



# TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

## Khoa Công Nghệ Phần Mềm

# CÔNG NGHỆ WEB VÀ ỨNG DỤNG

**ThS. Trần Thị Hồng Yến**

[yentth@uit.edu.vn](mailto:yentth@uit.edu.vn)

0907380471





# Chương 4: Thiết kế giao diện web

---

1. Giới thiệu tổng quan giao diện web
2. Ảnh cho web
3. Các bước triển khai giao diện web
4. Div Layout
5. Kỹ thuật thiết kế
6. Áp dụng Div CSS layout



# 1. Giới thiệu tổng quan giao diện web

---

- **Các loại hình web phổ biến:**

- Web thông tin điện tử
- Web giải trí
- Web doanh nghiệp – tổ chức
- Web cá nhân
- Web tìm kiếm (search engine)
- Web diễn đàn – mạng xã hội
- Web mail



# 1. Giới thiệu tổng quan giao diện web

---

- **Giao diện web:**

- Là tập hợp các thông tin đa truyền thông (Text, Image, Media, ...)
- Được trình bày theo một bố cục nhất định dựa trên các nguyên lý/luật thiết kế, nhằm đảm bảo yêu cầu về mặt thẩm mỹ.
- Giao diện web là hình ảnh đồ họa của trang web.



# 1. Giới thiệu tổng quan giao diện web

---

- **Các thành phần chính trên giao diện web:**
  - **Header:** logo, company name, slogan, banner.
  - **Navigation controller:** main – top - bottom menu.
  - **Side bar:** left – right components.
  - **Main content:** text, image, media, ...
  - **Footer:** info, copyright.
  - **Background**



# 1. Giới thiệu tổng quan giao diện web

---

- Thiết kế giao diện web là thực hiện việc trình bày các thành phần của giao diện một cách có hệ thống và đạt yêu cầu thẩm mỹ dựa theo:
  - Đặc thù thể hiện của từng loại hình web.
  - Chức năng của website.
  - Cách thức hiển thị trang web (thiết bị, trình duyệt)
  - Công nghệ và chuẩn thiết kế web.



## 2. Ảnh cho web

---

- Điều kiện:
  - Hệ màu: RGB hoặc Index
  - Độ phân giải: 72 - 100 ppi
  - Kích thước: # 20 Kbytes
  - Định dạng:
    - \*.GIF ( $\leq 256$  màu): ảnh đơn giản, ít màu (logo, mảng khối màu), ảnh động, trong suốt.
    - \*.JPG ( $> 256$  màu): ảnh nhiều màu (ảnh chụp).
    - \*.PNG (là tổ hợp của .GIF và .JPG): ảnh nhiều màu, trong suốt.
  - Công cụ xử lý ảnh: Photoshop hoặc Firework.



## 3. Các bước triển khai giao diện web

---

- **Triển khai giao diện web:**
  - Phác thảo giao diện dưới dạng hệ thống hình lưới (grid layout).
  - Thiết kế và phân bố các thành phần của giao diện lên hệ thống lưới tại các vị trí thích hợp.
  - Chuẩn hóa việc trình bày bằng các nguyên lý/luật thiết kế.
  - Tối ưu và kết xuất hình ảnh cho web.
  - Thiết kế hoàn chỉnh giao diện tĩnh cho trang web.

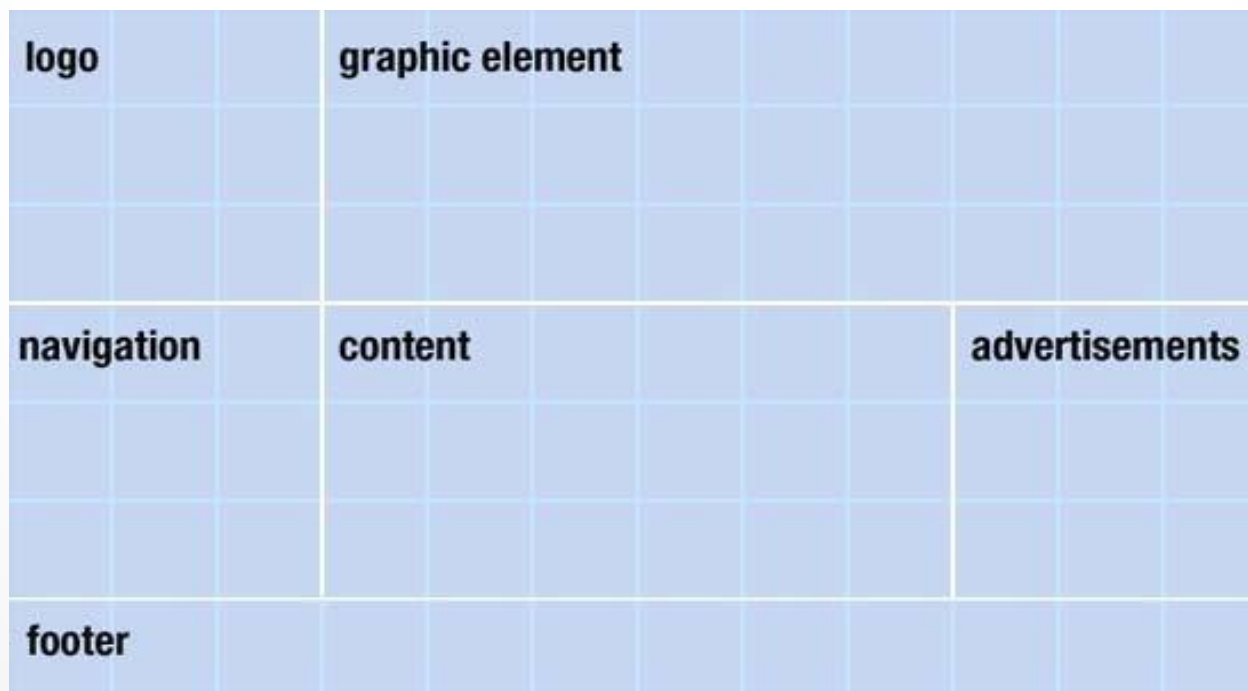




### 3. Các bước triển khai giao diện web

---

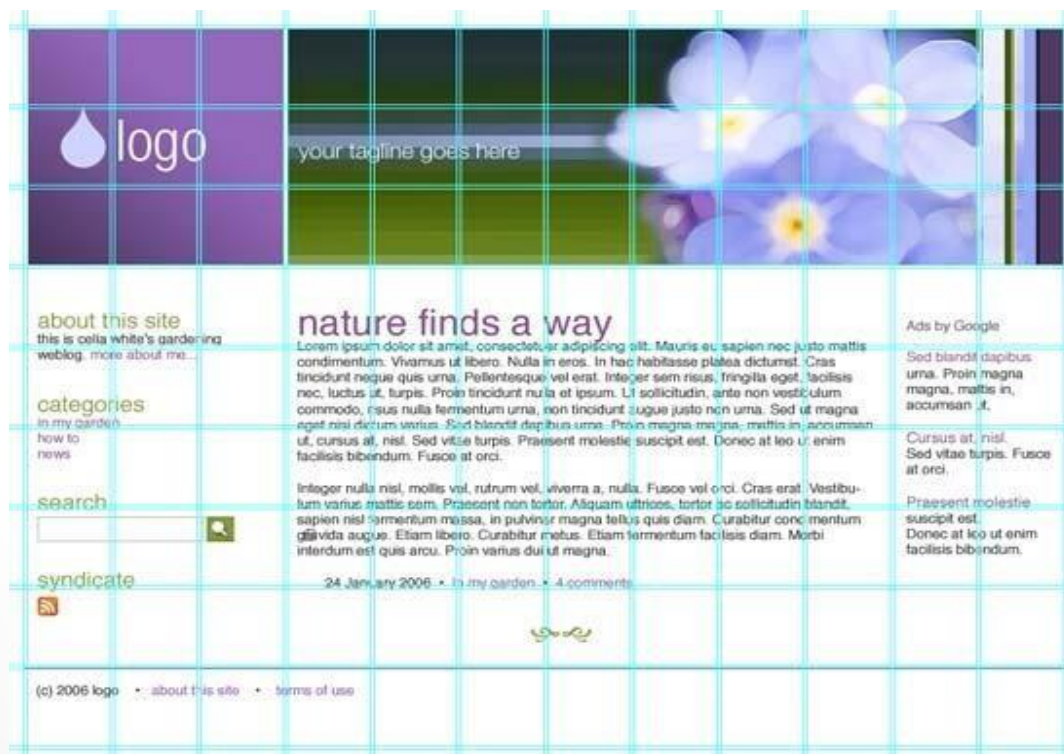
- **Minh họa:**
  - Phác thảo grid layout:





# 3. Các bước triển khai giao diện web

- **Minh họa:**
  - Thiết kế các thành phần của giao diện và chuẩn hóa.





## 4. Div Layout

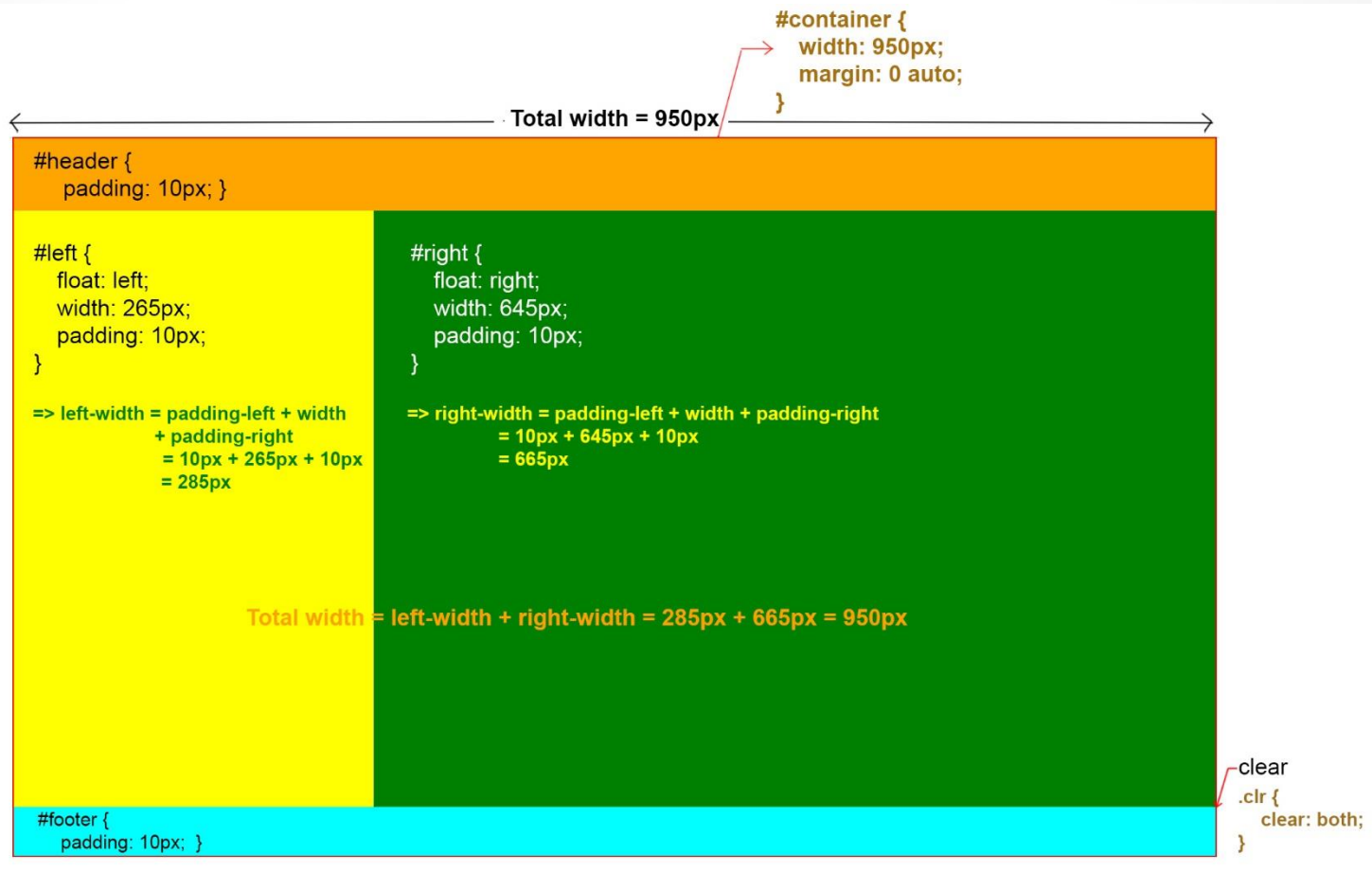
---

- **DIV Layout là gì?**
  - Là kỹ thuật thiết kế giao diện tĩnh của trang web bằng thẻ div của HTML kết hợp với CSS để trình bày các mảng nội dung của giao diện web theo một bố cục xác định.
- **Một số ưu điểm của DIV Layout:**
  - CSS tương thích với hầu hết các trình duyệt.
  - Sự kết hợp làm tăng sức mạnh cho HTML.
  - Linh hoạt trong việc định vị các thành phần trên giao diện web.
  - Hỗ trợ tối đa việc tùy biến giao diện.
  - Dễ dàng nâng cấp giao diện.
  - Hỗ trợ các công cụ tìm kiếm.



## 4. Div Layout

- Ví dụ về DIV Layout:





## 4. Div Layout

- **Code HTML:**

```
<html>
<body>
  <div id="container">
    <div id="header">
      <h1>Header</h1>
    </div>
    <div id="left">
      Nội dung cột trái
    </div>
    <div id="right">
      Nội dung cột phải
    </div>
    <div class="clr"></div>
    <div id="footer">
      Nội dung footer
    </div>
  </div>
</body>
</html>
```



## 4. Div Layout

---

- **Code CSS 1:**

```
* {  
    margin: 0;  
    padding: 0;  
}  
#container {  
    width: 950px;  
    margin: 0 auto;  
}  
#header {  
    background: orange;  
    padding: 10px;  
}
```



## 4. Div Layout

---

- **Code CSS 1:**

```
#left {  
    float: left;  
    width: 265px;  
    background: yellow;  
    padding: 10px;  
    min-height: 450px;  
}  
#right {  
    float: right;  
    width: 645px;  
    background: green;  
    padding: 10px;  
    min-height: 450px;  
}
```



## 4. Div Layout

---

- **Code CSS 1:**

```
.clr {  
    clear: both;  
}  
  
#footer {  
    background: cyan;  
    padding: 10px;  
}
```





## 4. Div Layout

---

- **Code CSS 2:**

```
* {  
    margin: 0;  
    padding: 0;  
}  
#container {  
    width: 70%;  
    margin: 0 auto;  
}  
#header {  
    background: orange;  
    padding: 10px;  
}
```



## 4. Div Layout

---

- **Code CSS 2:**

```
#left {  
    float: left;  
    width: 28%;  
    background: yellow;  
    padding: 1%;  
    min-height: 450px;  
}  
#right {  
    float: right;  
    width: 68%;  
    background: green;  
    padding: 1%;  
    min-height: 450px;  
}
```



## 4. Div Layout

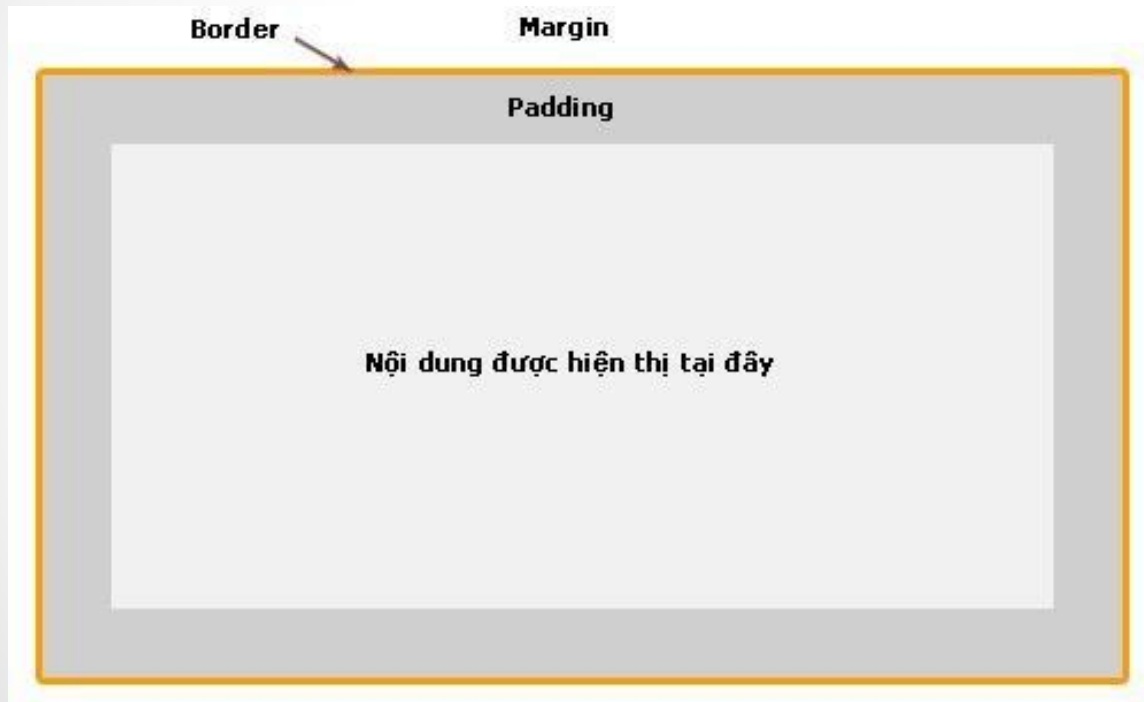
---

- **Code CSS 2:**

```
.clr {  
    clear: both;  
}  
#footer {  
    background: cyan;  
    padding: 10px;  
}
```

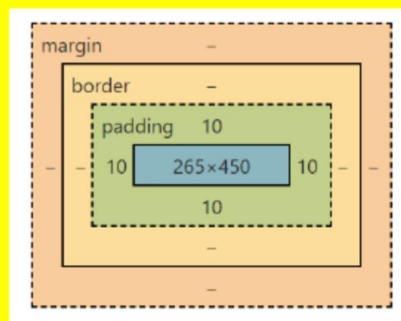


## 5. Kỹ thuật thiết kế



```
#left {  
  float: left;  
  width: 265px;  
  padding: 10px;  
}
```

=> left-width = padding-left + width  
+ padding-right  
= 10px + 265px + 10px  
= 285px





## 5. Kỹ thuật thiết kế

---

- **Box model trong CSS**

- Margin: là phần nằm ngoài cùng trong Box model, nó sẽ tạo ra khoảng cách giữa các thành phần với nhau.
- Padding: là vùng nằm giữa nội dung và đường viền (border), nó tạo ra một khoảng trống giúp cho nội dung được trình bày rõ ràng hơn.
- Border: sẽ quy định độ dày, màu sắc và kiểu hiển thị cũng như kiểu của đường viền.



## 5. Kỹ thuật thiết kế

---

- **Margin trong CSS**

- Khi ta khai báo thuộc tính Margin (canh lề) cho một thành phần nào đó, thì nó sẽ tạo ra một khoảng cách giữa thành phần đó với các thành phần xung quanh nó (top, right, bottom và left).
- Giá trị của margin:
  - auto: tự động canh đều 2 bên left và right, thường được sử dụng để canh giữa màn hình cho toàn bộ trang web.
  - Kích thước (pixels, pt, em, ...).
  - % kích thước của thành phần chứa nó.



## 5. Kỹ thuật thiết kế

- **Margin trong CSS**

- Ví dụ:

```
.first{  
    margin-top: 5px;  
    margin-right: 10px;  
    margin-bottom: 15px;  
    margin-left: 20px;  
}
```

```
<p class="first">
```

Đoạn văn bản này cách phía trên 5px, bên phải 10px; phía dưới 15px, bên trái 20px và không có padding.

```
</p>
```



# 5. Kỹ thuật thiết kế

- **Margin trong CSS**

- Hoặc viết gộp theo chiều kim đồng hồ bắt đầu từ: top – right – bottom – left như sau:
  - VD: "margin: 5px 10px 15px 20px;"
- Có thể viết 3, 2, hoặc 1 giá trị. Nếu một trong 4 giá trị bị thiếu, thì sẽ lấy giá trị của cạnh đối diện:
  - VD: "margin: 5px 10px 15px;" bị thiếu 1 giá trị của cạnh trái => cạnh trái sẽ lấy giá trị của cạnh phải (10px).
  - VD: "margin: 5px 10px;" bị thiếu 2 giá trị của cạnh trái và cạnh dưới => cạnh trái sẽ lấy giá trị của cạnh phải (10px) và cạnh dưới sẽ lấy giá trị của cạnh trên (5px).
  - VD: "margin: 5px;" => tất cả các cạnh đều có margin là 5px.





## 5. Kỹ thuật thiết kế

---

- **Padding trong CSS**

- Khi một thành phần được khai báo padding thì nó sẽ tạo ra một khoảng cách tới đường viền giúp nội dung dễ nhìn hơn.
- Giá trị của padding
  - Kích thước (pixels, pt, em, ...).
  - % kích thước của thành phần chứa nó.



## 5. Kỹ thuật thiết kế

- **Padding trong CSS**

- Ví dụ:

```
.second{  
    padding-top: 2px;  
    padding-right: 3px;  
    padding-bottom: 4px;  
    padding-left: 5px;  
}
```

```
<p class="second">
```

Đoạn văn bản này có lề trên 2px, lề phải 3px, lề dưới 4px và lề trái 5px.

```
</p>
```



## 5. Kỹ thuật thiết kế

- **Padding trong CSS**

- Hoặc viết gộp theo chiều kim đồng hồ bắt đầu từ: top – right – bottom – left như sau:
  - VD: "padding: 2px 3px 4px 5px;"
- Có thể viết 3, 2, hoặc 1 giá trị. Nếu một trong 4 giá trị bị thiếu, thì sẽ lấy giá trị của cạnh đối diện:
  - VD: "padding: 2px 3px 4px;" bị thiếu 1 giá trị của cạnh trái => cạnh trái sẽ lấy giá trị của cạnh phải (3px).
  - VD: "padding: 2px 3px;" bị thiếu 2 giá trị của cạnh trái và cạnh dưới => cạnh trái sẽ lấy giá trị của cạnh phải (3px) và cạnh dưới sẽ lấy giá trị của cạnh trên (2px).
  - VD: "padding: 2px;" => tất cả các cạnh đều có padding là 2px.



## 5. Kỹ thuật thiết kế

---

- **Border trong CSS**

- Thuộc tính Border quy định độ dày, màu sắc và kiểu hiển thị đường viền của một thành phần trong trang web.
- Các giá trị của border bao gồm:
  - border-style (kiểu viền).
  - border-width (độ dày).
  - border-color (màu sắc).
- Để đường viền được hiển thị thì giá trị border-style luôn phải được khai báo.



## 5. Kỹ thuật thiết kế

- Border trong CSS

- Ví dụ:

```
.myborder{
```

```
    border-top-style: solid; border-right-style: dashed;
```

```
    border-bottom-style : dotted; border-left-style: double;
```

```
}
```

```
p{
```

```
    border-style: solid;
```

```
    border-width: 1px;
```

```
    border-color: #000000;
```

```
}
```

hoặc viết gộp như sau:

border: solid 1px #000000;



## 5. Kỹ thuật thiết kế

---

- **Kích thước của Box model**

- Công thức tính kích thước của Box như sau:

**Rộng** = width + left padding + right padding + left border + right border + left margin + right margin

**Cao** = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin



# 5. Kỹ thuật thiết kế

- **Kích thước của Box model**

- Ví dụ:

```
#mybox{  
    width: 150px;  
    height: 100px;  
    padding: 10px;  
    border: 5px solid black;  
    margin: 10px;  
}
```

Độ rộng của Box trên là:

$150\text{px (width)} + 20\text{px (left + right padding)} + 10\text{px (left + right border)} + 20\text{px (left + right margin)} = 200\text{px (chiều rộng)}$

Tương tự, độ cao của Box là: 150px.



## 5. Kỹ thuật thiết kế

- **Thuộc tính box-sizing:**

- Khi sử dụng padding và border thì box bị biến đổi kích thước, nếu có đặt thêm thuộc tính width và height để thiết lập kích thước cho box.

VD: Box có width là 500px và height là 500px (500×500 px), nếu thêm padding là 15px nữa thì box sẽ có kích thước là 530×530 px (thêm 15px cho 4 padding của box bao gồm: padding-top, padding-right, padding-bottom và padding-left).

- Để tiện tính toán và làm chủ được kích thước của box, có thể dùng thuộc tính **box-sizing**. Khi đó, box giữ nguyên kích thước mặc dù có cộng thêm padding và border, nghĩa là width và height là kích thước đã bao gồm border và padding.





## 5. Kỹ thuật thiết kế

- **Thuộc tính box-sizing:**

- **box-sizing** là một thuộc tính trong CSS3 nên khi viết phải viết thành 3 lần với các tiền tố khác nhau.

- VD:

```
box-sizing: border-box;
```

```
-moz-box-sizing: border-box;
```

```
-webkit-box-sizing: border-box;
```

- Trong đó:

- Nếu viết không có tiền tố là dành cho trình duyệt IE8, Opera 7, Firefox và Google chrome bản mới.
- -moz là dành cho Firefox bản cũ.
- -webkit là dành cho Google Chrome bản cũ.



## 5. Kỹ thuật thiết kế

- Thuộc tính **box-sizing**:

- **box-sizing** hỗ trợ một số giá trị như sau:
  - **content-box**: (mặc định) giá trị width và height chỉ áp dụng cho khu vực nội dung bên trong, không bao gồm padding, border và margin.
  - **border-box**: width và height sẽ bao gồm cho cả phần nội dung, padding và border nhưng không bao gồm margin.
  - **padding-box**: (chỉ có tác dụng với trình duyệt Firefox) width và height chỉ bao gồm cho phần nội dung và padding, không bao gồm border và margin.

➡ Nên sử dụng giá trị **border-box** cho dễ tính toán hơn.



## 5. Kỹ thuật thiết kế

- Thuộc tính **box-sizing**:

- Sử dụng **box-sizing** với giá trị là **border-box** cho toàn bộ các phần tử trong website để các phần tử có kích thước chính xác khi khai báo, không phải cộng thêm kích thước các phần border và padding.
- Thiết lập **box-sizing: border-box** cho toàn bộ các phần tử trong website.

```
* {  
    box-sizing: border-box;  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
}
```



## 5. Kỹ thuật thiết kế

### Code CSS 1:

```
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
#container {  
    width: 950px;  
    margin: 0 auto;  
}  
#header {  
    background: orange;  
    padding: 10px;  
}
```



## 5. Kỹ thuật thiết kế

---

- **Code CSS 1:**

```
#left {  
    float: left;  
    width: 285px;  
    background: yellow;  
    padding: 10px;  
    min-height: 450px;  
}  
#right {  
    float: right;  
    width: 665px;  
    background: green;  
    padding: 10px;  
    min-height: 450px;  
}
```



## 5. Kỹ thuật thiết kế

---

- **Code CSS 1:**

```
.clr {  
    clear: both;  
}  
#footer {  
    background: cyan;  
    padding: 10px;  
}
```



## 5. Kỹ thuật thiết kế

- **Code CSS 2:**

```
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
#container {  
    width: 70%;  
    margin: 0 auto;  
}  
#header {  
    background: orange;  
    padding: 10px;  
}
```



## 5. Kỹ thuật thiết kế

---

- **Code CSS 2:**

```
#left {  
    float: left;  
    width: 30%;  
    background: yellow;  
    padding: 1%;  
    min-height: 450px;  
}  
#right {  
    float: right;  
    width: 70%;  
    background: green;  
    padding: 1%;  
    min-height: 450px;  
}
```





## 5. Kỹ thuật thiết kế

---

- **Code CSS 2:**

```
.clr {  
    clear: both;  
}  
#footer {  
    background: cyan;  
    padding: 10px;  
}
```



## 5. Kỹ thuật thiết kế

- **Float trong CSS**

- Với CSS float, một thành phần có thể được hiển thị ở bên trái hay bên phải của thành phần chứa nó, các thành phần khác sẽ di chuyển lên lấp đầy khoảng trống nếu có.
- Float có 3 giá trị sau:
  - Left: cố định phần tử về bên trái.
  - Right: cố định phần tử về bên phải.
  - None: không thay đổi.

Khi không được khai báo thì giá trị float là none.



# 5. Kỹ thuật thiết kế

- **Float trong CSS**

- Ví dụ:

```
<div id="container">
```

```
<div id="left">
```

```

```

```
<p>Cột trái</p>
```

```
</div>
```

```
<div id="right">
```

<p>Div cha "container" có độ rộng 400px, được cố định bên trái (trôi về bên trái - float: left), nền màu vàng và có khoảng cách lề là 10px.</p>

<p>Div con "left" bên trái có độ rộng 200px, được cố định bên trái (trôi về bên trái - float: left), nền màu trắng.</p>

<p>Div con "right" bên phải có độ rộng 200px, được cố định bên phải (trôi về bên phải - float: right), nền màu cam.</p>

```
</div>
```

```
</div>x
```

# 5. Kỹ thuật thiết kế



## • Float trong CSS

- Ví dụ:

```
#container{  
    width: 400px;  
    float: left;  
    background-color: yellow;  
    padding:10px;  
}  
#left {  
    width: 200px; float: left; background-color:white;  
}  
#right {  
    width: 200px; float: right; background-color:orange;  
}
```



Div con "left" bên trái có độ rộng 200px, được cố định bên trái (trôi về bên trái - float: left), nền màu trắng.

Div cha "container" có độ rộng 500px, được cố định bên trái (trôi về bên trái - float: left), nền màu vàng và có khoảng cách lề là 10px.

Div con "left" bên trái có độ rộng 200px, được cố định bên trái (trôi về bên trái - float: left), nền màu trắng.

Div con "right" bên phải có độ rộng 300px, được cố định bên phải (trôi về bên phải - float: right), nền màu cam.



## 5. Kỹ thuật thiết kế

---

- **Clear trong CSS**

- Clear luôn đi cùng với float, có thể coi clear là ngược lại của float.
- Sử dụng clear để ngăn không cho các thành phần phía dưới tràn lên trên.
- Clear có 4 giá trị sau:
  - Left
  - Right
  - Both
  - None



## 5. Kỹ thuật thiết kế

- **Clear trong CSS**

- Ví dụ:

```
<div id="container">
```

```

```

Đoạn 1 được cố định bên trái (trôi về bên trái - float: left).

```
<div class="clear"></div>
```

```

```

Đoạn 2 được cố định bên trái (trôi về bên trái - float: left).

```
</div>
```



## 5. Kỹ thuật thiết kế

- **Clear trong CSS**
  - Ví dụ:



Đoạn 1 được cố định bên trái (trôi về bên trái - float: left).

Đoạn 2 được cố định bên trái (trôi về bên trái - float: left).



## 5. Kỹ thuật thiết kế

---

- **Display trong CSS**

- Thuộc tính Display quy định phương thức hiển thị của các thành phần trên trang web.
- Được ứng dụng để tạo các hiệu ứng ẩn/hiện một hay nhiều nhóm đối tượng.
- Display có các giá trị sau:
  - None: ẩn một thành phần trong trang web
  - Block: hiển thị dạng khối
  - Inline: hiển thị trên dòng





## 5. Kỹ thuật thiết kế

- **Display trong CSS**

- Ví dụ:

- ```
<div>Thành phần div 1</div>
```

- ```
<div>Thành phần div 2</div>
```

- ```
<span>Thành phần span 1</span>
```

- ```
<span>Thành phần span 2</span>
```

Đoạn code sau sẽ định dạng thành phần block div thành inline và thành phần inline span thành block

```
div{display: inline}
```

```
span{display: block}
```

Thành phần div 1

Thành phần div 2

Thành phần span 1 Thành phần span 2

Thành phần div 1 Thành phần div 2

Thành phần span 1

Thành phần span 2

1

2



## 5. Kỹ thuật thiết kế

---

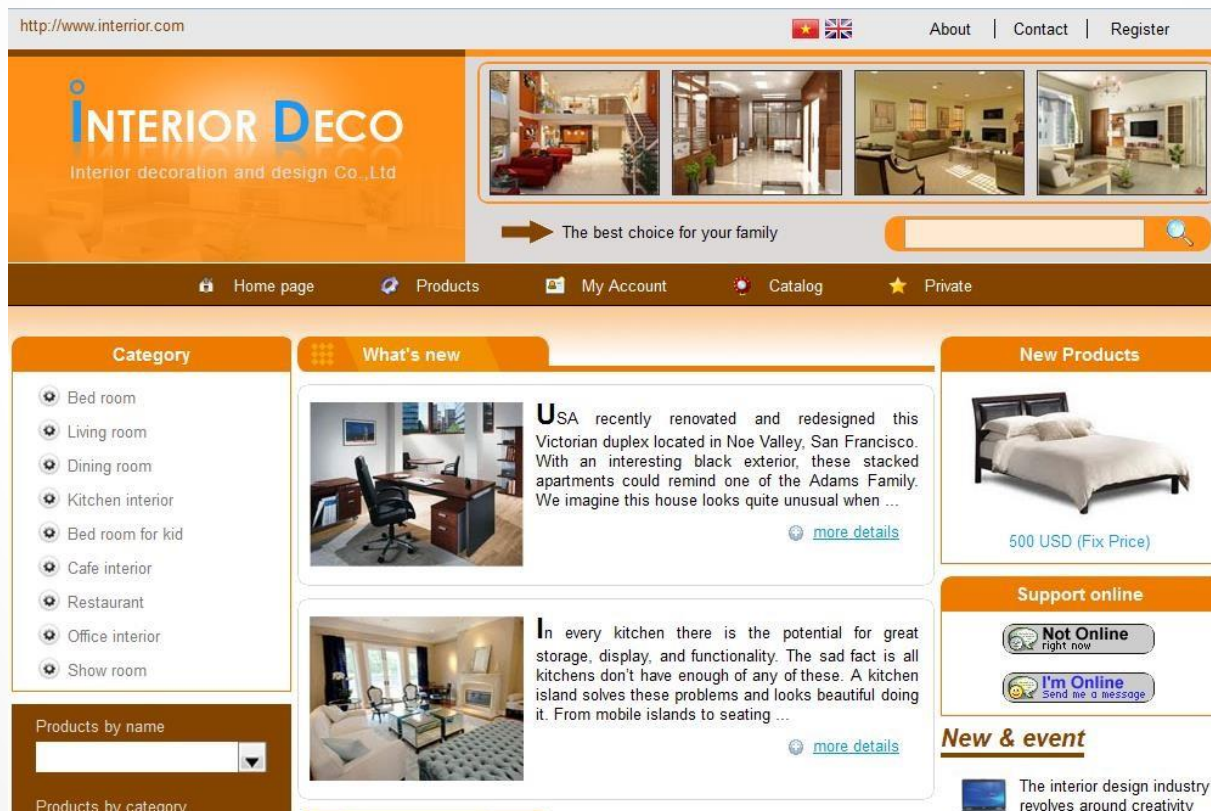
- **HTML Responsive Web Design:**

[https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp)



## 6. Áp dụng Div CSS layout

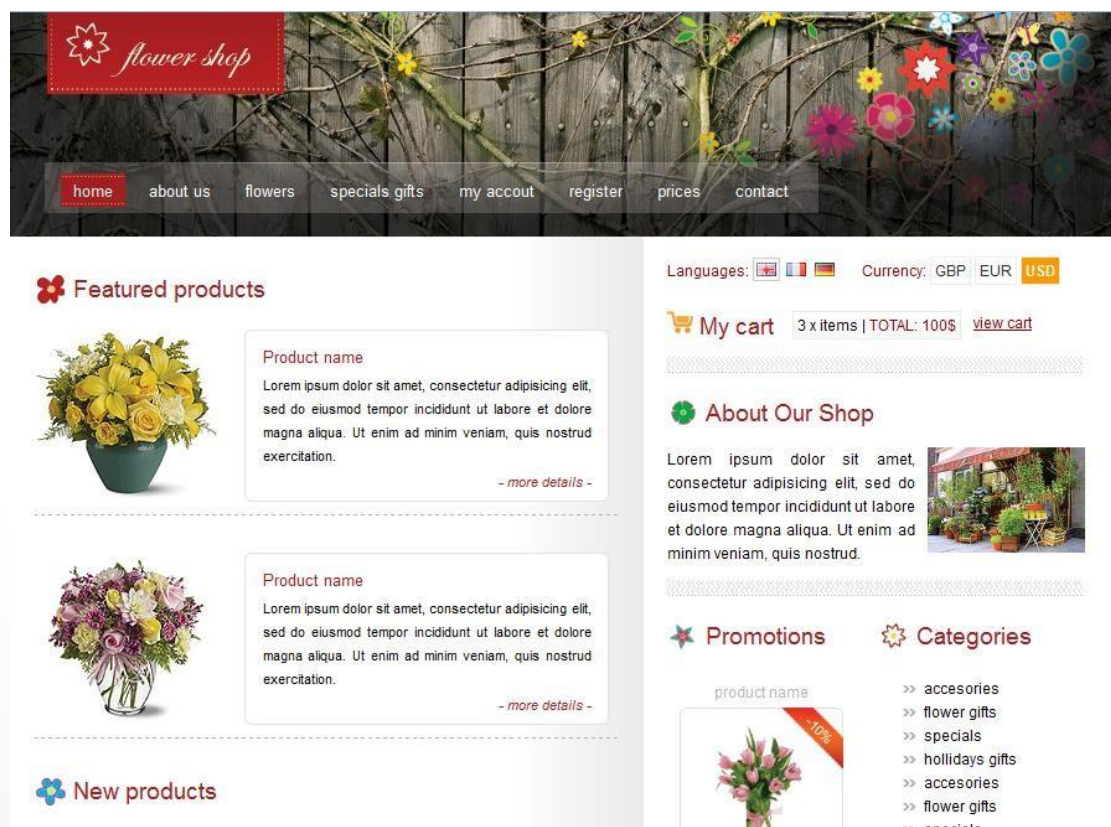
- Thiết kế giao diện web nội thất





# 6. Áp dụng Div CSS layout

- Thiết kế giao diện web shop hoa





# Thảo luận

---

