

# *FLOAT, FLEX, GRID*

Nền tảng công nghệ web



# Các cách truyền thống sắp xếp phần tử trên website

Table  
Frame  
Float



Nhược điểm	
<ul style="list-style-type: none"><li>• Khó dùng</li><li>• Phức tạp</li><li>• Phần tử hiển thị hơi thô</li><li>• Khó tùy biến</li></ul>	

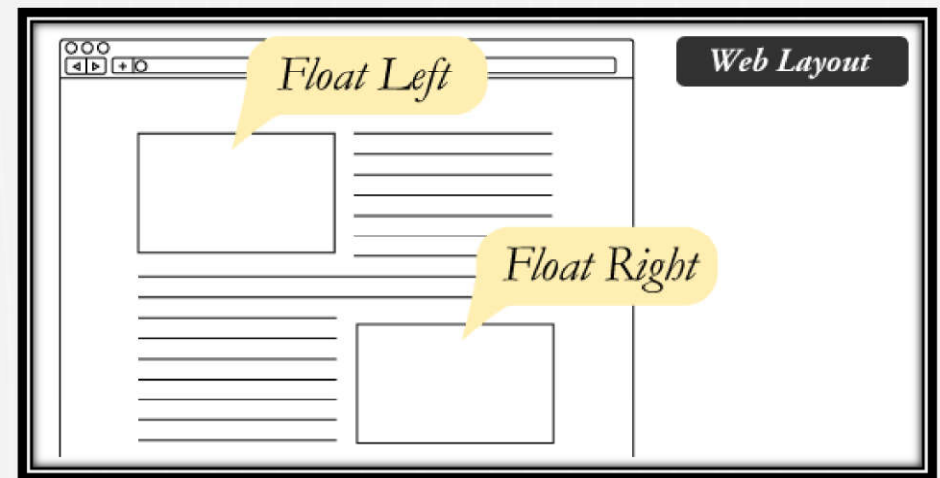
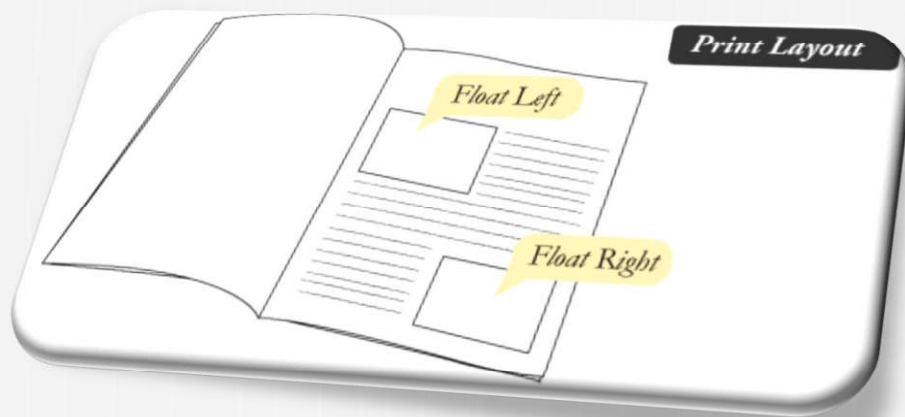
# Float

- ▶ **Float** là thuộc tính quyết định việc hiển thị và cho phép các phần tử nằm phía sau nó lấp phủ phần không gian trống mà nó để lại
  - ▶ None: mặc định
  - ▶ Left: Ép vào phía bên trái và để lại khoảng trống bên phải nó cho phần tử sau có thể dồn lên
  - ▶ Right: Ép vào phía bên phải và để lại khoảng trống bên trái cho các phần tử phía sau nó dồn lên

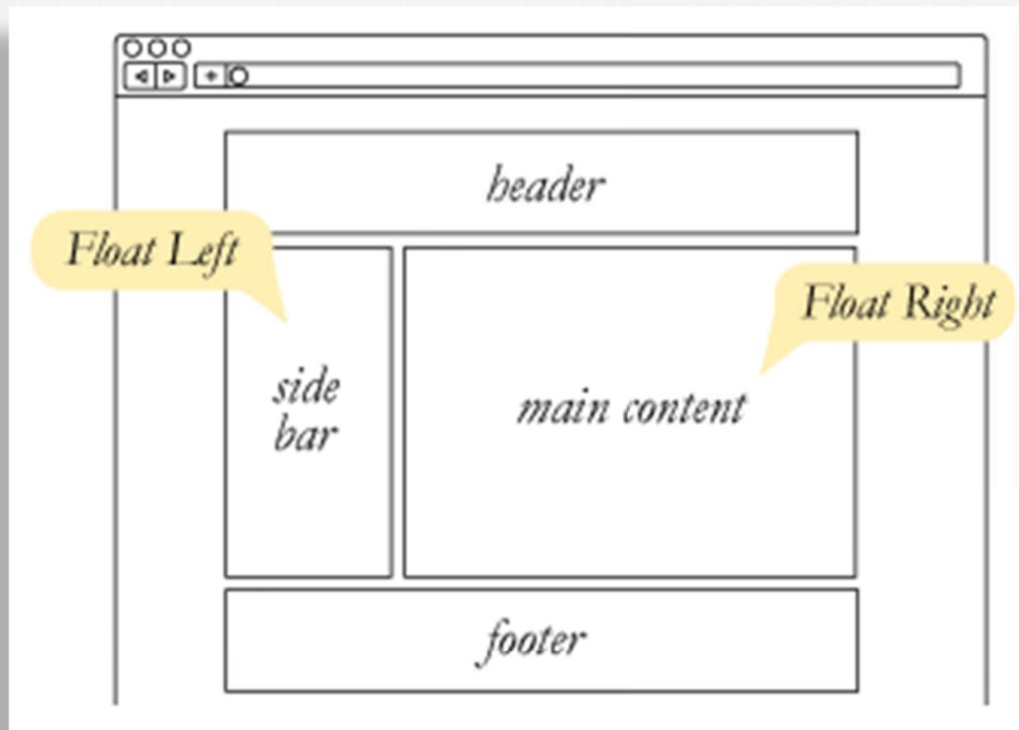
# Float

- ▶ **Clear:** ngăn chặn việc nhảy lên và che phủ khoảng trống mà phần tử trước nó để lại
  - ▶ Left: Chặn không cho dồn lên khoảng trống mà phần tử trước nó sử dụng float:left để lại
  - ▶ Right: Chặn không cho dồn lên khoảng trống mà phần tử trước nó sử dụng float:right để lại
  - ▶ Both: Chặn không cho che phủ bất kỳ khoảng trống nào

# Float

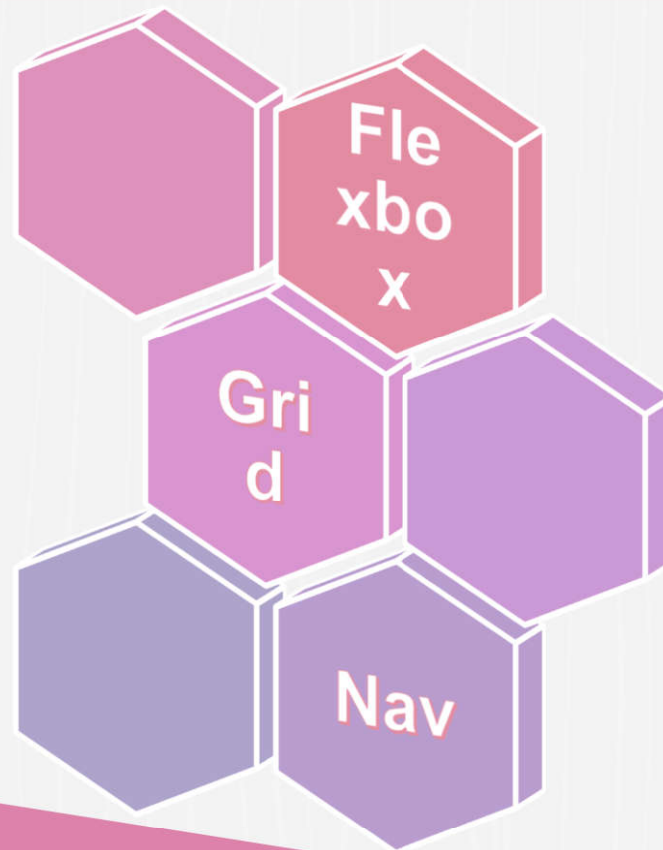


# Float



- **Float:** left;
- **Float:** right;

# Cách thông dụng hiện nay để định vị phần tử



# Flexbox là gì?

- là một mô hình bố cục một chiều cung cấp một phương pháp để cung cấp khả năng phân phối không gian và có thể tùy chỉnh nhiều yếu tố
- Sử dụng dễ dàng
- **CSS Grid** là một tùy chọn khác để căn chỉnh, định vị



# Flex containers và Flex items

- **Flex containers** là phần tử chứa các **Flex items**
- **Flex items** là con trực tiếp của **Flex containers**
- Một **containers** được thiết lập với thuộc tính *display:flex*

```
<div class="flex-container">  
  <div class="flex-item">Item 1</div>  
  <div class="flex-item">Item 2</div>  
  <div class="flex-item">Item 3</div>  
    
</div>
```

# Thuộc tính

## Flex Container

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items

## Flex Item

- order
- flex-grow
- flex-shrink
- flex-basis
- align-self

# Flex Container

Flex-direction	xác định hướng container xếp các item	column	Item theo cột (giống block)
		column-reverse	Item theo cột đảo chiều (giống block đảo)
		row	Item theo hàng (giống float left)
		row-reverse	Item theo hàng (đảo chiều (giống float right)
flex-wrap	chỉ định xem các item có nên bọc hay không	no-wrap	không xuống dòng, chỉ hiển thị trên 1 hàng
		wrap	không đủ chiều rộng của khối cha thì tự động xuống dòng
		wrap-reverse	Giống wrap nhưng đảo ngược
Flex-flow	thuộc tính viết tắt để thiết lập cả flex-direction và flex-wrap.	row wrap	

# Flex Container

Justify-content hoặc align-items	căn chỉnh các item theo chiều ngang/chiều dọc	center	từ tâm của phần tử cha
		flex-start	từ đầu vùng chứa (mặc định)
		flex-end	từ cuối vùng chứa
Justify-content		Space-around	hiển thị thêm khoảng trắng trước, giữa và sau các dòng
		space-between	tạo khoảng cách giữa các item
Align-items		stretch	kéo dài các item để lấp đầy vùng chứa (mặc định)
		space-between	hiển thị các đường uốn với khoảng cách bằng nhau
		space-around	hiển thị các đường gấp khúc có khoảng trắng trước, giữa và sau chúng
		baseline	căn cho nội dung thẳng hàng

# Flex Item

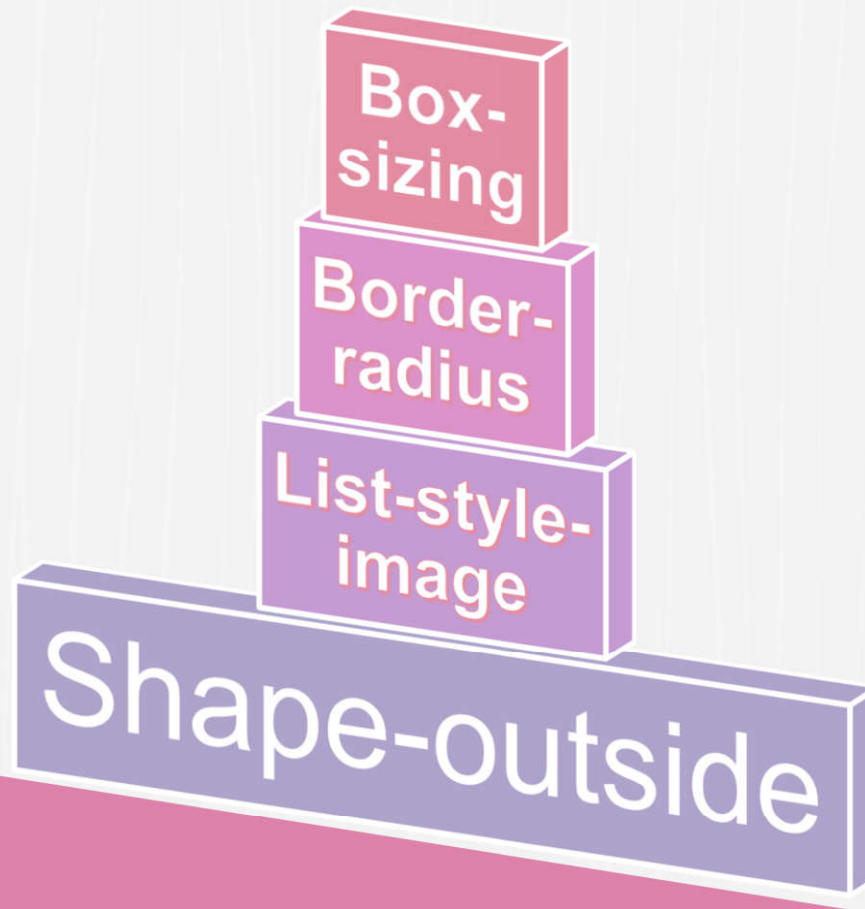
- Yêu cầu SV tự tìm hiểu r tự tạo bảng điền vào. Tự tạo ví dụ demo

# *SHAPE*

Nền tảng công nghệ web



# Một số thuộc tính CSS liên quan shape



# Box-sizing

- ✓ Xác định cách tính chiều rộng và chiều cao của phần tử

## Content-box

- Chỉ áp dụng cho nội dung phần tử

## Border-box

- Áp dụng cho tất cả các phần tử



# Box-sizing

- border-width: 12px
- padding: 30px
- width: 200px

## Content-box

$Width = 24px + 60px + 200px = \mathbf{284px}$ .

## Border-box

Width=200px còn  
content=200px - 60px  
- 24px = **116px**

# Border-radius

- ✓ Làm tròn các góc cạnh viền ngoài của phần tử
- ✓ Theo 2 hướng và 4 vị trí

## Cách gán giá trị

- Tuyệt đối
- Tương đối

- `border-top-left-radius`
- `border-top-right-radius`
- `border-bottom-right-radius`
- `border-bottom-left-radius`

# List-style-image

- ✓ Đặt điểm đánh dấu chỉ mục là một hình ảnh
- ✓ Theo 2 hướng và 4 vị trí

```
list-style-image: url(' ');
```

## Cách gán địa chỉ

- Tuyệt đối
- Tương đối

# Shape-outside

- ✓ Xác định một hình dạng xung quanh mà nội dung inline liền kề sẽ bao bọc.
- ✓ Có nhiều cách tạo hình:
  - `circle()`: tạo hình tròn
  - `ellipse()`: tạo hình elip
  - `Inset()`: tạo hình chữ nhật
  - `polygon()`: tạo bất kỳ hình dạng nào có 3 đỉnh trở lên
  - `url()`: xác định hình ảnh nào sử dụng để bao văn bản

**FORM:**

**ĐĂNG NHẬP DIỄN ĐÀN**

Tên đăng nhập:

Mật khẩu:

Đăng nhập

Phục hồi lại

FORM:

ĐĂNG NHẬP DIỄN ĐÀN	
Tên đăng nhập:	<input type="text"/>
Mật khẩu:	<input type="password"/>
<input type="button" value="Đăng nhập"/>	<input type="button" value="Phục hồi lại"/>