

CSS LAYOUT

Khoá học "Building website with HTML & CSS"

Mục tiêu

- Tạo được bố cục trang web bằng cách sử dụng thẻ div và các thuộc tính css
- Tạo được bố cục trang web bằng cách sử dụng các thẻ `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`

Phần tử cấp khối – Block

- Một phần tử cấp khối (Block) luôn bắt đầu trên một dòng mới và chiếm toàn bộ chiều rộng có sẵn (trải dài sang trái và phải hết mức có thể).
- Ví dụ về các phần tử cấp khối:
 - <div>
 - <h1> – <h6>
 - <p>
 - <form>
 - <header>
 - <footer>
 - <section>

Phần tử nội tuyến – Inline

- Một phần tử nội tuyến (Inline) không bắt đầu trên một dòng mới và chỉ chiếm nhiều chiều rộng khi cần thiết.
- Ví dụ về phần tử nội tuyến:
 - ``
 - `<a>`
 - ``


Thuộc tính display

- Thay đổi phần tử nội tuyến (Inline) thành phần tử khối (Block) hoặc ngược lại, có thể hữu ích để làm cho trang trông theo một cách cụ thể và vẫn tuân theo các tiêu chuẩn web.
- Thuộc tính `display` được dùng để quy định kiểu “hộp” hiển thị cho một phần tử
- Các kiểu display thông dụng là:
 - `block`: hiển thị dưới dạng một khối, chiếm toàn bộ độ rộng
 - `inline`: hiển thị trong một dòng
 - `inline-block`: hiển thị dưới dạng một khối trong một dòng, chỉ chiếm độ rộng cần thiết
 - `none`: Không hiển thị, không chiếm không gian của trang web

Thuộc tính display: Block

- Hiện thị dưới dạng một khối, chiếm toàn bộ dòng

```
<!DOCTYPE html>
<html>
<head>
<style>
.example{
display: block;
background-color: red;
}
</style>
</head>
<body>
<span class="example">Xin chào</span>
<span class="example">Xin chào</span>
</body>
</html>
```



Xin chào
Xin chào

Thuộc tính display: Inline

- Hiển thị trong một dòng

```
<!DOCTYPE html>
<html>
<head>
<style>
.example{
display: inline;
background-color: red;
}
</style>
</head>
<body>
<span class="example">Xin chào</span>
<span class="example">Xin chào</span>
</body>
</html>
```



Xin chào Xin chào

Thuộc tính display: Inline-Block

- Hiện thị dưới dạng một khối trong một dòng, chỉ chiếm độ rộng cần thiết

```
<!DOCTYPE html>
<html>
<head>
<style>
.example{
display: inline-block;
width: 200px;
height: 50px;
background-color: red;
}
</style>
</head>
<body>
<span class="example">Xin chào</span>
<span class="example">Xin chào</span>
</body>
</html>
```

Xin chào

Xin chào

Thuộc tính display: None

- Không hiển thị, không chiếm không gian của trang web

```
<!DOCTYPE html>
<html>
<head>
<style>
.example{
display: none;
background-color: red;
}
</style>
</head>
<body>
<span class="example">Xin chào</span>
<span class="example">Xin chào</span>
</body>
</html>
```

Width và max-width

- Một phần tử cấp khối luôn chiếm toàn bộ chiều rộng có sẵn (trải dài sang trái và phải hết mức có thể).
- Đặt chiều rộng (width) của phần tử cấp khối sẽ ngăn nó kéo dài ra các cạnh của vùng chứa.
- Sử dụng max-width, sẽ cải thiện khả năng xử lý các cửa sổ nhỏ của trình duyệt. Điều này quan trọng khi làm cho một trang web có thể sử dụng được trên các thiết bị nhỏ.

Ví dụ width và max-width

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
width: 500px;
margin: auto;
border: 3px solid #73AD21;
}
div.ex2 {
max-width: 500px;
margin: auto;
border: 3px solid #73AD21;
}
</style>
</head>
<body>
<h2>CSS Max-width</h2>
<div class="ex1">This div element has width:
500px;</div>
<br>
<div class="ex2">This div element has max-width:
500px;</div>
</body>
</html>
```

CSS Max-width

This div element has width: 500px;

This div element has max-width: 500px;

Chú ý: Kéo cửa sổ trình duyệt nhỏ hơn 500px để nhìn thấy sự khác biệt giữa 2 div

Horizontal & Vertical Align

- Để căn giữa một phần tử khối theo chiều ngang (như `<div>`), hãy sử dụng `margin: auto`.
- Để căn giữa văn bản bên trong một phần tử, hãy sử dụng `text-align: center`.
- Để căn giữa một hình ảnh, hãy đặt `margin-left` và `margin-right` thành `auto` và biến nó thành một phần tử khối (block).
- Một phương pháp để căn chỉnh các phần tử là sử dụng `position: absolute`.
- Một phương pháp khác để căn chỉnh các phần tử là sử dụng thuộc tính `float`.

Ví dụ Horizontal & Vertical Align

Căn giữa các phần tử

```
.center {  
margin: auto;  
width: 60%;  
border: 3px solid #73AD21;  
padding: 10px;  
}
```

Ví dụ Horizontal & Vertical Align

Căn giữa hình ảnh

```
img {  
display: block;  
margin-left: auto;  
margin-right: auto;  
}
```

Ví dụ Horizontal & Vertical Align

Căn trái và Phải – Sử dụng position

```
.right {  
position: absolute;  
right: 0px;  
width: 300px;  
border: 3px solid #73AD21;  
padding: 10px;  
}
```

Ví dụ Horizontal & Vertical Align

Căn trái và Phải – Sử dụng float

```
.right {  
float: right;  
width: 300px;  
border: 3px solid #73AD21;  
padding: 10px;  
}
```


Clearfix Hack

Nếu một phần tử cao hơn phần tử chứa nó và nó được thả nổi, nó sẽ tràn ra bên ngoài vùng chứa của nó. Bạn có thể sử dụng “hack clearfix” để sửa lỗi này.

```
.clearfix::after {  
  content: "";  
  clear: both;  
  display: table;  
}
```

Thuộc tính float và clear

- Thuộc tính `float` được dùng để quy định một thành phần có “trôi” về phía trái hoặc phía phải hay không.
- Thuộc tính `clear` được dùng để chỉ định rằng không cho phép các thành phần khác được “trôi” ở bên trái hoặc bên phải hoặc cả hai bên của một thành phần khác
- Demo thuộc tính `float` và `clear`:
<https://jsfiddle.net/codegym/03dgwzfg/>

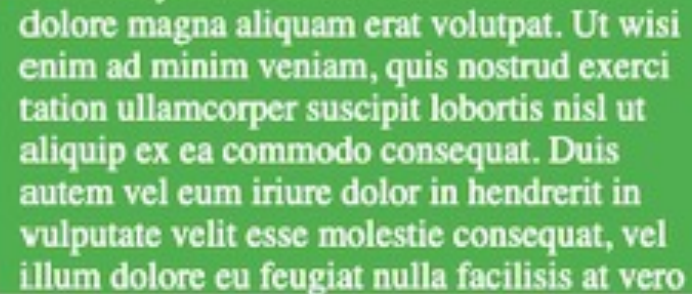
Thuộc tính Overflow

- Thuộc tính overflow kiểm soát những gì xảy ra với nội dung quá lớn để vừa với một khu vực.
- Thuộc tính Overflow chỉ định cắt nội dung hoặc thêm thanh cuộn khi nội dung của một phần tử quá lớn để vừa với khu vực được chỉ định.
- Thuộc tính Overflow có các giá trị sau:
 - **visible** – Mặc định. Phần tràn không được cắt bớt. Nội dung hiển thị bên ngoài hộp của phần tử
 - **hidden** – Phần tràn bị cắt bớt và phần còn lại của nội dung sẽ không hiển thị
 - **scroll** – Phần tràn được cắt bớt và một thanh cuộn được thêm vào để xem phần còn lại của nội dung
 - **auto** – Tương tự như scroll, nhưng nó chỉ thêm thanh cuộn khi cần thiết

Ví dụ thuộc tính Overflow

- Thuộc tính overflow kiểm soát những gì xảy ra với nội dung quá lớn để vừa với một khu vực.

```
#overflowTest {  
background: #4CAF50;  
color: white;  
padding: 15px;  
width: 50%;  
height: 100px;  
overflow: scroll;  
border: 1px solid #ccc;  
}
```



dolore magna aliquam erat volutpat. Ut wisi
enim ad minim veniam, quis nostrud exerci
tation ullamcorper suscipit lobortis nisl ut
aliquip ex ea commodo consequat. Duis
autem vel eum iriure dolor in hendrerit in
vulputate velit esse molestie consequat, vel
illum dolore eu feugiat nulla facilisis at vero

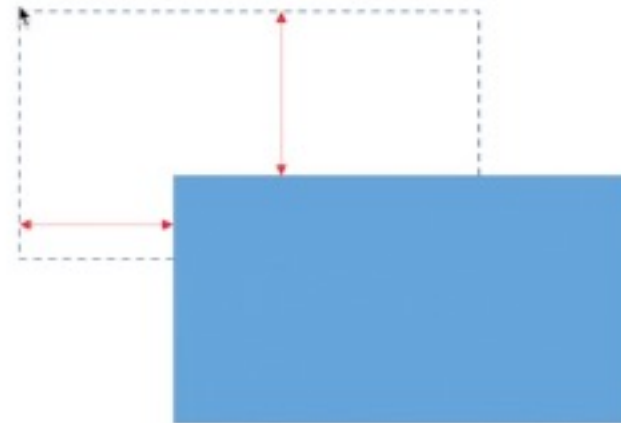
Thuộc tính Position

- Thuộc tính position được sử dụng để quy định cách thức sắp xếp vị trí của một thành phần trong trang web
- Các giá trị:
 - Static
 - Relative
 - Absolute
 - Fixed

Relative Position

- Định vị một thành phần so với vị trí mặc định của chính bản thân nó

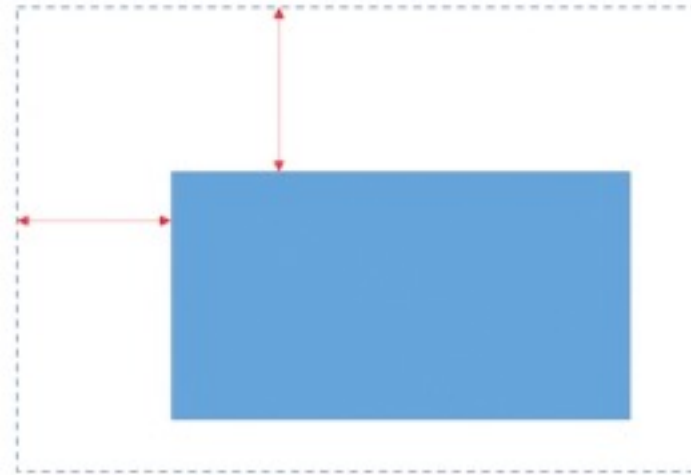
```
div {  
  position: relative;  
  left: 100px;  
  top: 150px;  
}
```



Absolute Position

- Định vị một thành phần so với vị trí của thành phần cha gần nhất của nó (thành phần cha phải có thuộc tính position khác static)

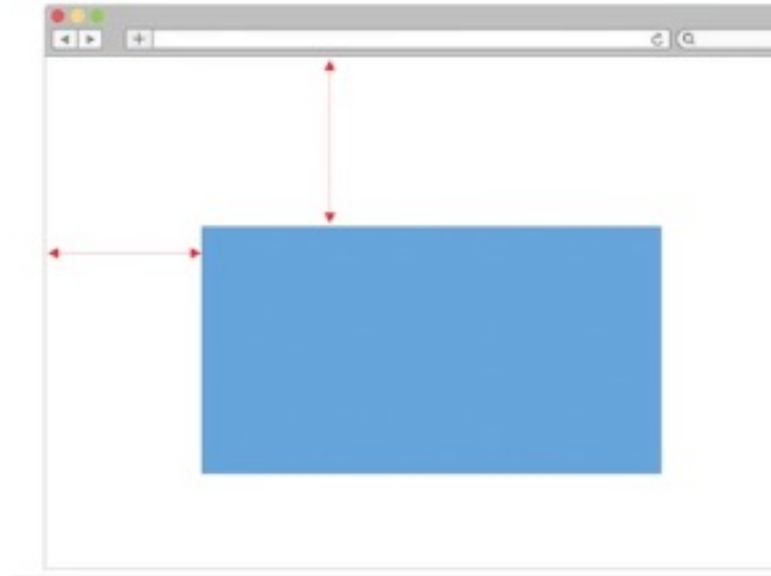
```
div.parent {  
  position: relative;  
}  
div.child {  
  position: absolute;  
  left: 100px;  
  top: 150px;  
}
```



Fixed Position

- Định vị một thành phần so với trình duyệt

```
div {  
  position: fixed;  
  left: 100px;  
  top: 150px;  
}
```



Static Position

- Static Position là mặc định cho các phần tử HTML
- Các phần tử HTML có Position là Static không bị ảnh hưởng bởi các thuộc tính top, bottom, left, right

```
div {  
  position: static;  
  border: 3px solid  
    #73AD21;  
}
```

Thuộc tính visibility

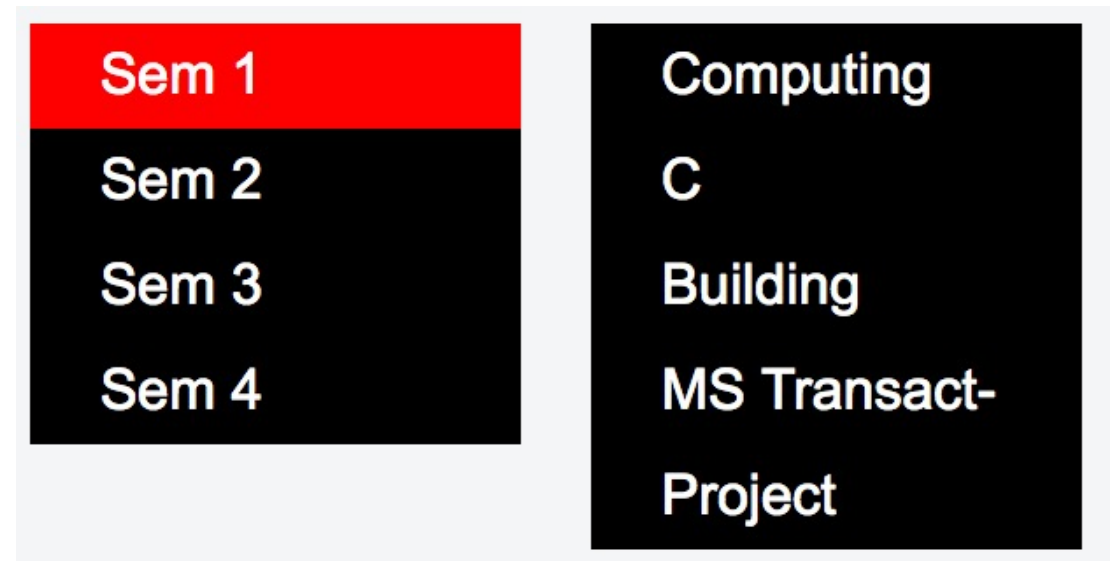
- Thuộc tính Visibility xác định một phần tử HTML có hiển thị hay không

```
h2.a {  
visibility: visible;  
}
```

```
h2.b {  
visibility: hidden;  
}
```

Tạo menu dọc

- Thẻ ``, ``
- Các thuộc tính CSS:
 - `list-style`
 - `position`
 - `width`
 - `height`
 - `padding`
 - `line-height`
 - `cursor`
 - `text-decoration`
 - `visibility`



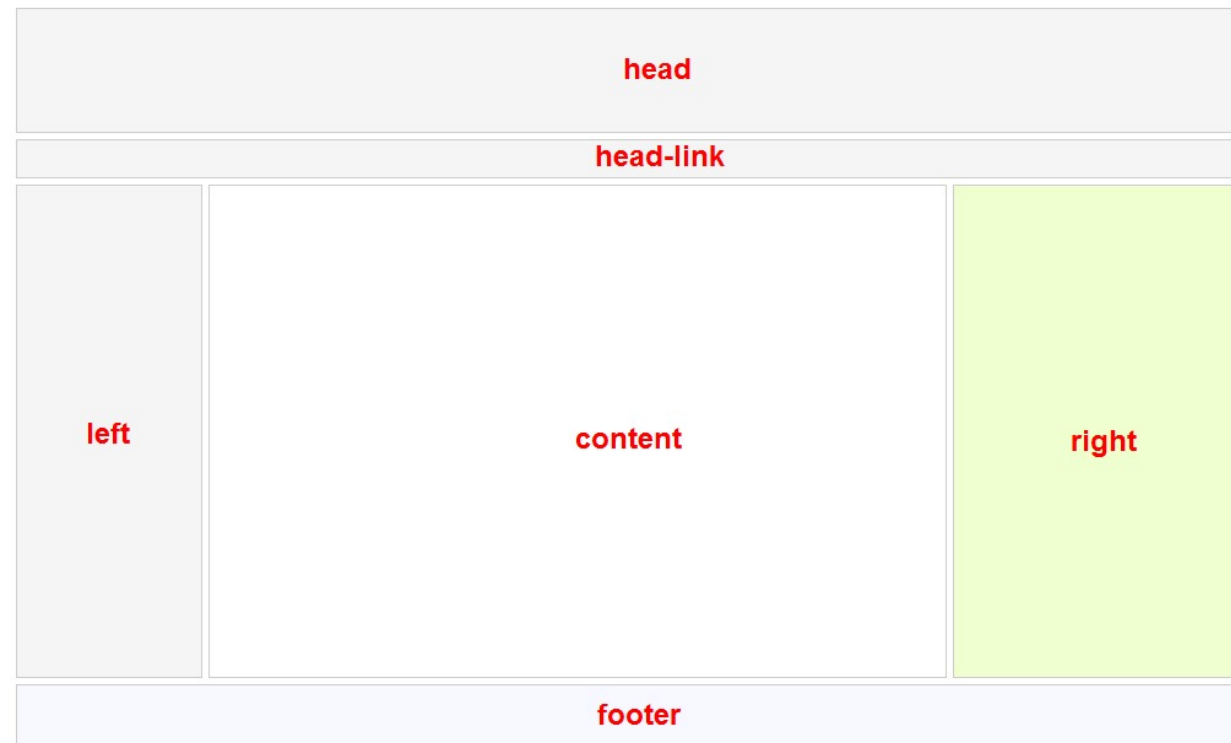
Tạo menu ngang

- Thẻ ``, ``
- Các thuộc tính CSS:
 - `list-style`
 - `position`
 - `width`
 - `height`
 - `padding`
 - `line-height`
 - `cursor`
 - `text-decoration`
 - `visibility`
 - `display`
 - `float`

Sem 1	Sem 2	Sem 3	Sem 4
	Cloud Computing		
	APJ - I		
	APJ - II		
	APJ - III		
	eProject		

Tạo bố cục trang web cơ bản với thẻ div

- Khai báo các thẻ <div>
- Định vị các thẻ <div> sử dụng các thuộc tính CSS:
 - width
 - height
 - margin
 - padding
 - border
 - **clear**
 - **float**



Demo bố cục cơ bản

<https://jsfiddle.net/codegym/j286L0yg/>

Tạo bố cục trang web với các thẻ HTML5

- Các thẻ HTML5:

- `<header>`
- `<nav>`
- `<section>`
- `<article>`
- `<aside>`
- `<footer>`



CODEGYM

Raising the bar

HỆ THỐNG ĐÀO TẠO LẬP TRÌNH HIỆN ĐẠI