



R a i s i n g t h e b a r

Bài 7: Thiết kế bố cục trang Web

Module: JWBD

Trao đổi về bài trước

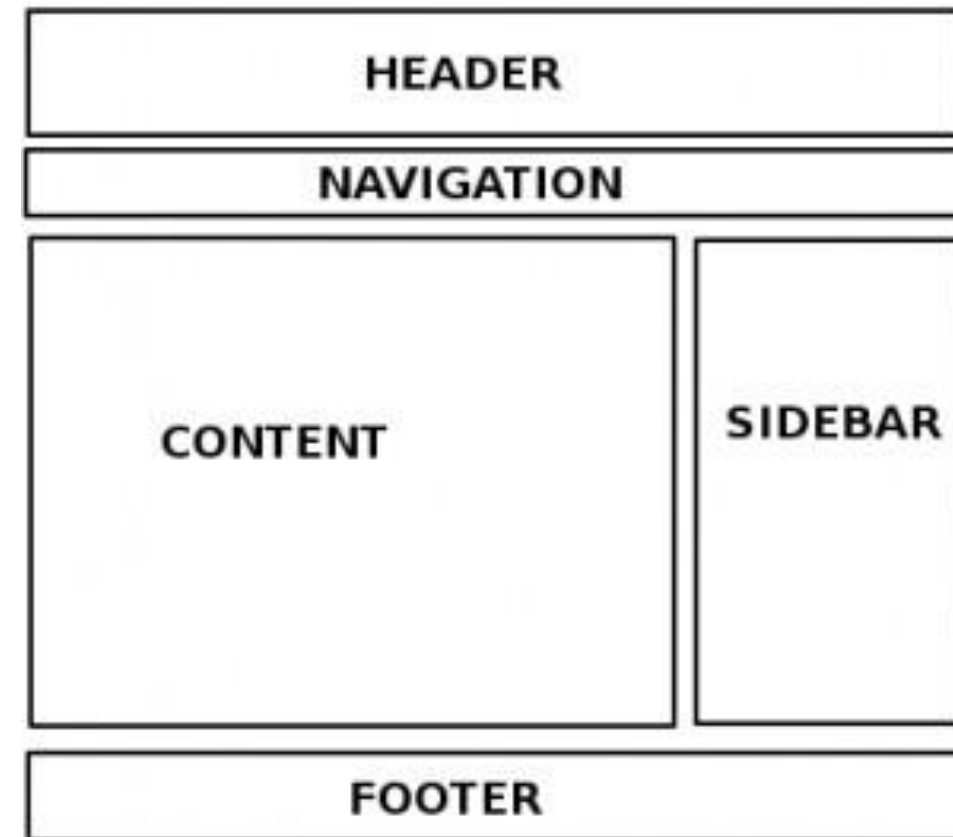
Hỏi và trao đổi về các khó khăn gặp phải trong bài “CSS”
Tóm tắt lại các phần đã học từ bài “CSS”

Mục tiêu

- Trình bày được khái niệm bố cục (layout) của một trang web
- Trình bày được mô hình hộp (box model) trong CSS
- Sử dụng được thuộc tính position thiết kế bố cục trang
- Trình bày được khái niệm Responsive trong thiết kế website
- Trình bày và sử dụng được Grid View trong thiết kế website
- Sử dụng được RWD Media Queries thiết kế bố cục website

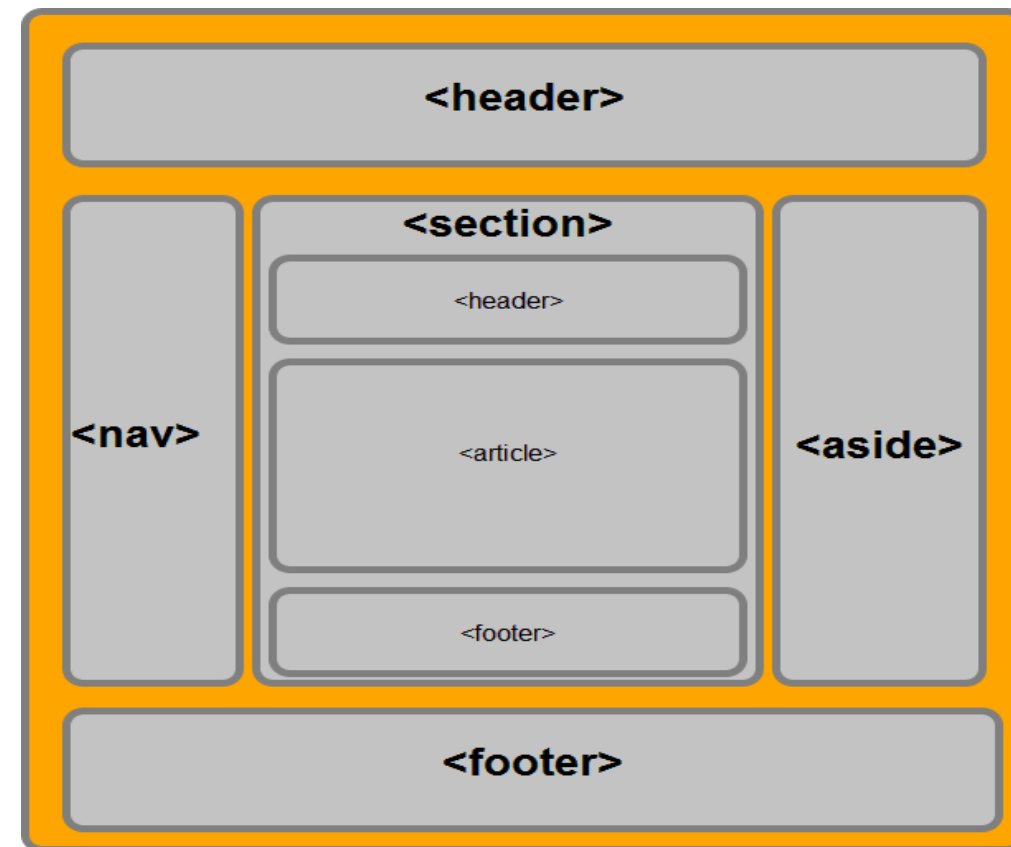
Layout – Bố cục trang web

- Bố cục (layout) của website rất quan trọng, nó quyết định đến việc bố trí các thành phần trên trang có hợp lý, chuyên nghiệp hay không
- Các website hiện nay thường có các thành phần cơ bản và được bố cục như bên:
 - ✓ Header: phần đầu tiên của trang web như logo, banner, menu phụ, ô tìm kiếm...
 - ✓ Navigation: thanh điều hướng hay menu chính
 - ✓ Content: Nội dung chính của website
 - ✓ Sidebar: thanh bên chứa các phần như menu phụ, các bài viết liên quan, quảng cáo...
 - ✓ Footer: Phần chân của trang chứa các thông tin như bản quyền...



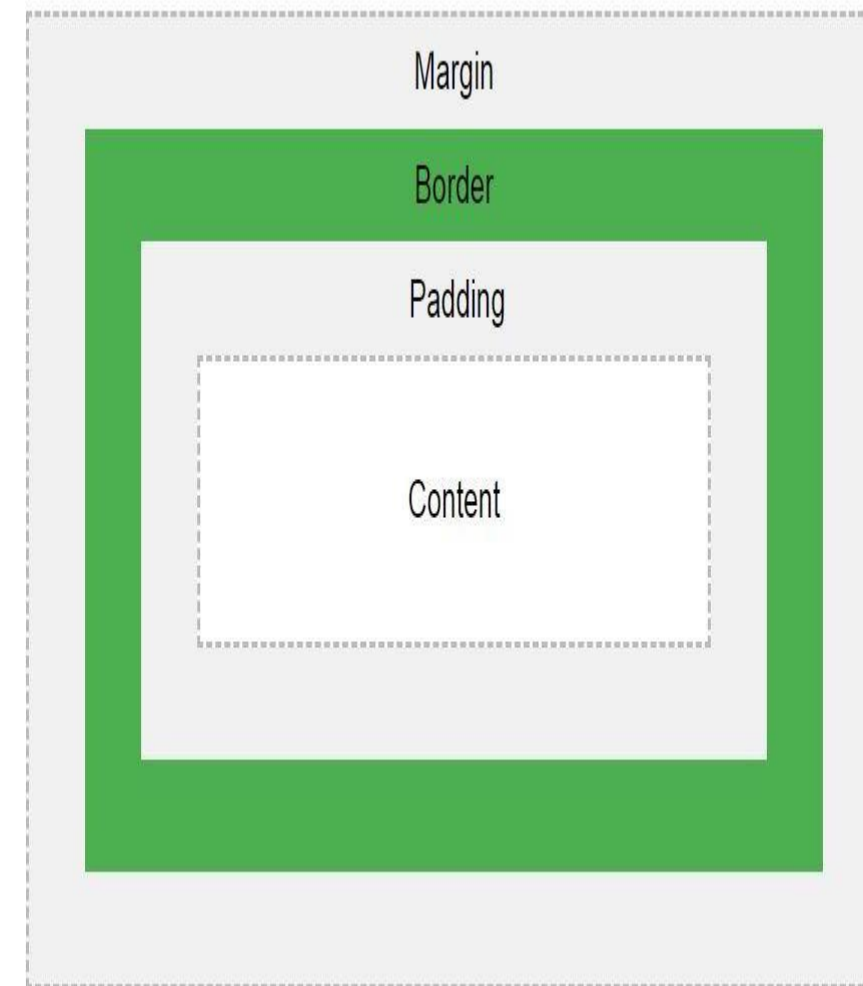
Bố cục trong HTML5

- HTML5 có chứa các thẻ ngữ nghĩa, giúp cơ chế tìm kiếm hiểu sát hơn về cấu trúc website
 - ✓ Thẻ header: định nghĩa phần đầu của trang web
 - ✓ Thẻ nav: thanh điều hướng menu
 - ✓ Thẻ section: định nghĩa 1 phần website
 - ✓ Thẻ article: định nghĩa nội dung bài viết độc lập
 - ✓ Thẻ aside: định nghĩa phần bên cạnh nội dung, như sidebar
 - ✓ Thẻ footer: định nghĩa phần chân của trang



Mô hình hộp trong CSS

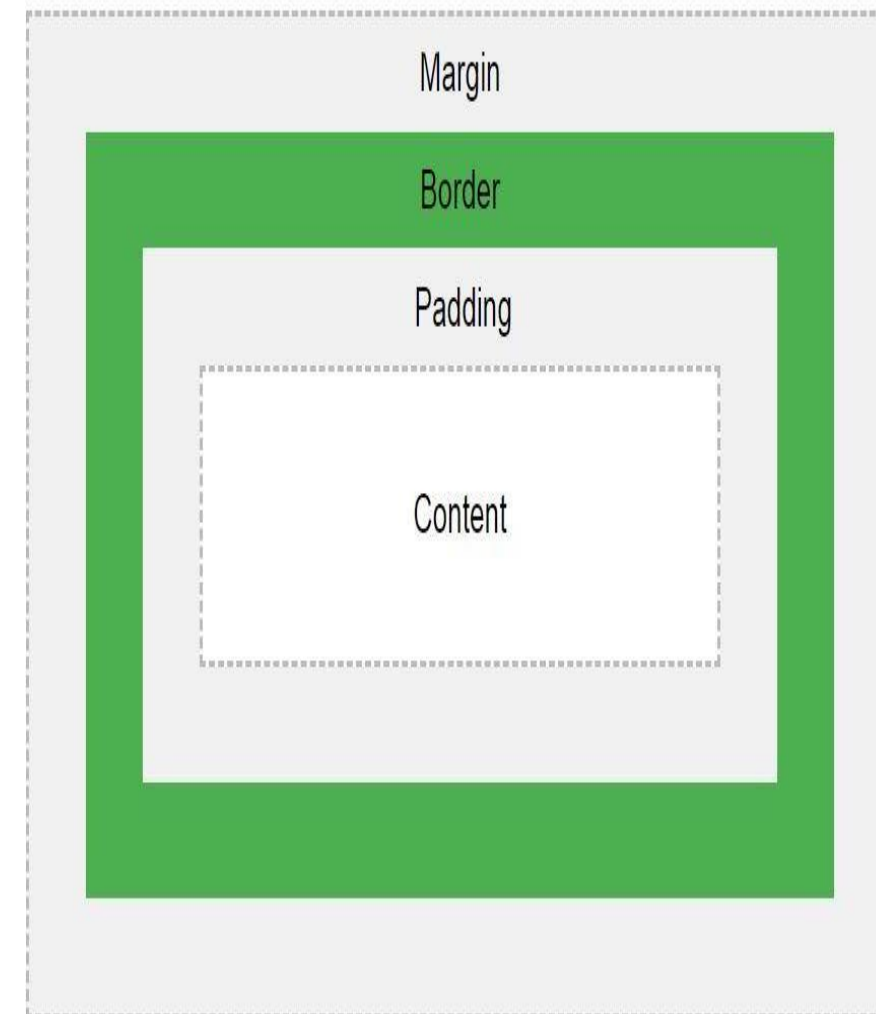
- Mỗi phần tử HTML có thể được coi như là một chiếc hộp
- Trong CSS thuật ngữ mô hình hộp (box model) được dùng để nói về việc thiết kế bố cục (layout)
- Về cơ bản mô hình hộp CSS là một chiếc hộp bao quanh phần tử HTML như:
 - ✓ **Content**: nội dung trong hộp, nơi hiển thị văn bản và hình ảnh
 - ✓ **Padding**: Khoảng cách từ nội dung đến viền, chính là phần khoảng trắng xung quanh nội dung, trong suốt
 - ✓ **Border**: Đường viền xung quanh nội dung và padding
 - ✓ **Margin**: Phần lề bên ngoài đường viền, trong suốt



Mô hình hộp trong CSS

- Ví dụ:

```
div {  
  width: 300px;  
  border: 25px solid green;  
  padding: 25px;  
  margin: 25px;  
}
```
- Độ rộng của phần tử thực tế:
Chiều rộng tổng thể = chiều rộng Content + padding trái + padding phải + đường viền trái + đường viền phải + margin trái + margin phải
- Chiều cao của phần tử thực tế:
Chiều cao tổng thể = chiều cao Content + padding trên + padding dưới + đường viền trên + đường viền dưới + margin trên + margin dưới



Thuộc tính position của CSS

- Thuộc tính position của CSS quy định cách thức xếp đặt vị trí của một thành phần trong trang web, hay nói cách khác position chỉ định phương thức định vị sẽ áp dụng cho một phân tử HTML
- Có 4 giá trị khác nhau cho thuộc tính này: static, relative, fixed, absolute
- Các phần tử sau đó được định vị bằng cách sử dụng các thuộc tính top, bottom, left, right. Các thuộc tính định vị này làm việc khác nhau phụ thuộc vào giá trị của position
- Cú pháp:

```
selector {  
    position: giá trị;  
}
```


Position:static

- Các phần tử HTML mặc định được định vị là static (tĩnh)
- Khi một phần tử là static điều đó có nghĩa là phần tử này sẽ được sắp xếp đúng theo vị trí tự nhiên của nó
- Các phần tử được định vị static không bị ảnh hưởng bởi các thuộc tính top, bottom, left, right
- Một phần tử định vị static thì luôn luôn được định vị theo luồng (flow) bình thường của trang, tức các thành phần sẽ nằm theo thứ tự trong văn bản

- Ví dụ:

```
#two {  
    background: green;  
}
```

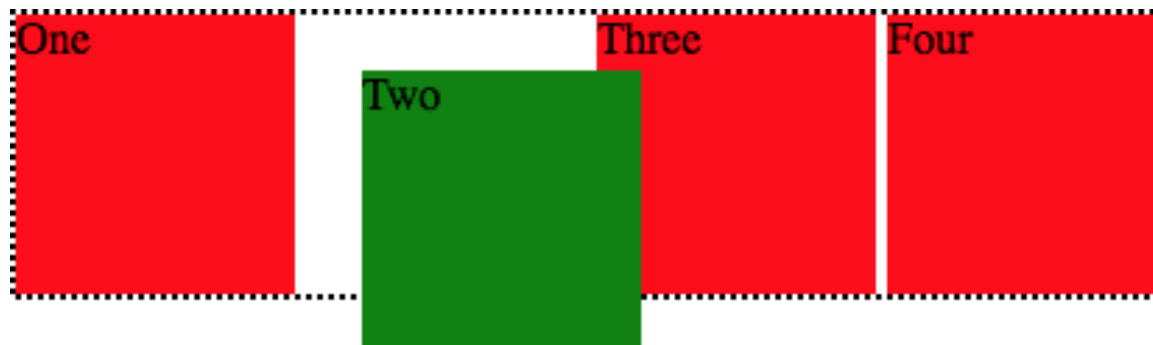


Position:relative

- Một phần tử với Position:relative được định vị tương đối so với vị trí bình thường của nó, hay nói cách khác relative sẽ định vị một thành phần so với vị trí mặc định của chính bản thân nó
- Thiết lập cho các thuộc tính định vị top, right, bottom, left cho một phần tử là relative là nguyên nhân khiến nó bị điều chỉnh ra khỏi vị trí bình thường của nó

- Ví dụ:

```
#two {  
    top: 20px;  
    left: 20px;  
    background: green;  
    position: relative;  
}
```



Position: absolute

- Định vị một thành phần so với vị trí của thành phần cha (tổ tiên) gần nhất của nó (thành phần cha phải có thuộc tính position khác static)
- Nếu một phần tử được định vị absolute không có các phần tử tổ tiên được định vị, nó sử dụng phần tử, và di chuyển cùng với di chuyển của trang

- Ví dụ:

```
#two {  
    top: 20px;  
    left: 20px;  
    background: green;  
    position: absolute;  
}
```



Position:fixed

- Định vị một thành phần so với trình duyệt
- Một phần tử với position:fixed được định vị tương đối so với chế độ xem (viewport), có nghĩa là nó luôn luôn ở cùng một vị trí ngay cả khi trang được cuộn
- Các thuộc tính top, right, bottom, left được sử dụng để định vị phần tử
- Một phần tử fixed không để lại một khoảng trống trong trang nơi nó thường được định vị
- Ví dụ:

```
#two {  
    position: fixed;  
    left: 150px;  
    top: 100px;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```

Responsive Web Design - RWD

- RWD (Thiết kế web thích ứng) là kỹ thuật được sử dụng để trang web có thể hiển thị tốt trên nhiều loại thiết bị với kích thước của trình duyệt khác nhau: máy tính để bàn, máy tính bảng, điện thoại...
- Các trang web sẽ không bỏ bớt các thông tin để phù hợp với các thiết bị nhỏ hơn, nhưng sẽ thích ứng nội dung của nó để phù hợp với bất kỳ thiết bị nào
- Một trang web được gọi là thiết kế responsive khi bạn sử dụng CSS và HTML để thay đổi kích cỡ, ẩn, thu nhỏ, phóng to, hoặc di chuyển nội dung để làm cho nó có giao diện đẹp trên bất kỳ màn hình nào



Grid View

- Grid-view phân bố nội dung theo dạng một lưới, nghĩa là các trang được chia thành các cột/hàng
- Giúp cho việc thiết kế trở nên dễ dàng hơn, dễ đặt nội dung lên các vùng phù hợp của trang web
- Một Responsive Grid-View thường có 12 cột, chiếm 100% chiều rộng của trang. Các cột sẽ được thu nhỏ hay mở rộng theo kích thước của cửa sổ trình duyệt
- Các khái niệm:
 - ✓ Grid Container
 - ✓ Grid Line
 - ✓ Grid Cell
 - ✓ Grid Area
 - ✓ Grid Track



RWD Media Queries

- Media Queries (Truy vấn phương tiện) là một kỹ thuật của CSS, cho phép kết xuất nội dung để thích ứng với các điều kiện như kích thước và độ phân giải màn hình
- Là công nghệ nền tảng của thiết kế web đáp ứng RWD

```
/* (1) Màn hình nhỏ để đứng (phones, 600px và nhỏ hơn) */  
@media only screen and (max-width: 600px) {...}  
  
/* (2) Màn hình nhỏ để ngang (phones 600px và lớn hơn) */  
@media only screen and (min-width: 600px) {...}  
  
/* (3) Màn hình cỡ thường để đứng (tablets, 768px và lớn hơn) */  
@media only screen and (min-width: 768px) {...}  
  
/* (4) Màn hình lớn (laptops/desktops, 992px và lớn hơn) */  
@media only screen and (min-width: 992px) {...}  
  
/* (5) Màn hình lớn với độ phân giải cao (màn hình 1200px, 2k, 4k, 8k và lớn hơn) */  
@media only screen and (min-width: 1200px) {...}
```



Tổng kết

- Bố cục (layout) của website rất quan trọng, nó quyết định đến việc bố trí các thành phần trên trang web có hợp lý, chuyên nghiệp hay không
- Trong CSS thuật ngữ mô hình hộp (box model) được dùng để nói về việc thiết kế bố cục (layout). Mỗi phần tử HTML có thể được coi như là một chiếc hộp
- Thuộc tính position của CSS quy định cách thức xếp đặt vị trí của một thành phần trong trang web
- RWD (Thiết kế web thích ứng) là kỹ thuật được sử dụng để trang web có thể hiển thị tốt trên nhiều loại thiết bị với kích thước và độ phân giải khác nhau
- Grid-view phân bố nội dung theo dạng một lưới, nghĩa là các trang được chia thành các cột/hàng. Điều đó giúp cho việc thiết kế trở nên dễ dàng hơn, dễ đặt nội dung lên các vùng phù hợp của trang web
- Media Queries (Truy vấn phương tiện) là một kỹ thuật của CSS, là công nghệ nền tảng của thiết kế web đáp ứng RWD

Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: Bootstrap



R a i s i n g t h e b a r