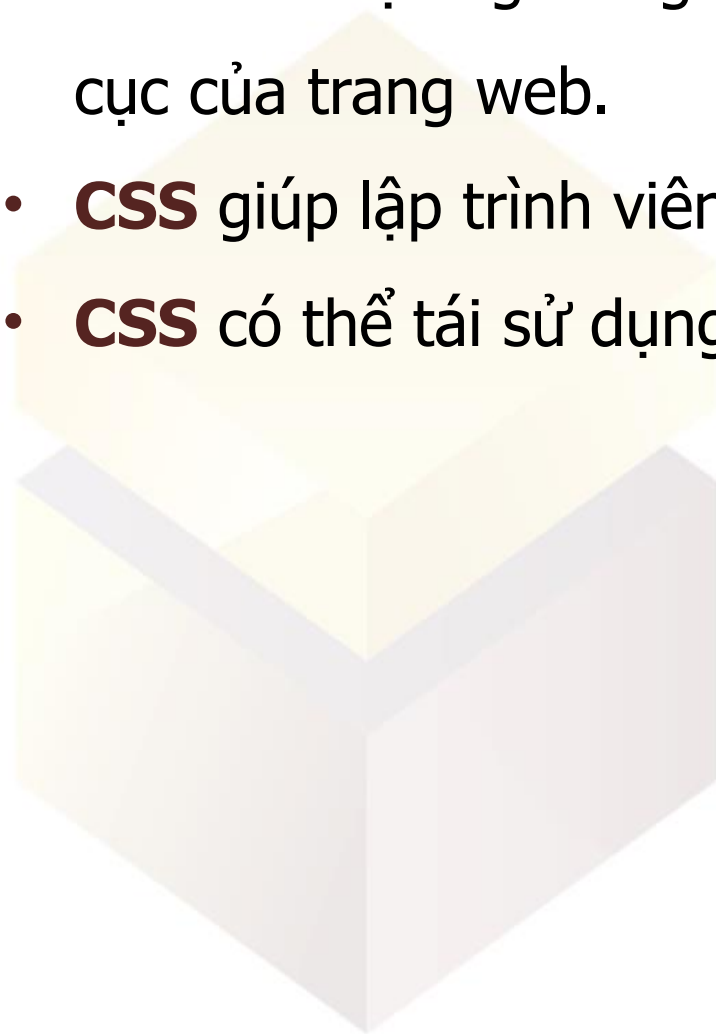
- **src** : đường dẫn đến file ảnh, sử dụng đường dẫn tương đối.
- **alt** : chú thích ảnh
- **border** : độ dày viền
- **width/height** : thiết lập chiều rộng/dài cho ảnh

### Đặt ảnh nền cho trang web:

```
<body background="url"> </body>
```

# CSS - Cascading Style Sheets

- **CSS** viết tắt của **Cascading Style Sheets**
- **CSS** Là một ngôn ngữ dùng để định dạng cho các phần tử HTML và bố cục của trang web.
- **CSS** giúp lập trình viên tiết kiệm công sức và thời gian
- **CSS** có thể tái sử dụng



CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

# Các khái niệm trong css

- ❑ **Property** là thuộc tính của phần tử HTML, ví dụ:
  - font-size (kích cỡ chữ của phần tử)
  - color (màu chữ của phần tử)
  - background-color (màu nền của phần tử)
- ❑ **Value** là giá trị của thuộc tính.
- ❑ **Selector** (bộ chọn) là phần tử mà bạn muốn định dạng.

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

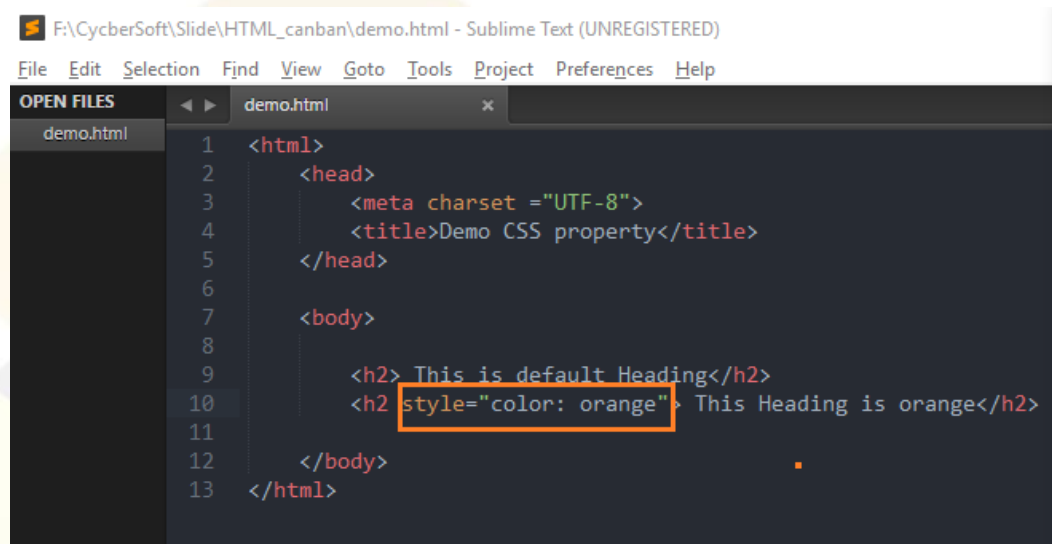
Có 3 loại:

- ❑ **Inline Style ( react-native thường dùng tương tự)**
- ❑ Internal Style Sheet
- ❑ External Style Sheet

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Thêm trực tiếp thuộc tính style vào thành phần HTML cần thay đổi mô tả

`<tagname style="property: value;... "> content... </tagname>`



```
1 <html>
2   <head>
3     <meta charset = "UTF-8">
4     <title>Demo CSS property</title>
5   </head>
6   <body>
7     <h2> This is default Heading</h2>
8     <h2 style="color: orange;"> This Heading is orange</h2>
9   </body>
10  </html>
```

This is default Heading

This Heading is orange

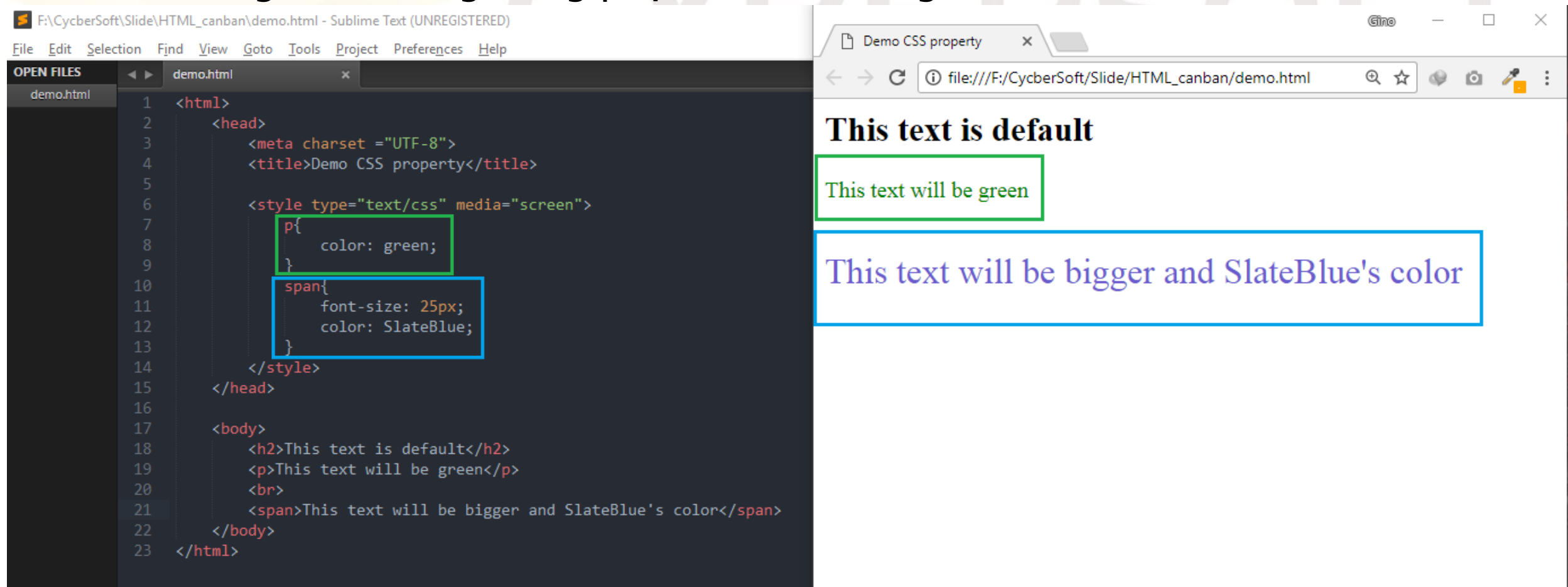
FT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



## Internal Style Sheet

- Còn được gọi là Embedding Style Sheet
- Thuộc tính CSS được định nghĩa trong **<style>**, nhúng trong **<head>** của trang HTML
- Có khả năng ảnh hưởng trong phạm vi toàn trang HTML



The image shows a side-by-side comparison of an HTML file in a code editor and its rendered output in a web browser.

**Code Editor (Sublime Text):** The file `demo.html` contains the following HTML code:

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Demo CSS property</title>
  <style type="text/css" media="screen">
    p{
      color: green;
    }
    span{
      font-size: 25px;
      color: SlateBlue;
    }
  </style>
</head>
<body>
  <h2>This text is default</h2>
  <p>This text will be green</p>
  <br>
  <span>This text will be bigger and SlateBlue's color</span>
</body>
</html>
```

**Web Browser (Ginno):** The browser displays the rendered HTML. The title is "Demo CSS property". The content shows three elements:

- A heading: "This text is default" (default black font).
- A paragraph: "This text will be green" (green text, highlighted with a green border).
- A span: "This text will be bigger and SlateBlue's color" (larger font size, SlateBlue color, highlighted with a blue border).

## External Style Sheet

- Còn được gọi là Linking Style Sheet
- Thuộc tính CSS được định nghĩa trong file **\*.css\***
- Được liên kết thông qua phần tử **<link>**, trong phần **<head>** của trang HTML
- Có khả năng ảnh hưởng đến nhiều trang HTML có liên kết đến file đó

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

# CSS - External Style sheet

F:\CycberSoft\Slide\HTML\_canban\demo.html - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

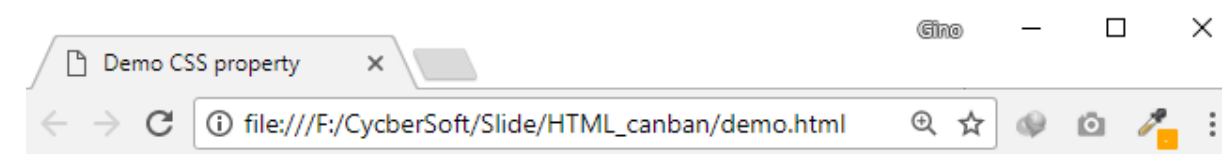
OPEN FILES

demo.html x demo.css x

```

1 <html>
2   <head>
3     <meta charset="UTF-8">
4     <title>Demo CSS property</title>
5     <link rel="stylesheet" href="demo.css" type="text/css">
6   </head>
7   <body>
8     <h2>This text is default</h2>
9     <p>This text will be white and green background</p>
10  </body>
11 </html>

```



This text is default

This text will be white and green background

F:\CycberSoft\Slide\HTML\_canban\demo.css - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES

demo.html x demo.css x

```

1 p{
2   background-color: lightgreen;
3   color: white;
4 }

```

## ❖ Các thuộc tính css thường được sử dụng

- Background-color: chỉnh màu nền cho phần tử
- Text: tinh chỉnh kiểu chữ, font, ...
- Font: tinh chỉnh kiểu chữ, kích thước, in đậm, in nghiêng, ...
- Padding: là khoảng từ đường viền của 1 phần tử tới phần tử nằm bên trong phần tử đó
- Margin: là khoảng cách giữa các phần tử html với nhau
- Display (flex, justify-content, flex-direction, ...): bố cục các phần layout trái phải responsive
- Position: dùng để căn chỉnh vị trí
- Border: viền của thẻ
- Color: màu text
- Width, height: độ rộng, dài của thẻ

## ❑ Một số thuộc tính background trong css:

- Background-color: Thiết lập màu nền
- Background-image: Thiết lập hình nền

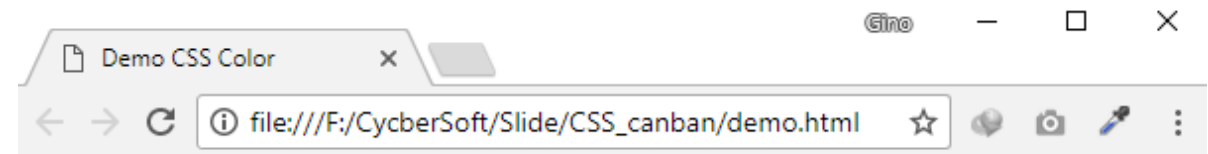
## ❑ Các thuộc tính background đi chung với background-image

- Background-repeat: Thiết lập xem hình nền có lặp lại hay không
- Background-position: Thiết lập vị trí hình nền

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



```
1 h2{
2   color: #ff0000;
3 }
4
5 div{
6   width: 400px;
7   height: 100px;
8 }
9 #div1{
10  background-color: #0000ff;
11 }
12 #div2{
13  background-color: #00ff00;
14 }
15 #div3{
16  background-color: #3c3c3c;
17 }
18 #div4{
19  background-color: #f5a4d1;
20 }
```



## Demo Color Name



## ❑ Một số thuộc tính css Text thường dùng:

- ✓ **Color** : Xét màu cho text
- ✓ **Text-decoration** : kiểu gạch chân cho text
- ✓ **Text-align**: Căn lề cho text
- ✓ **Line-height**: Xét chiều cao cho các dòng.
- ✓ **Word-spacing**: Khoảng cách giữa các từ trong đoạn văn
- ✓ **Letter-spacing**: Khoảng cách giữa các kí tự trong 1 t

CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

## ❑ Text-decoration:

- ✓ Thiết lập gạch chân cho text và kiểu gạch chân với các giá trị : none (không có) | underline(gạch chân) | overline(gạch trên) | line-through: gạch bỏ

```
h1{  
  text-decoration: none;  
}  
h2{  
  text-decoration: underline;  
}  
h3{  
  text-decoration: overline;  
}  
h4{  
  text-decoration: line-through;  
}
```

Cybersoft academy

Cybersoft academy

Cybersoft academy

~~Cybersoft academy~~

## ❑ Line-height:

- ✓ Thiết lập chiều cao cho các dòng.

```
h1{  
  line-height: 100px;  
}
```

Cybersoft academy

Cybersoft academy

Cybersoft academy

Cybersoft academy

T  
H

## ❑ Word-spacing:

✓ Thiết lập khoảng cách giữa các từ trong đoạn văn bản.

```
h1{  
  word-spacing:10px;  
}  
h2{  
  word-spacing: -10px;  
}
```

Cybersoft academy

Cybersoftacademy

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



## ❑ Letter-spacing:

- ✓ Thiết lập khoảng cách giữa các kí tự

```
h1{  
  letter-spacing:10px;  
}  
h2{  
  letter-spacing: -10px;  
}
```

C y b e r s o f t a c a d e m y

Cyber

CYBERSOFT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

## ❑ Một số thuộc tính CSS Fonts thường dùng:

- **font-size:** thiết lập kích cỡ chữ
- **font-weight:** thiết lập sự tô đậm của chữ
- **font-style:** thiết lập sự in nghiêng của chữ
- **font-family:** thiết lập kiểu chữ



CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

## ❑ Font-size:

✓ Thiết lập kích thước cho chữ

✓ Các đơn vị cơ bản :

- Px
- Rem,em : kích thước thay đổi theo kích thước chuẩn của html(root).

```
.text-1{  
  font-weight: bold;  
}  
.text-2{  
  font-weight: normal;  
}
```

**Lorem ipsum dolor sit amet.**

Lorem ipsum dolor sit amet.

## ❑ Font-weight:

✓ Thiết lập mức độ đậm nhạt cho text.

✓ 2 giá trị cơ bản :

- **Normal | bold**
- **100 - 900**

```
.text-1{  
  font-weight: bold;  
}  
.text-2{  
  font-weight: normal;  
}
```

**Lorem ipsum dolor sit amet.**

Lorem ipsum dolor sit amet.

## ❑ Font-style:

✓ Thiết lập độ nghiêng cho text.

✓ Các giá trị cơ bản :

▪ **Normal | italic**

```
.text-1{  
  font-style: italic;  
}  
.text-2{  
  font-size: normal;  
}
```

*Lorem ipsum dolor sit amet.*

Lorem ipsum dolor sit amet.

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

## ❑ Font-family:

- ✓ Thiết lập font chữ cho text.
- ✓ Một số font chính : Times New Roman, Arial,...
- ✓ Có ba nhóm font chữ cơ bản là:
  - Serif
  - Sans-Serif
  - Monospace

CYBERSOFT

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

## ❑ Font-family: một số font chữ mặc định.

| Font chữ                                      | Ví dụ                                           |
|-----------------------------------------------|-------------------------------------------------|
| Georgia                                       | <b>This is a heading</b><br>This is a paragraph |
| Palatino Linotype<br>Book Antiqua<br>Palatino | <b>This is a heading</b><br>This is a paragraph |
| Times New Roman<br>Times                      | <b>This is a heading</b><br>This is a paragraph |
| serif                                         | <b>This is a heading</b><br>This is a paragraph |

| Font chữ                 | Ví dụ                                           |
|--------------------------|-------------------------------------------------|
| Courier New<br>Courier   | <b>This is a heading</b><br>This is a paragraph |
| Lucida Console<br>Monaco | <b>This is a heading</b><br>This is a paragraph |
| monospace                | <b>This is a heading</b><br>This is a paragraph |

| Font chữ                             | Ví dụ                                                  |
|--------------------------------------|--------------------------------------------------------|
| Arial<br>Helvetica                   | <b>This is a heading</b><br>This is a paragraph        |
| Arial Black<br>Gadget                | <b>This is a heading</b><br><b>This is a paragraph</b> |
| Comic Sans MS<br>cursive             | <b>This is a heading</b><br>This is a paragraph        |
| Impact<br>Charcoal                   | <b>This is a heading</b><br>This is a paragraph        |
| Lucida Sans Unicode<br>Lucida Grande | <b>This is a heading</b><br>This is a paragraph        |
| Tahoma<br>Geneva                     | <b>This is a heading</b><br>This is a paragraph        |
| Trebuchet MS<br>Helvetica            | <b>This is a heading</b><br>This is a paragraph        |
| Verdana<br>Geneva                    | <b>This is a heading</b><br>This is a paragraph        |
| sans-serif                           | <b>This is a heading</b><br>This is a paragraph        |



## ❑ Font-family: Ví dụ

```
.text-1{  
  font-family: "Times New Roman";  
}  
.text-2{  
  font-family: sans-serif;  
}
```

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

❖ Padding là khoảng cách từ đường viền của phần tử tới nội dung bên trong nó.

❖ Một số thuộc tính của padding:

- padding-top (vùng đệm phía trên)
- padding-right (vùng đệm phía bên phải)
- padding-bottom (vùng đệm phía dưới)
- padding-left (vùng đệm phía bên trái)

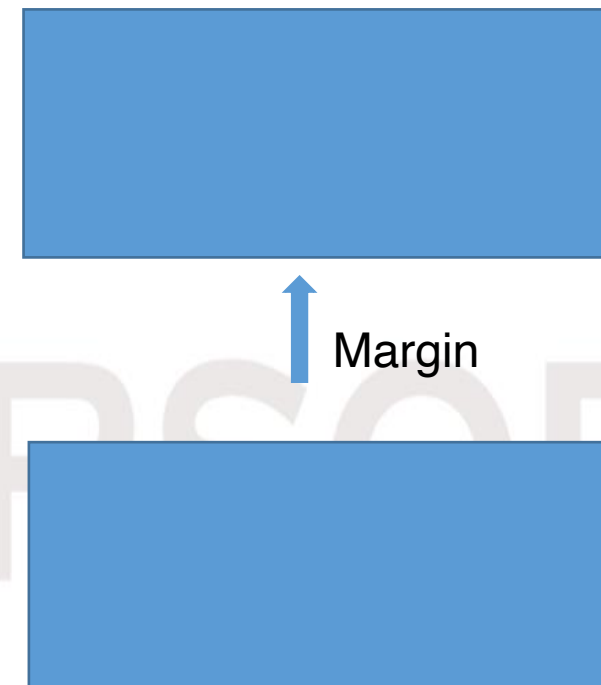


## ❖ Ví dụ:

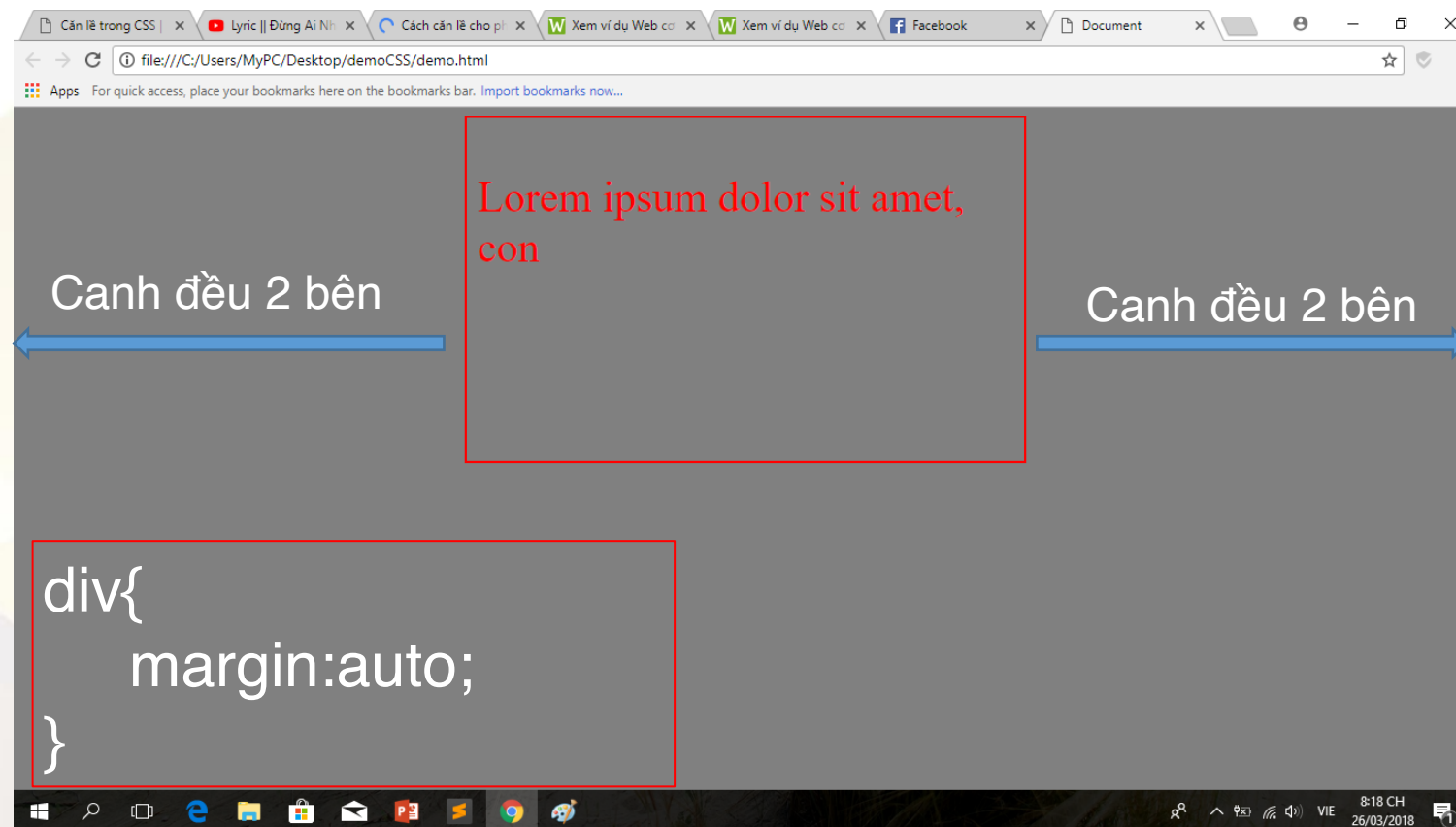
```
/* Cách 1 */
div{
    padding-top:10px;
    padding-right:10px;
    padding-bottom:10px;
    padding-left:10px;
}
/* Cách 2 */
div{
    padding:20px; /* Chỉnh padding cho cả 4 phía */
}
/* Cách 3 */
div{
    padding:20px 30px; /* Giá trị đầu tiên đại diện cho top bottom
                       Giá trị thứ 2 đại diện cho left right */
}
/* Cách 4 */
div{
    padding:10px 20px 25px 30px; /* Chỉnh padding cho cả 4 phía theo chiều kim đồng hồ */
}
```

FT  
RÌNH

- ❖ Thuộc tính margin dùng để canh lề cho phần tử.
- ❖ Một số thuộc tính của margin
  - margin-top (canh lề phía trên)
  - margin-right (canh lề bên phải)
  - margin-bottom (canh lề phía dưới)
  - margin-left (canh lề bên trái)
- ❖ Cách viết giống như padding, thay padding thành margin



- ❖ Thuộc tính margin với giá trị auto.



- ❖ Thuộc tính `display` chỉ định phần tử phải hiển thị như thế nào :
  - Block: hiển thị thành một khối, mỗi khối 1 dòng
  - Inline: hiển thị nội tuyến, các phần tử nằm trên 1 dòng
  - None: ẩn phần tử.

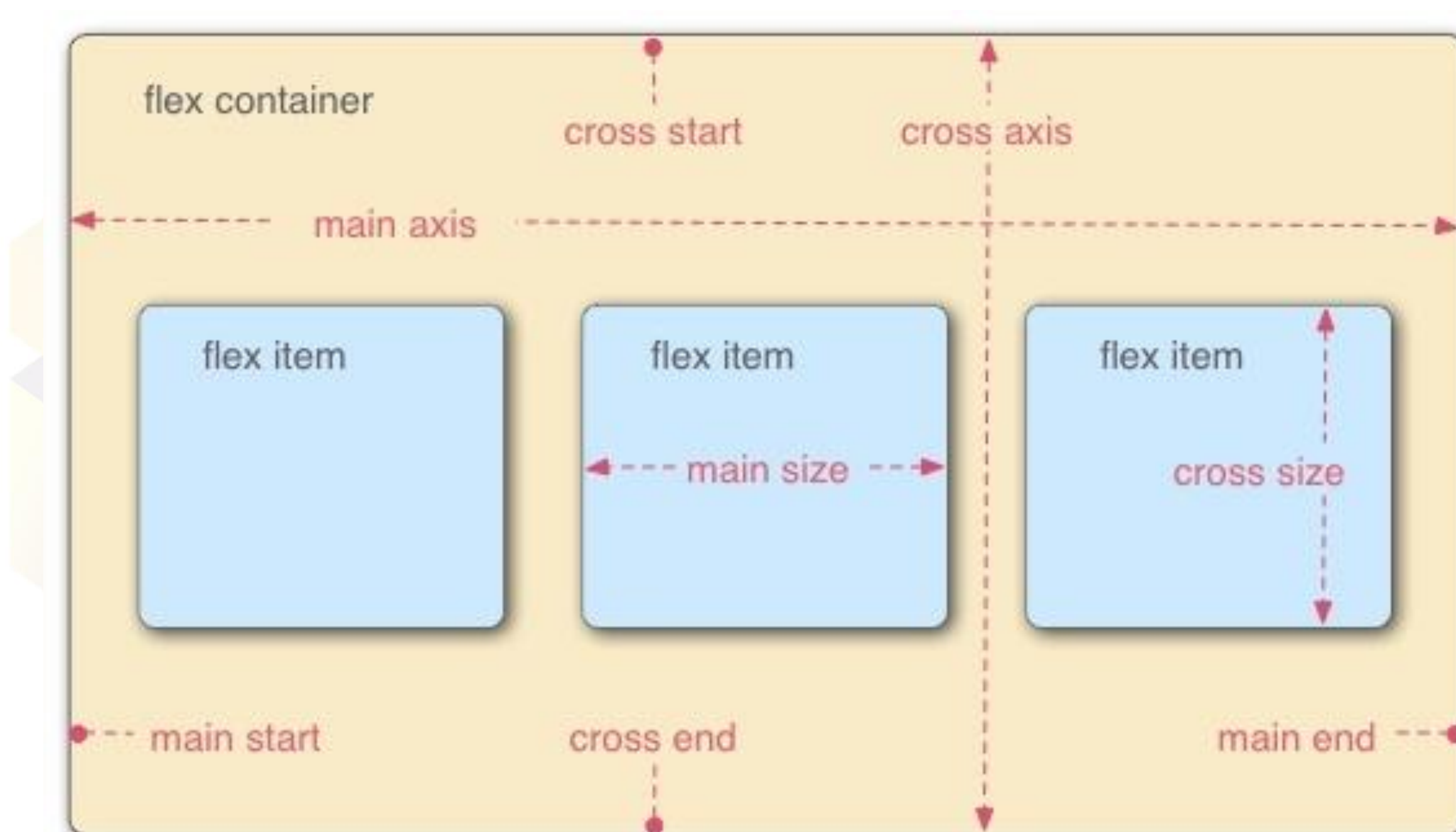
Canh đều 2 bên

```
.demo{  
  display:block | inline | inline-block | none | unset;  
}
```

- ❖ Ngoài thuộc tính display :none còn một số thuộc tính giúp chúng ta ẩn phần tử.
  - visibility: hidden;
  - Opacity:0;
  - 2 thuộc tính này chỉ giúp ẩn phần tử đi về mặt hình thức...phần tử vẫn chiếm chỗ trên giao diện

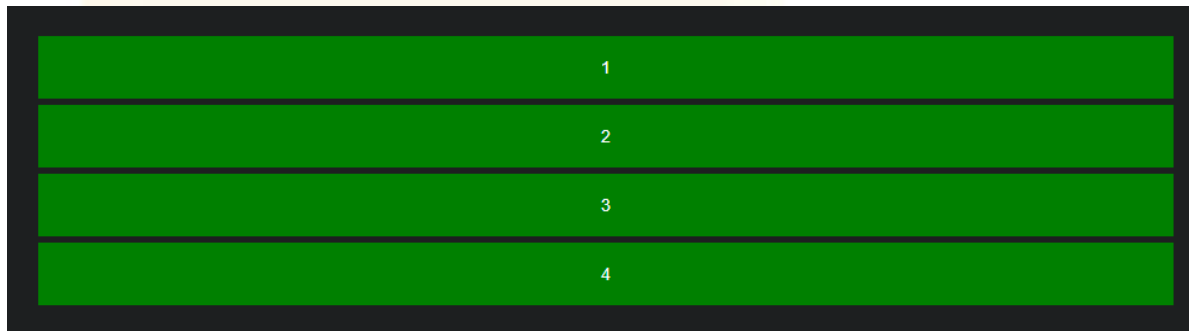
CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH





## \*display : flex

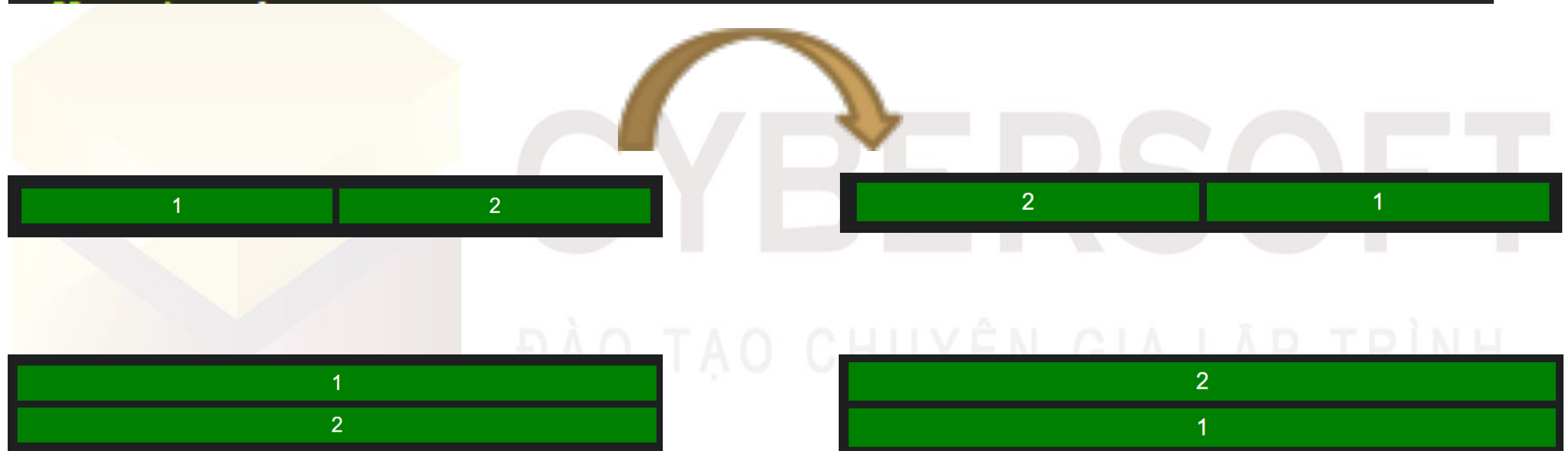
```
.flex-container{  
  display: flex;  
}
```



ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

\*flex-direction: dùng để đổi hướng sắp xếp của các phần tử

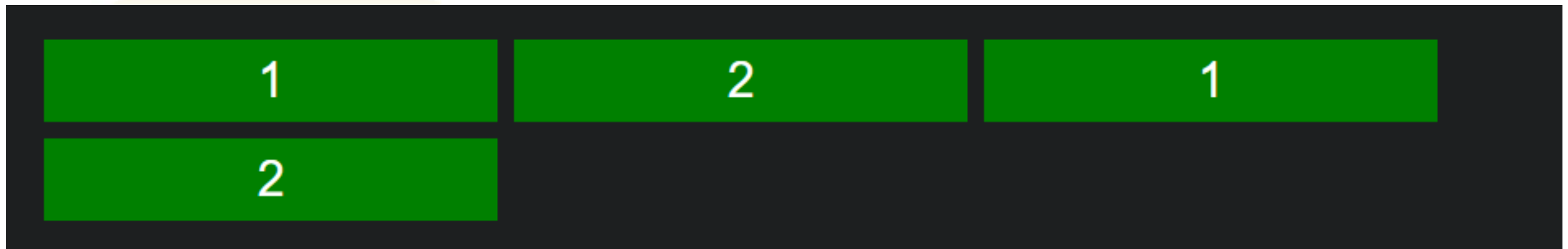
```
.flex-container{  
  flex-direction: row | column | row-reverse | column-reverse;  
}
```



# CSS - Display (FlexBox - kiến thức quan trọng)

\*flex-wrap: cho phép các phần tử xuống hàng khi không đủ chỗ chứa hay không

```
.flex-container{  
  flex-wrap: wrap | nowrap | wrap-reverse;  
}
```



ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

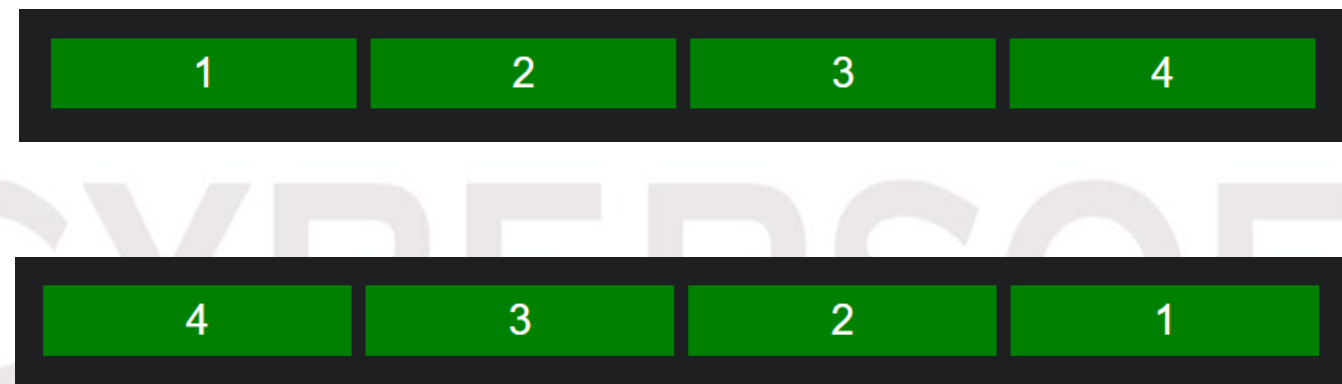
\*justify-content: căn chỉnh vị trí phần tử theo chiều ngang Align-item: căn chỉnh vị trí phần tử theo chiều dọc.

```
.flex-container{  
  justify-content:flex-start | flex-end | center;  
}
```



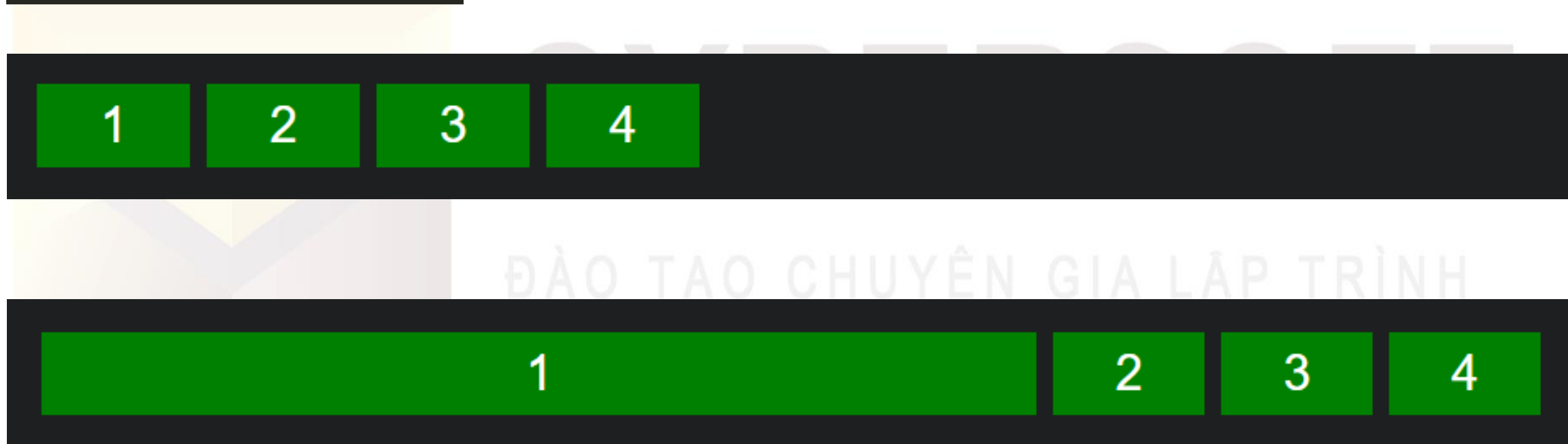
## \* Order: sắp xếp thứ tự các phần tử trong flexbox

```
.item1{  
  order:4;  
}  
.item2{  
  order:3;  
}  
.item3{  
  order:2;  
}  
.item4{  
  order:1;  
}
```



\*flex-grow: item được khai báo thuộc tính này sẽ chiếm toàn bộ diện tích còn trống trong flexbox

```
.item1{  
  flex-grow: 1;  
}
```



127.0.0.1:5500/index.html

iPhone 6/7... 414 x 736 75% Online

Elements Console Sources Network

```
<!doctype html>
<html lang="en">
  <head>...</head>
  <body>
    <div style="background:rgb(235, 233, 233)">
      <span class="text">Nike</span>
      <div class="flex-container">
        <div class="flex-item">...</div>
        <div class="flex-item">...</div>
        <div class="flex-item">...</div>
        <div class="flex-item">...</div>
        <div class="flex-item">...</div>
        <div class="flex-item">...</div>
        <div class="flex-item">...</div>
        <div class="flex-item">...</div>
      </div>
    </div>
    <!-- Code injected by live-server -->
    <script type="text/javascript">...</script>
  </body>
</html>
```

html body div div.flex-container div.flex-item

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element.style {

.flex-item { styles.css:15

border-radius: 10px;

background: #fff;

padding: 5%;

margin: 2%;

width: 36%;

div { user agent stylesheet

display: block;

Inherited from html

background-attachment

7.547

border -

padding 18.891

18.891 136.078 x 251 18.891

18.891

7.547

Filter Show all