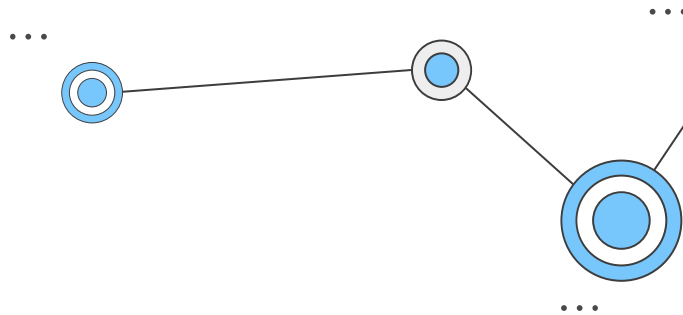
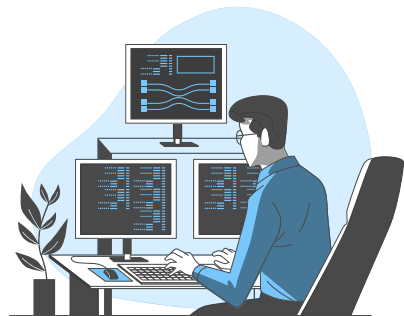


CSS



Khóa học Backend

Bài 03: Học CSS cơ bản



Nội dung

01

Khái niệm, cú pháp,
selectors

...

02

Simple selectors
(Bộ chọn đơn giản)

...

03

Ba kiểu chèn CSS

...

04

Colors, Backgrounds

...

05

Box Model, Borders,
Padding, Margins

...

06

Text, Fonts, Icons

...

07

Display (Hiển thị)

...

08

Combinator selectors
(Bộ chọn tổ hợp)

...

09

Position (Vị trí)

...

10

z-index

...



01

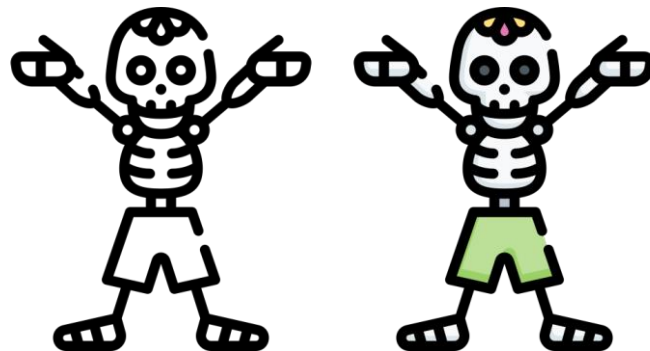
**Khái niệm, cú pháp,
selectors**

01. Khái niệm, cú pháp, selectors

- **Khái niệm:**
 - **CSS:** viết tắt của **C**ascading **S**tyle **S**heets.
 - Được dùng để **làm đẹp HTML**.
- **Cú pháp:**

```
selector {  
  property: value;  
}
```

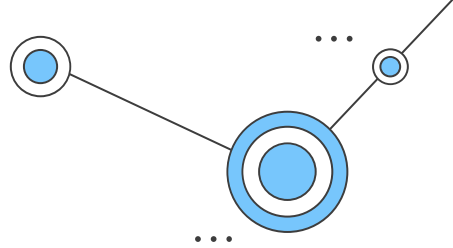
- **Trong đó:**
 - **selector:** Được gọi là bộ chọn.
 - **property:** Được gọi là thuộc tính.
 - **value:** Được gọi là giá trị của thuộc tính.



HTML

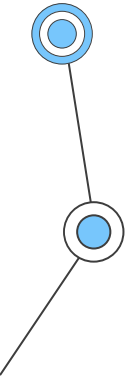
CSS

01. Khái niệm, cú pháp, selectors

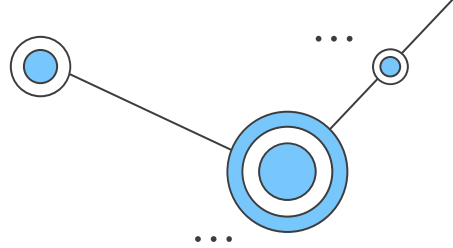


- **Selectors (Bộ chọn):**
 - Bộ chọn CSS chọn (các) phần tử HTML mà bạn muốn tạo kiểu.
 - Ví dụ: Các thẻ <h1>, <p>, <table>,... thì viết trong css sẽ chỉ lấy tên thẻ là h1, p, table.

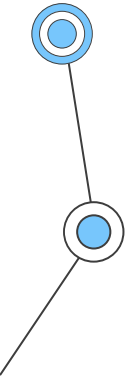
```
h1 {  
  /* code css */  
}  
  
p {  
  /* code css */  
}  
  
table {  
  /* code css */  
}
```



01. Khái niệm, cú pháp, selectors



- **Selectors (Bộ chọn):**
 - Bộ chọn CSS được chia thành 5 loại:
 - **Simple selectors (Bộ chọn đơn giản)**
 - **Combinator selectors (Bộ chọn tổ hợp)**
 - Pseudo-class selectors (Bộ chọn lớp giả)
 - Pseudo-elements selectors (Bộ chọn phần tử giả)
 - Attribute selectors (Bộ chọn thuộc tính)



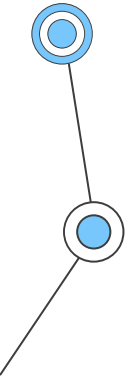
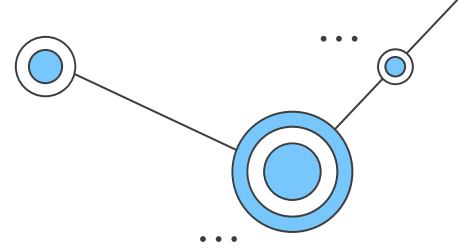
02

Simple selectors (Bộ chọn đơn giản)

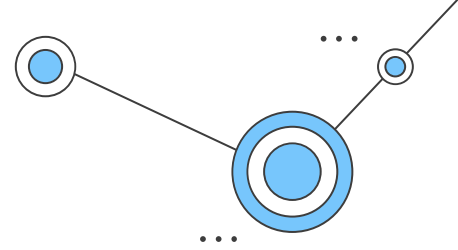
02. Simple selectors (Bộ chọn đơn giản)

- **Bộ chọn element (element selector):**
 - Chọn các phần tử dựa trên **tên phần tử**.
 - Ví dụ:

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

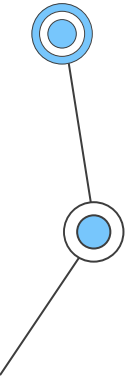


02. Simple selectors (Bộ chọn đơn giản)

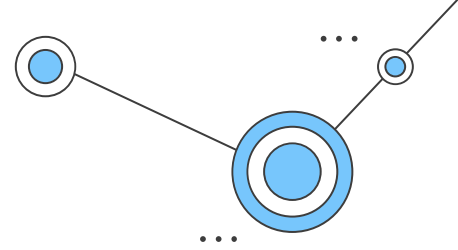


- **Bộ chọn id (id selector):**
 - Chọn các phần tử dựa trên **thuộc tính id**.
 - Ví dụ:

```
#title {  
  text-align: center;  
  color: red;  
}
```

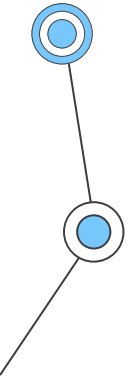


02. Simple selectors (Bộ chọn đơn giản)



- **Bộ chọn class (class selector):**
 - Chọn các phần tử dựa trên **thuộc tính class**.
 - Ví dụ:

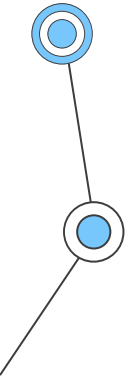
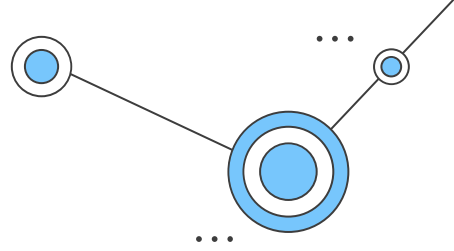
```
.title {  
  text-align: center;  
  color: red;  
}
```



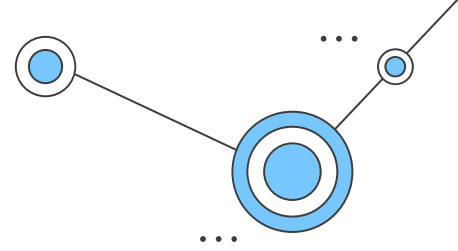
02. Simple selectors (Bộ chọn đơn giản)

- **Bộ chọn chung (universal selector):**
 - **Chọn tất cả** các phần tử.
 - Ví dụ:

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

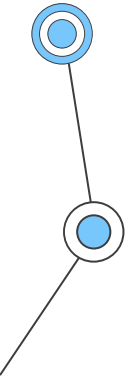


02. Simple selectors (Bộ chọn đơn giản)



- **Bộ chọn nhóm (grouping selector):**
 - Chọn **tất cả** các **phần tử**, **id**, **class** có cùng thuộc tính và giá trị thuộc tính.
 - Ví dụ:

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```



03

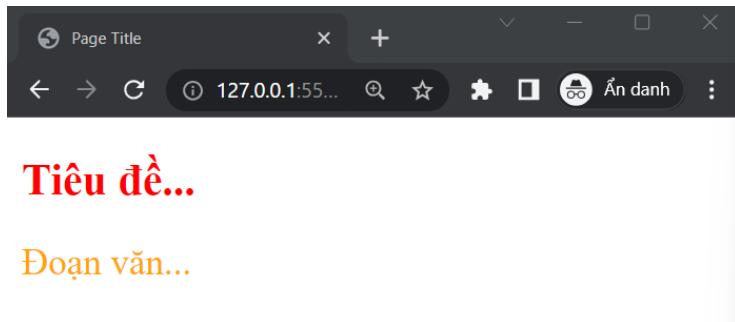
Ba kiểu chèn CSS

03. Ba kiểu chèn CSS

- **Inline CSS (Nội tuyến):**

- Thêm thuộc tính **style** vào trong phần tử mà muốn sửa CSS.
- Chỉ nên dùng nếu chỉ có **một phần tử** sử dụng đoạn CSS đó.
- Ví dụ:

```
<h2 style="color:red;">Tiêu đề...</h2>  
<p style="color: orange; font-size: 20px">Đoạn văn...</p>
```



03. Ba kiểu chèn CSS

- **Internal CSS (Nội bộ):**

- Thêm thẻ **<style></style>** và bất cứ đâu trong file html.
- Sau đó viết CSS vào trong thẻ **<style>**.
- Chỉ nên dùng nếu chỉ có **một trang** sử dụng đoạn CSS đó.
- Ví dụ:

```
<style>
  h2 {
    color: red;
  }
</style>

<h2>Tiêu đề 1</h2>
<h2>Tiêu đề 2</h2>
<h2>Tiêu đề 3</h2>
<h2>Tiêu đề 4</h2>
<h2>Tiêu đề 5</h2>
```

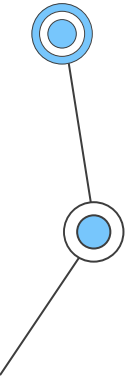
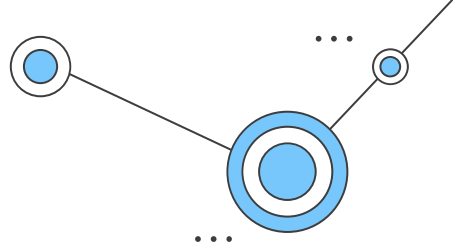
Tiêu đề 1

Tiêu đề 2

Tiêu đề 3

Tiêu đề 4

Tiêu đề 5

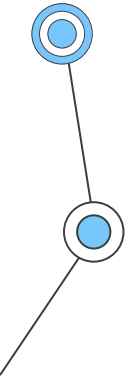
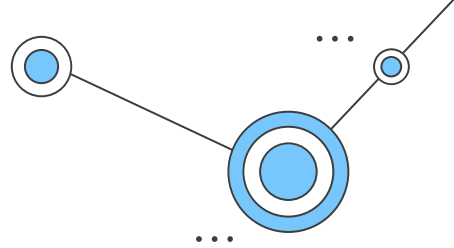


03. Ba kiểu chèn CSS

- **External CSS (Bên ngoài):**

- Thêm một file CSS, sau đó dùng thẻ **<link />** đặt ở trong **<head></head>** và chèn đường dẫn file vào thuộc tính **href**.
- Áp dụng cho **nhiều trang** cần CSS chung.
- Ví dụ:

```
<head>  
  <title>Page Title</title>  
  <link rel="stylesheet" href="style.css">  
</head>
```





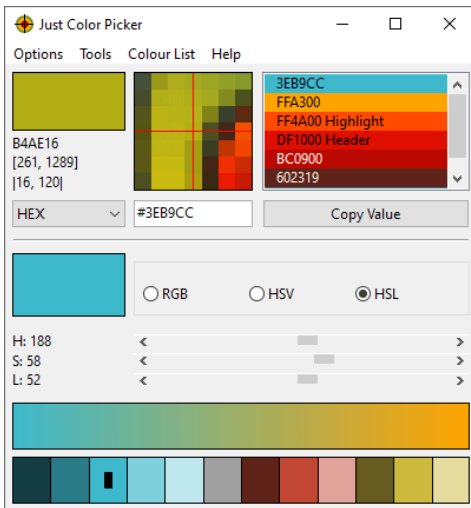
04

Colors, Backgrounds



04. Colors, Backgrounds

- Cài đặt phần mềm **Just Color Picker**
 - Dùng để lấy mã màu nhanh.
 - Link cài đặt: <https://annystudio.com/software/colorpicker/>
 - Cách dùng:
 - Bước 1: Di chuột vào vị trí màu muốn lấy
 - Bước 2: Bấm **Alt + X** để lấy mã màu



04. Colors, Backgrounds

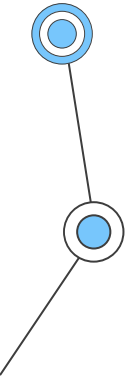
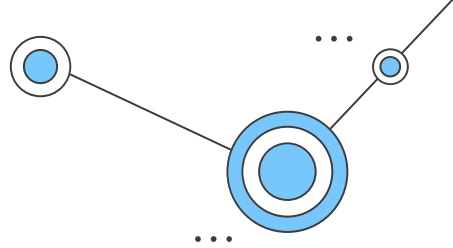
- **Colors (màu sắc):**
 - Có 2 cách hay sử dụng là:
 - **Đặt tên sẵn:** white, black, red, green, blue, yellow, orange,...
 - **HEX:**
 - Cú pháp: **#RRGGBB**
 - Trong đó:
 - RR: Red (đỏ)
 - GG: Green (xanh lục)
 - BB: Blue (xanh dương)
 - Ví dụ:
 - #000000 (màu đen)
 - #FFFFFF (màu trắng)
 - #FF0000 (màu đỏ)

```
h2 {  
  color: #FF0000;  
}
```

04. Colors, Backgrounds

- **Background (nền):**
 - Dùng để thêm nền vào cho một phần tử.
 - Tên thuộc tính: **background**
 - Ví dụ:

```
h2 {  
  color: #FF0000;  
  background: #000000;  
}
```

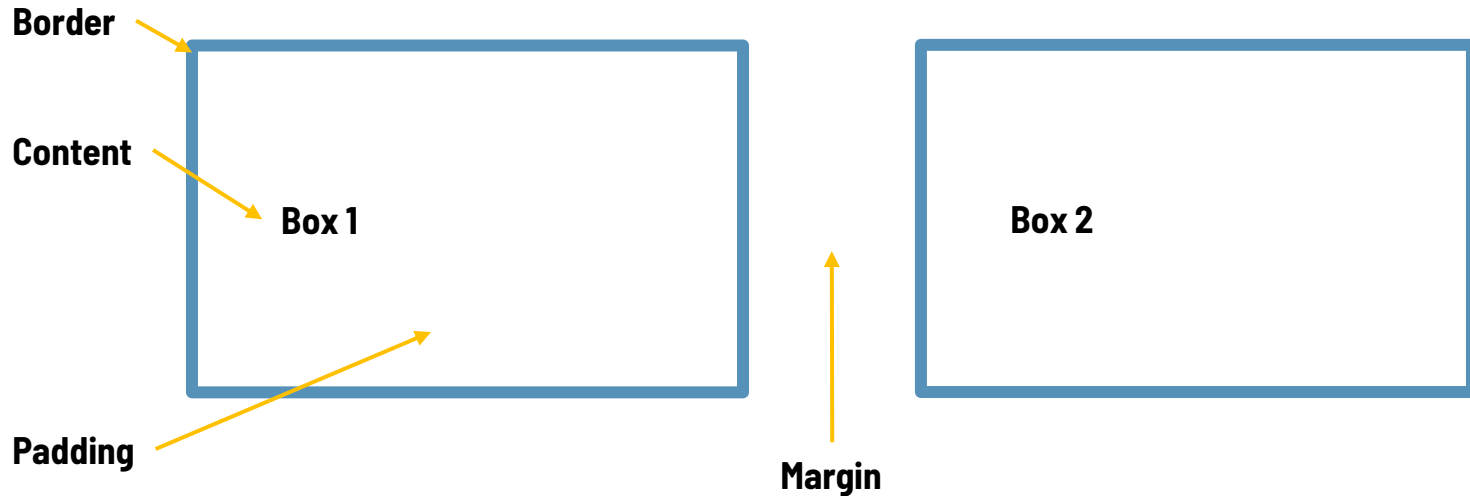


05

Box Model, Content, Padding, Border, Margin

5.1. Box Model (Mô hình hộp)

- Một phần tử trong HTML có thể được coi là một cái **hộp**.
- Hộp này bao gồm có 4 thành phần:
 - Content (Nội dung) - Là phần nội dung bên trong hộp.
 - Border (Đường viền): Đường viền của hộp.
 - Padding (Phần đệm) - Là khoảng cách giữa nội dung và đường viền.
 - Margin (Lề) - Là khoảng cách giữa các hộp.



5.2. Borders (Đường viền)

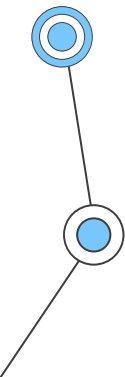
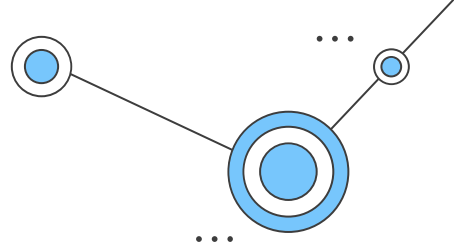
- Là đường viền của hộp.
- Cú pháp:

```
selector {  
  border: [border-width] [border-style] [border-color];  
}
```

- Trong đó:
 - **border-width**: Độ dày của đường viền.
 - **border-style**: Kiểu của đường viền.
 - **dotted**: Đường viền chấm
 - **dashed**: Đường viền nét đứt
 - **solid**: Đường viền liền
 - **border-color**: Màu của đường viền.

```
div {  
  border: 5px solid black;  
}
```

Box 1



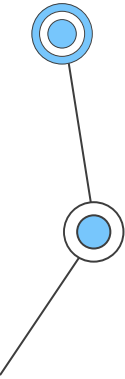
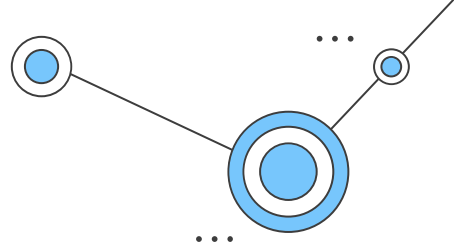
5.2. Borders (Đường viền)

- Cú pháp **bo góc** đường viền:

```
selector {  
  border-radius: giá trị;  
}
```

```
div {  
  border: 5px solid black;  
  border-radius: 8px;  
}
```

Box 1

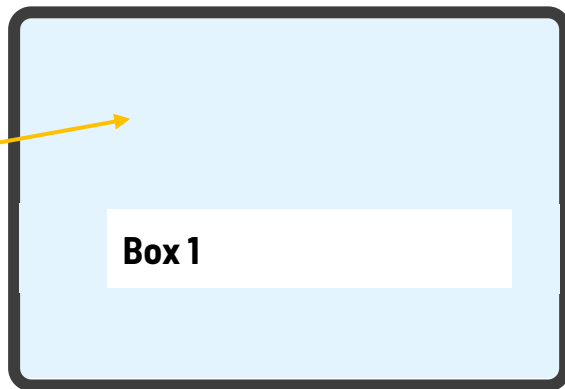


5.3. Padding (Phần đệm)

- Là khoảng cách giữa nội dung và đường viền.
- Cú pháp:

```
selector {  
  padding: top right bottom left;  
}
```

```
div {  
  border: 5px solid black;  
  border-radius: 5px;  
  padding: 100px 20px 50px 50px;  
}
```

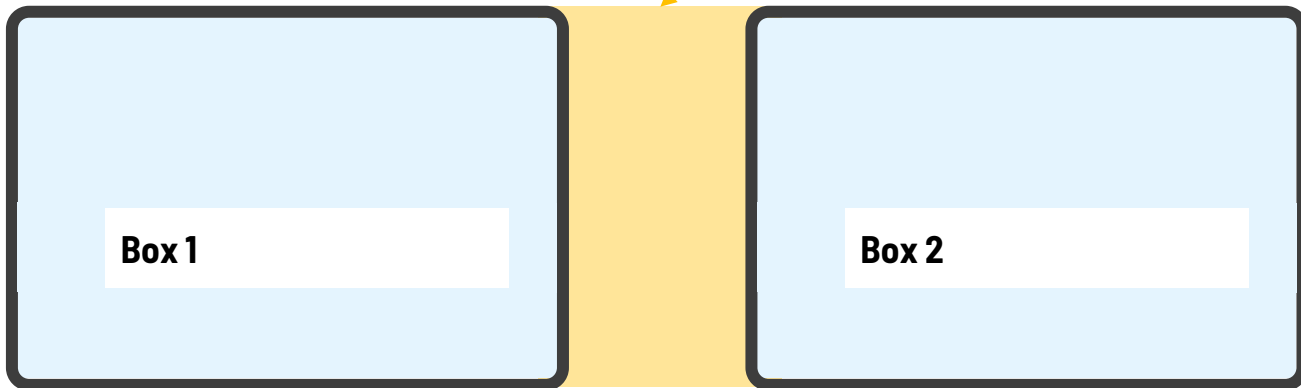


5.4. Margin (Lề)

- Là khoảng cách giữa các hộp.
- Cú pháp

```
selector {  
  margin: top right bottom left;  
}
```

```
div {  
  border: 5px solid black;  
  border-radius: 5px;  
  padding: 15px 20px 10px 5px;  
  margin: 0px 80px 0px 0px;  
}
```





06

Text, Fonts, Icons

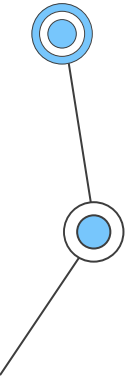
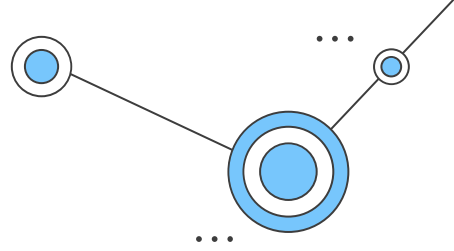


6.1. Text

- **color:**
 - Dùng để đặt màu cho chữ.

```
h1 {  
  color: #5691B9;  
}
```

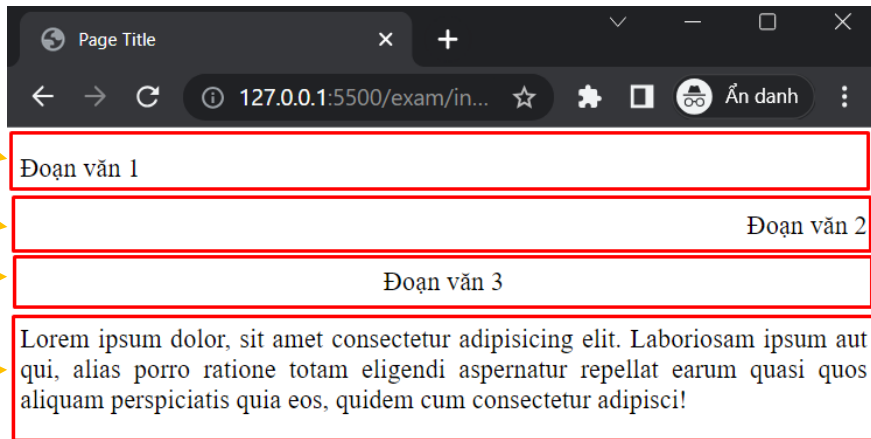
Tiêu đề 1



6.1. Text

- **text-align:**
 - Dùng để thiết lập căn lề ngang cho văn bản.
 - **left:** Căn trái
 - **right:** Căn phải
 - **center:** Căn giữa
 - **justify:** Căn đều 2 bên

```
.text-1 {  
  text-align: left;  
}  
.text-2 {  
  text-align: right;  
}  
.text-3 {  
  text-align: center;  
}  
.text-4 {  
  text-align: justify;  
}
```

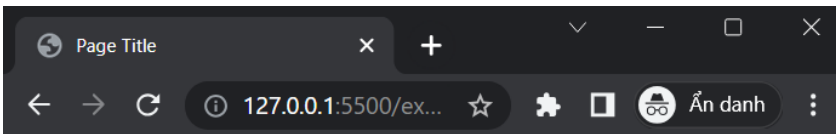


6.1. Text

- **text-transform:**

- Dùng để chỉ định chữ hoa hoặc chữ thường trong văn bản.
 - **uppercase:** VIẾT HOA TẤT CẢ
 - **capitalize:** Viết Hoa Các Chữ Đầu
 - **lowercase:** viết thường tất cả

```
.text-1 {  
  text-transform: uppercase;  
}  
  
.text-2 {  
  text-transform: capitalize;  
}  
  
.text-3 {  
  text-transform: lowercase;  
}
```



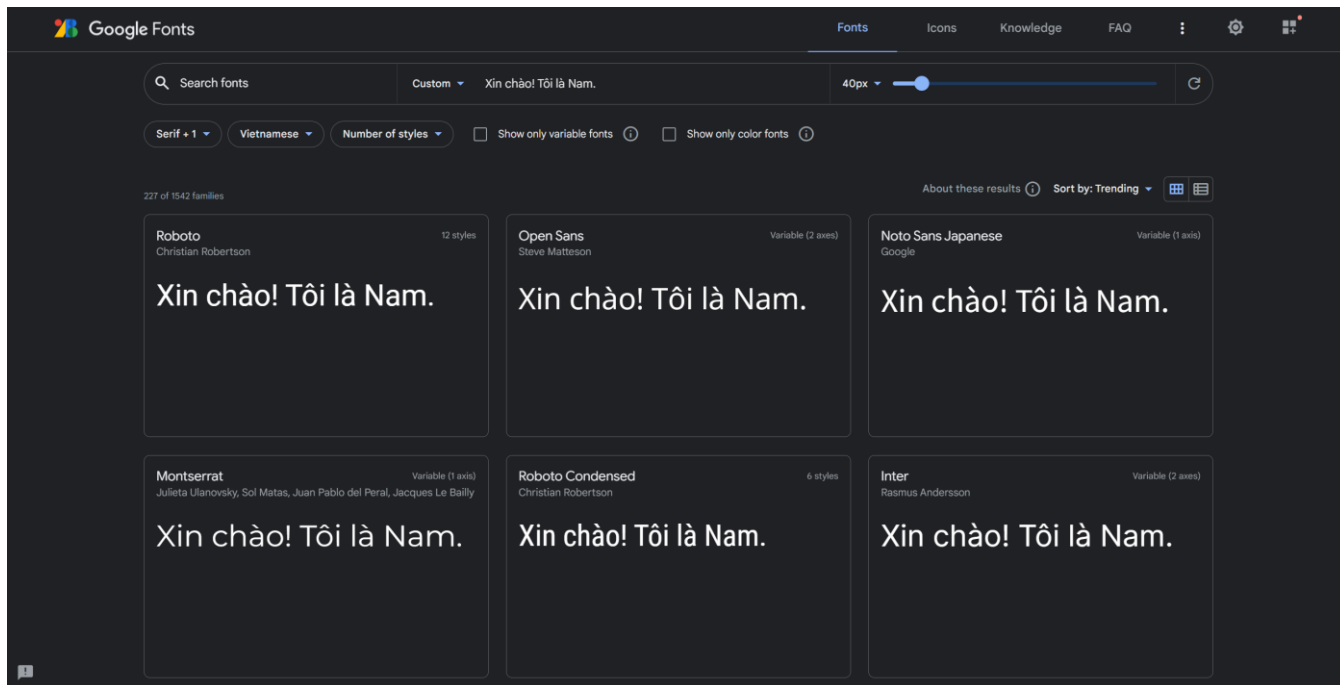
ĐOẠN VĂN 1

Đoạn Văn 2

đoạn văn 3

6.2. Fonts

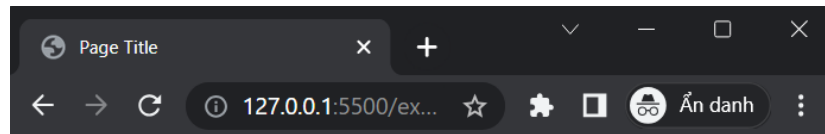
- **font-family:**
 - Để chỉ định font chữ của văn bản.
 - Trang google fonts: <https://fonts.google.com>



6.2. Fonts

- **font-style:**
 - Dùng để chỉ định văn bản in nghiêng.
 - **normal:** Văn bản được hiển thị bình thường.
 - **italic:** Văn bản được hiển thị in nghiêng.

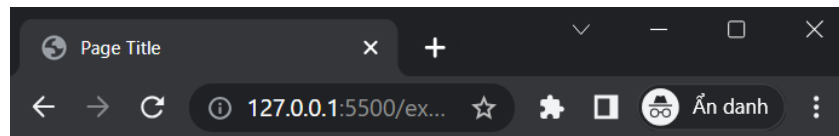
```
.text-1 {  
  font-style: normal;  
}  
  
.text-2 {  
  font-style: italic;  
}
```



6.2. Fonts

- **font-weight:**
 - Dùng để chỉ định độ dày của chữ.
 - Giá trị có thể là: normal, bold, 100, 200, 300,..., 800,900.

```
.text-1 {  
  font-weight: 400;  
}  
.text-2 {  
  font-weight: 600;  
}  
.text-3 {  
  font-weight: 900;  
}
```



Đoạn văn 1

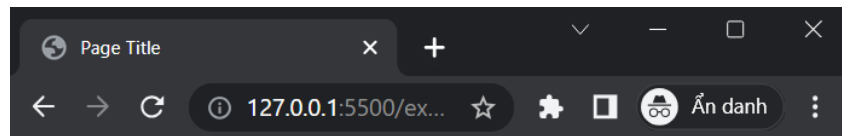
Đoạn văn 2

Đoạn văn 3

6.2. Fonts

- **font-size:**
 - Dùng để chỉnh kích cỡ của chữ.

```
.text-1 {  
  font-size: 12px;  
}  
.text-2 {  
  font-size: 16px;  
}  
.text-3 {  
  font-size: 20px;  
}
```



Đoạn văn 1

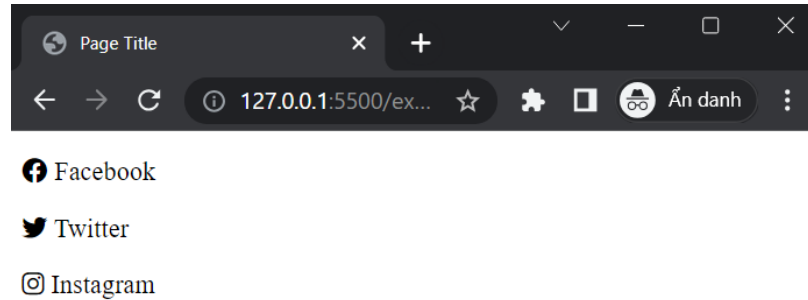
Đoạn văn 2

Đoạn văn 3

6.3. Icons

- Link trang chủ: <https://fontawesome.com/search?o=r&m=free>
- Link web để lấy mã nhúng: <https://cdnjs.com/libraries/font-awesome>

```
<p><i class="fa-brands fa-facebook"></i> Facebook</p>
<p><i class="fa-brands fa-twitter"></i> Twitter</p>
<p><i class="fa-brands fa-instagram"></i> Instagram</p>
```



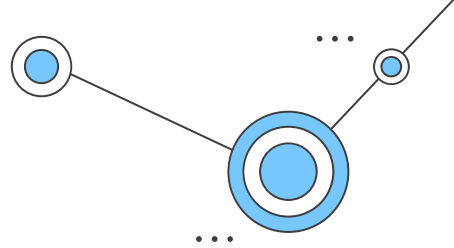


07

Display (Hiển thị)



7.1. Thuộc tính display



- **display: inline;**
 - Không đặt được chiều rộng và chiều cao.
 - Không thêm được margin và padding cho bên trên và bên dưới.

Phần tử 1

Phần tử 2

- **display: block;**
 - Chiều rộng kéo dài từ trái sang phải, luôn bắt đầu ở một hàng mới.
 - Cho phép đặt chiều rộng và chiều cao.
 - Cho phép đặt margin, padding cho bên trên và bên dưới.

Phần tử 1

Phần tử 2

- **display: inline-block;**
 - Chiều rộng mặc định bằng chiều rộng của phần tử.
 - Không bắt đầu ở một hàng mới.
 - Cho phép đặt chiều rộng và chiều cao.
 - Cho phép đặt margin, padding cho bên trên và bên dưới.

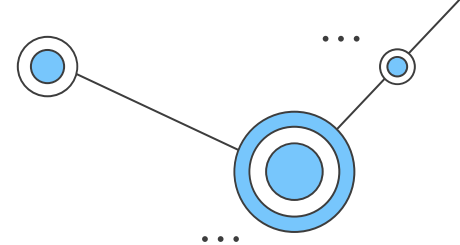
Phần tử 1

Phần tử 2

- **display: none;**
 - Ẩn phần tử hoàn toàn



7.2. Thuộc tính visibility



- **visibility: visible;**

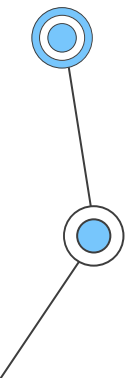
- Mặc định. Phần tử luôn được hiển thị.

Phần tử 1

Phần tử 2

- **visibility: hidden;**

- Ẩn nhưng vẫn chiếm diện tích của phần tử đó.

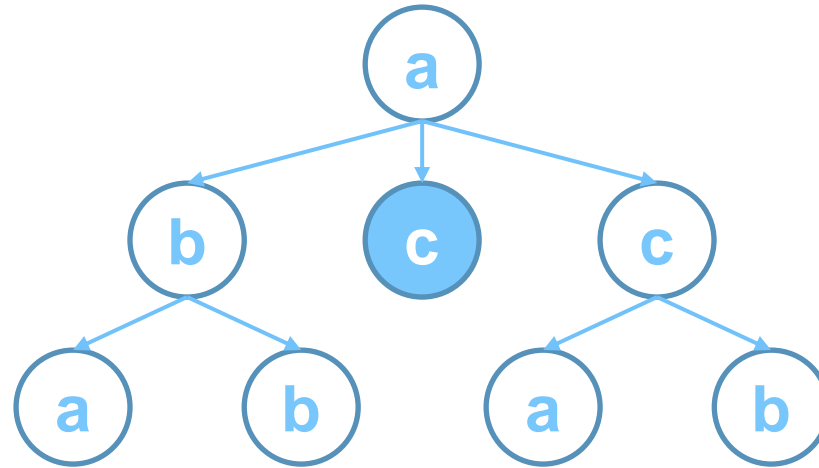


08

Combinator selectors (Bộ chọn tổ hợp)

08. Combinator selectors (Bộ chọn tổ hợp)

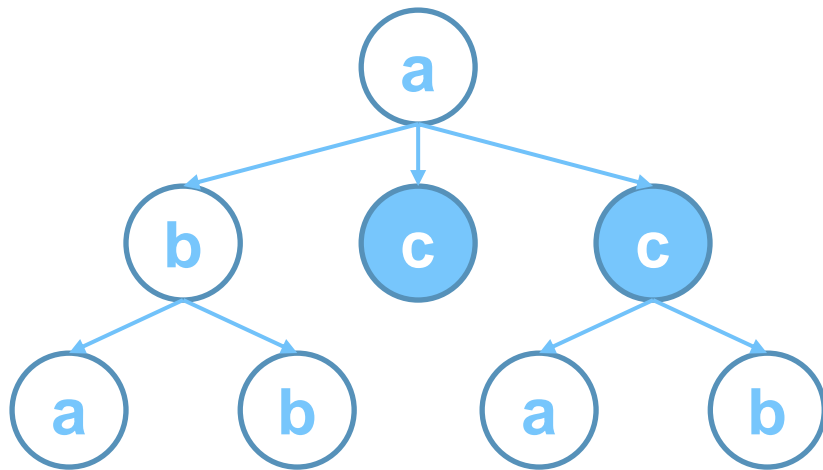
- **Adjacent sibling selectors (Bộ chọn anh chị em liền kề)**
 - Chọn phần tử **c** được đặt cùng cấp và phải **kề ngay sau** phần tử **b**.
 - Cú pháp: **b + c { }**



08. Combinator selectors (Bộ chọn tổ hợp)

- **General sibling selectors (Bộ chọn anh chị em chung)**

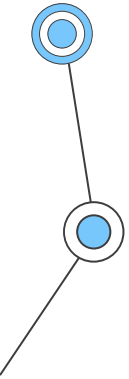
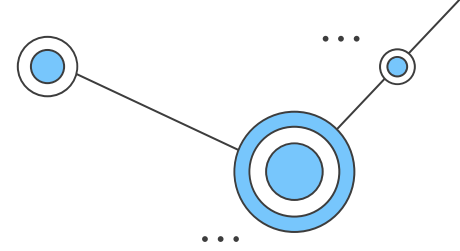
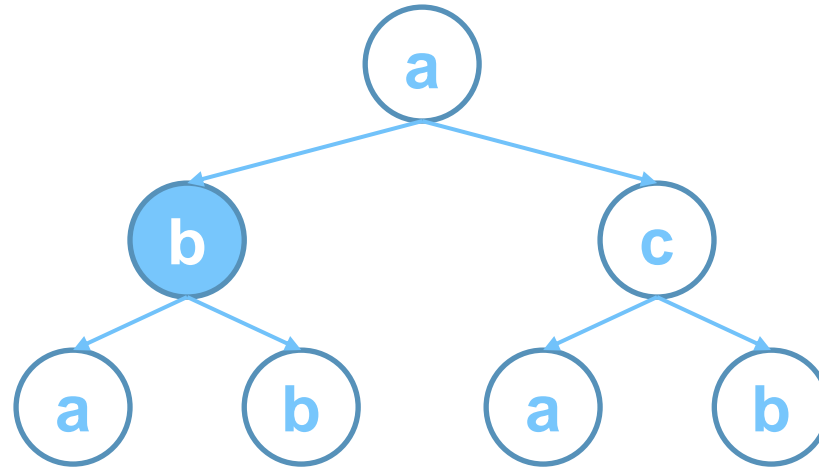
- Chọn tất cả phần tử **c** cùng cấp với phần tử **b**, và những phần tử **c** phải nằm sau phần tử **b**.
- Cú pháp: **b ~ c { }**



08. Combinator selectors (Bộ chọn tổ hợp)

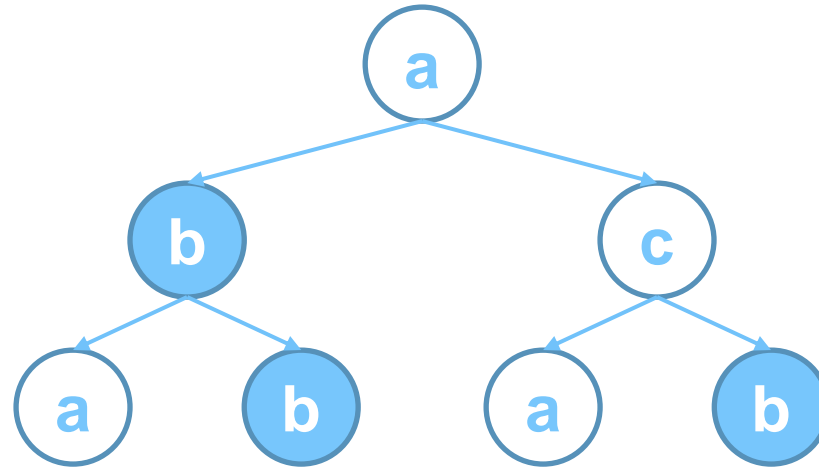
- **Child selectors (Bộ chọn con)**

- Chọn tất cả phần tử **b** là cấp con đầu tiên (cấp 1) của phần tử **a**.
- Cú pháp: **a > b { }**



08. Combinator selectors (Bộ chọn tổ hợp)

- **Descendant selectors (Bộ chọn hậu duệ)**
 - Chọn tất cả phần tử **b** bên trong phần tử **a**.
 - Cú pháp: **a b { }**



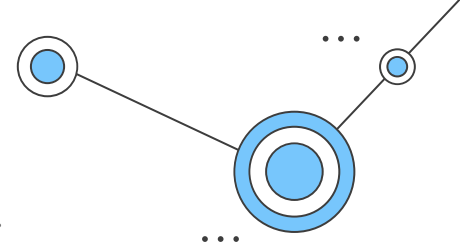


09

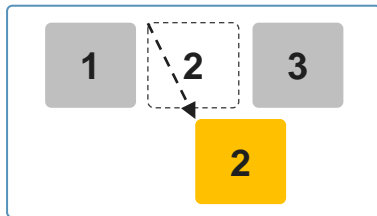
Position (Vị trí)



09. Position (Vị trí)



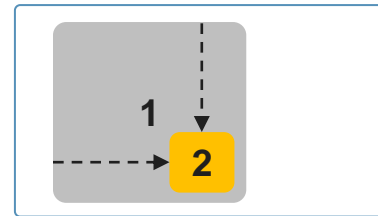
- Thuộc tính **position** được sử dụng để xác định vị trí của một phần tử trên trang web.
- Có **5 giá trị**:



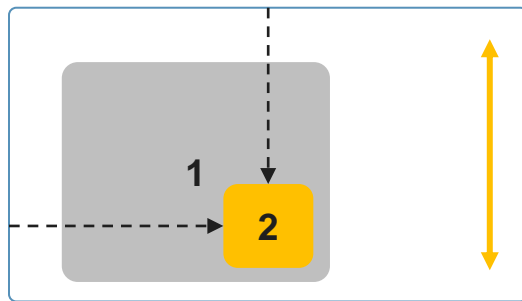
relative (tương đối)



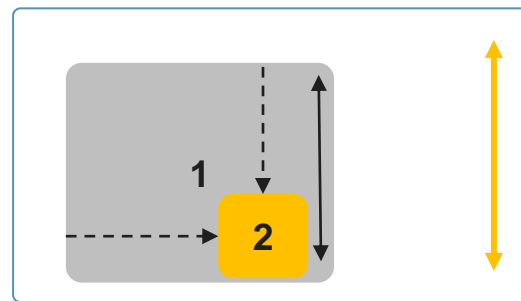
static (tĩnh)



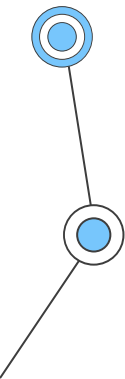
absolute (tuyệt đối)



fixed (cố định)

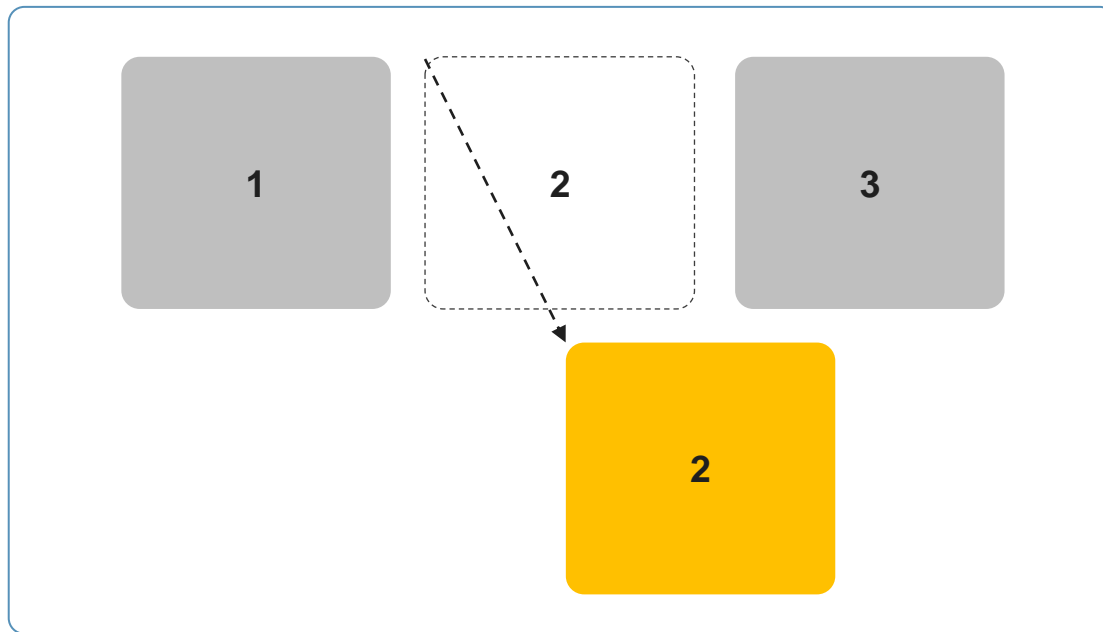


sticky (tương đối + tuyệt đối)

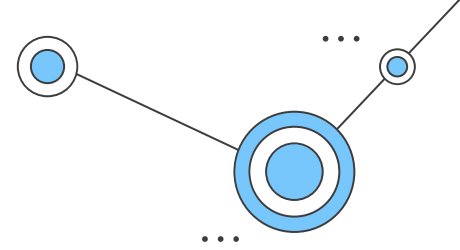


09. Position (Vị trí)

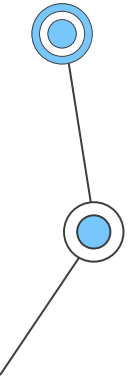
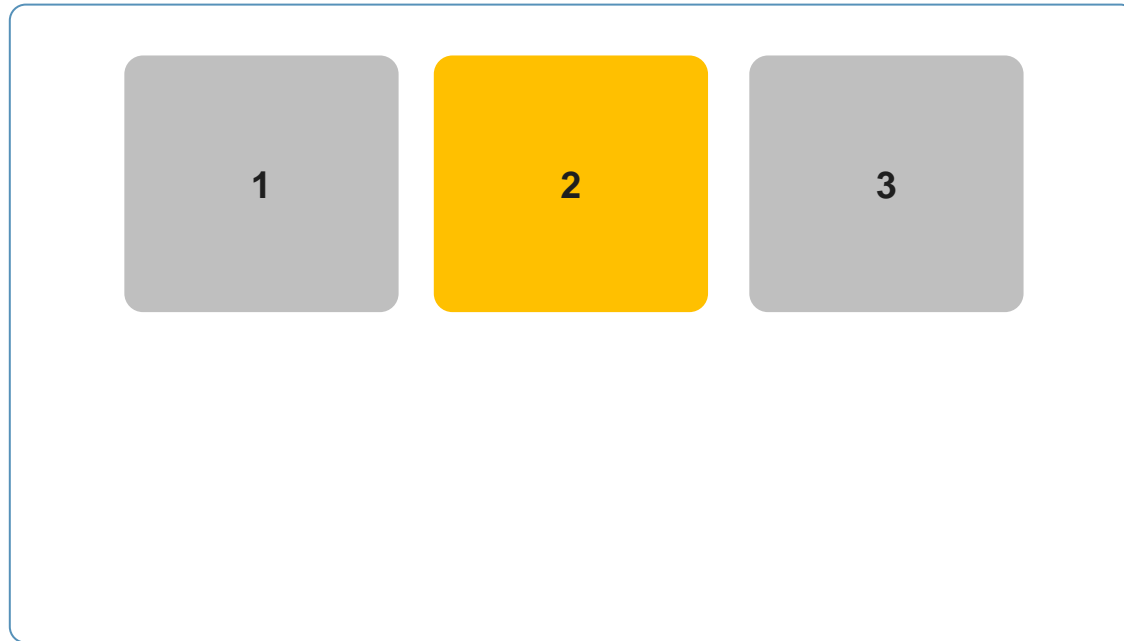
- **relative** (tương đối)
 - Vị trí tương đối so với vị trí mặc định của chính nó.
 - Sử dụng được các thuộc tính top, right, bottom, left.



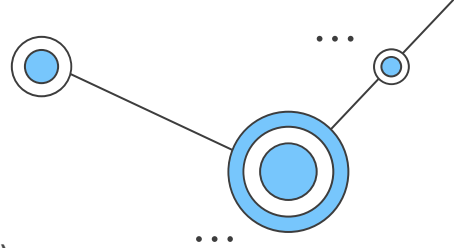
09. Position (Vị trí)



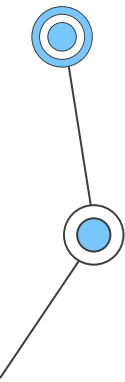
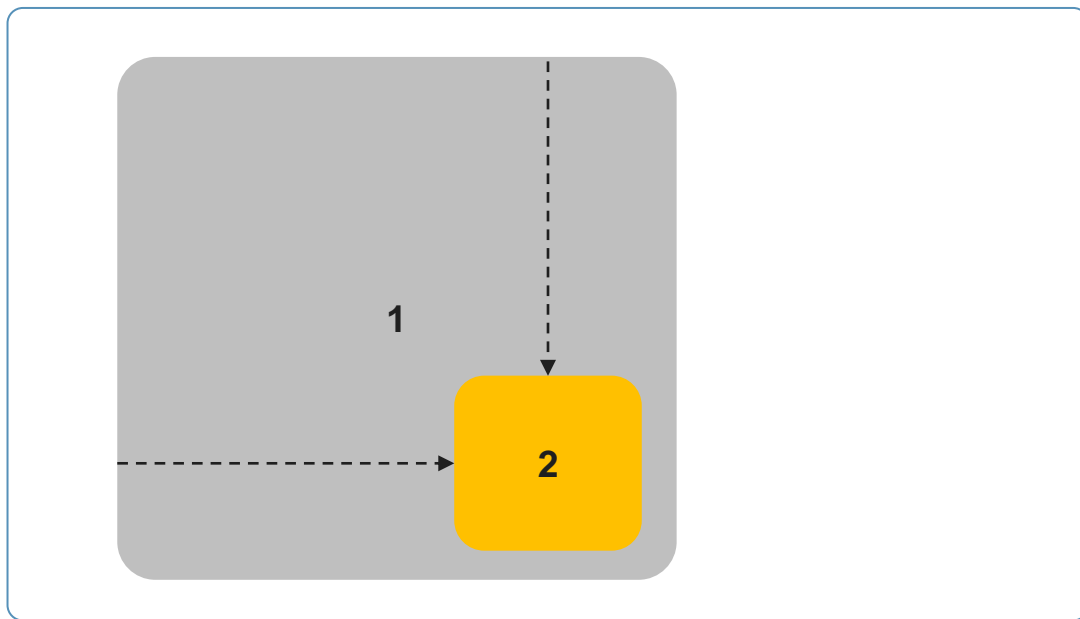
- **static** (tĩnh)
 - Vị trí luôn cố định so với vị trí mặc định của chính nó.
 - Không sử dụng được các thuộc tính top, bottom, right, left.



09. Position (Vị trí)

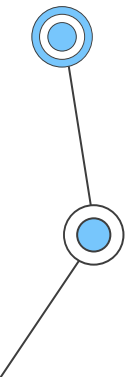
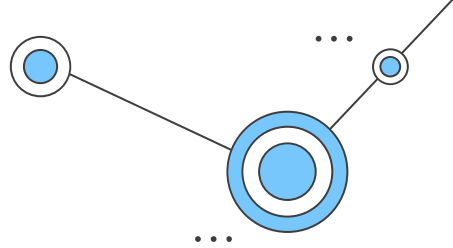
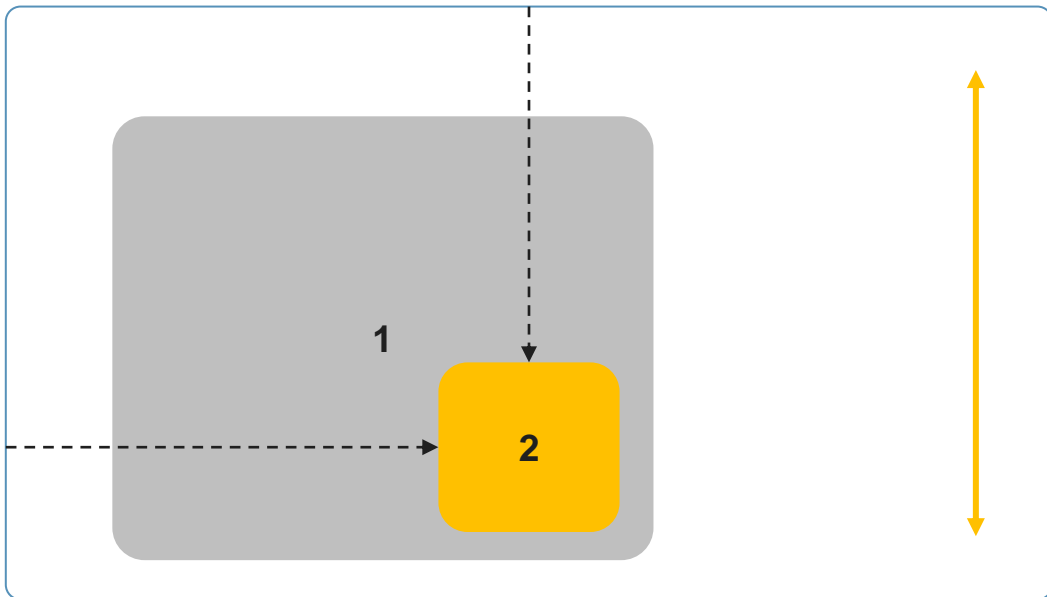


- **absolute** (tuyệt đối)
 - Vị trí tuyệt đối so với phần tử bao ngoài (phần tử cha - có position khác static).
 - Sử dụng được các thuộc tính top, right, bottom, left.

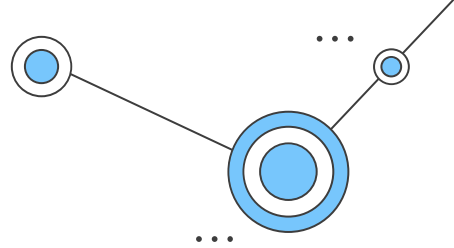


09. Position (Vị trí)

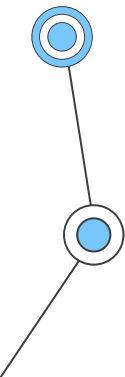
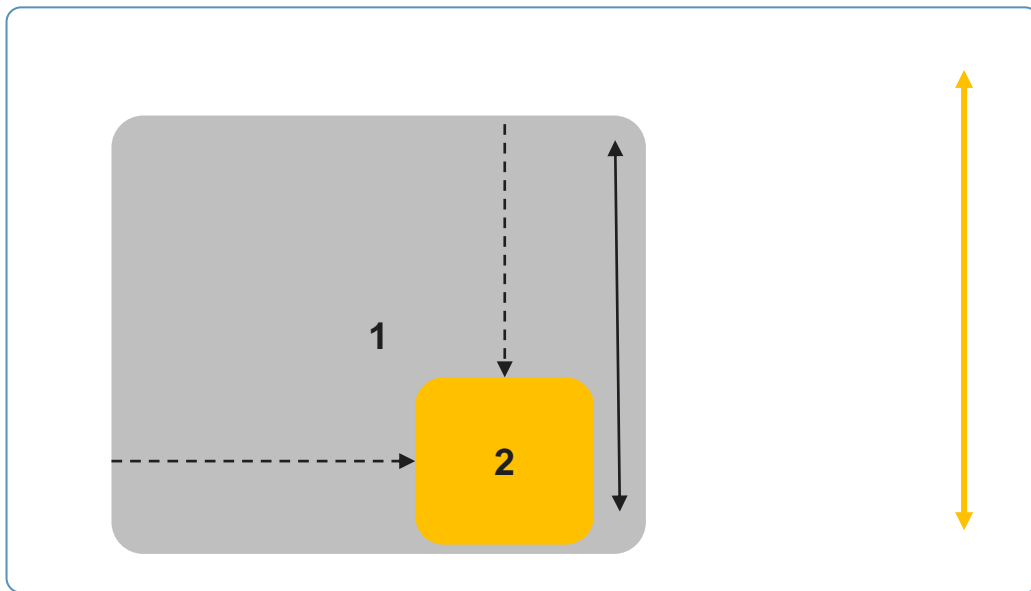
- **fixed** (cố định)
 - Phần tử được đặt vị trí tương đối với cửa sổ trình duyệt.
 - Dù có cuộn trang, phần tử sẽ vẫn giữ vị trí ban đầu của nó.
 - Sử dụng được các thuộc tính top, right, bottom, left.



09. Position (Vị trí)



- **sticky** (tương đối + tuyệt đối)
 - Là sự kết hợp của 2 kiểu **relative** và **fixed**.
 - Khi **chưa scroll đến** phần tử đó thì phần tử sẽ **hiển thị kiểu relative**.
 - Khi **scroll đến** phần tử đó thì phần tử sẽ **hiển thị kiểu fixed**.





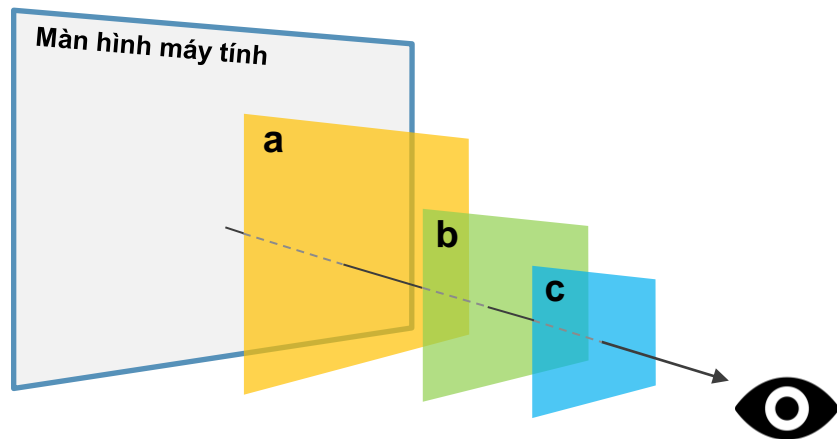
10

z-index

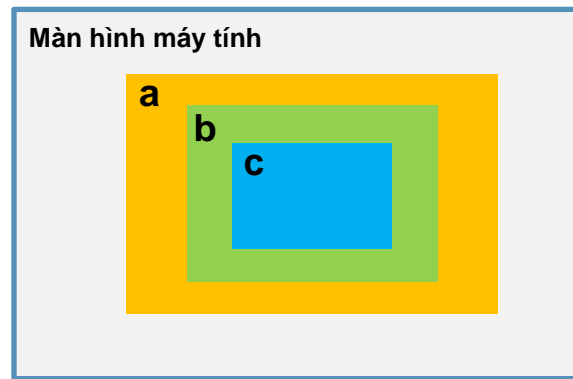


10. z-index

- Thuộc tính **z-index** được sử dụng để xác định thứ tự xếp chồng của các phần tử trùng vị trí.
- Giá trị mặc định là 0.
- z-index càng cao thì phần tử đó càng nằm trên.
- **Chú ý:** **z-index chỉ có tác dụng** khi thuộc tính **position khác static**.



Hình 3D



Hình 2D

Bài tập

- **Link bài tập:** <https://course-back-end-24.vercel.app/lessons/lesson-3/index.html>

