HTML5 Layout

Ba Nguyễn

Semantic vs Non-Semantic Elements

Một phần tử ngữ nghĩa *(semantic)* mô tả rõ ràng ý nghĩa của nội dung bên trong nó cho trình duyệt, bộ máy phân tích, lập trình viên, ...

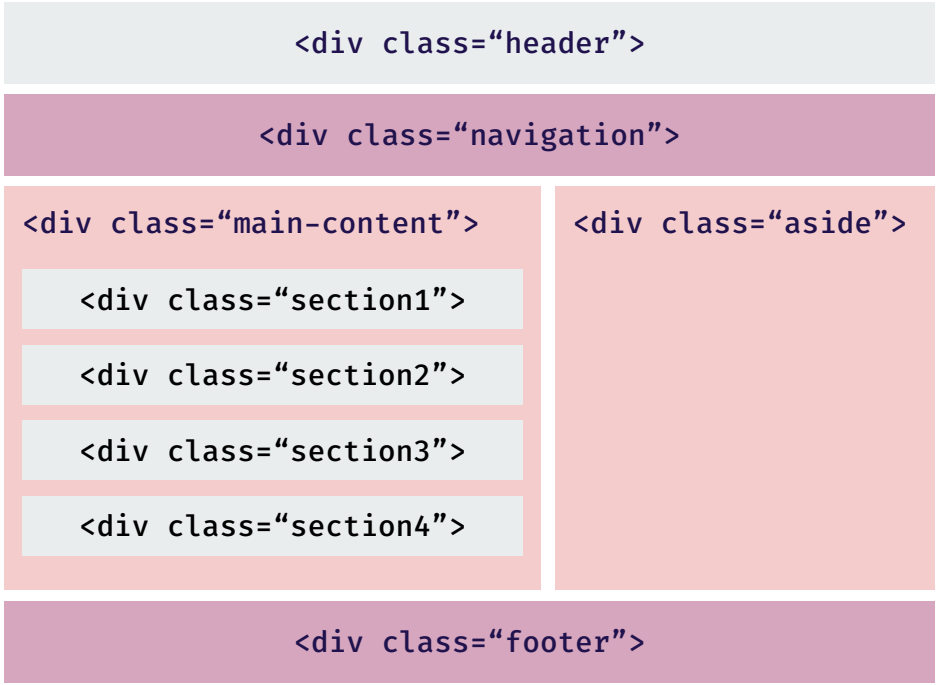
- Các phần tử **non-semantic** như `<div>`, `<span>` không cho biết bất kỳ ý nghĩa nào về nội dung bên trong nó
- Các phần tử **semantic** như `<header>`, `<footer>`, `<main>` mô tả rõ ràng ý nghĩa nội dung bên trong nó

HTML5 cung cấp một số phần tử ngữ nghĩa để xác định các phần nội dung khác nhau của trang web, mỗi phần thường chứa những nội dung riêng biệt

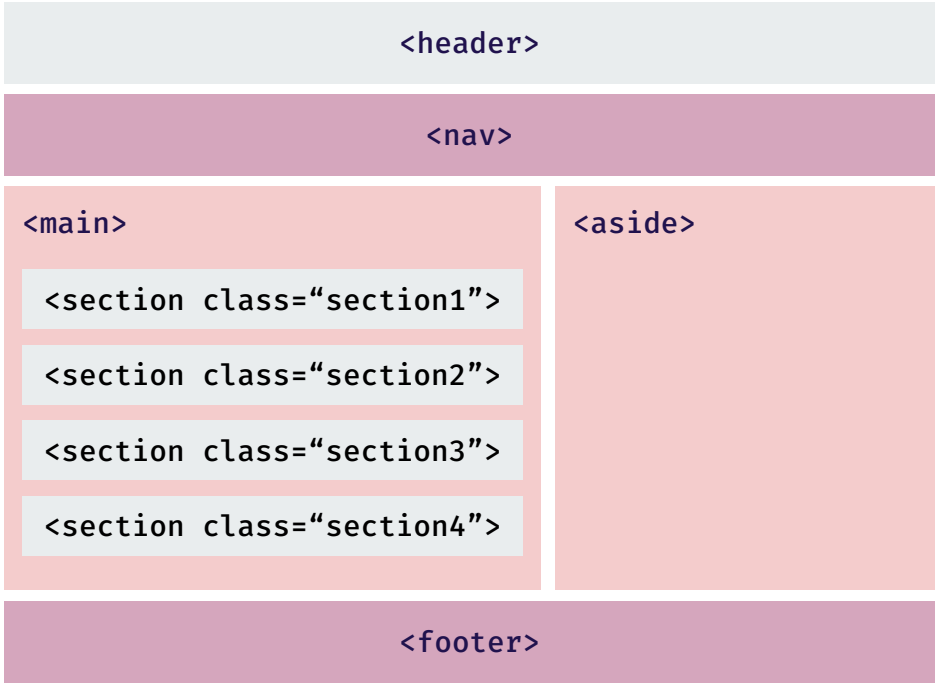
💡 Việc tổ chức nội dung thành các phần khác nhau, kết hợp với CSS cho phép tạo bố cục đẹp mắt cho trang web

💡 Tham khảo thêm về semantic elements: [w3school.com/html5_semantic]

HTML4 Layout



HTML5 Layout



CSS Box Model

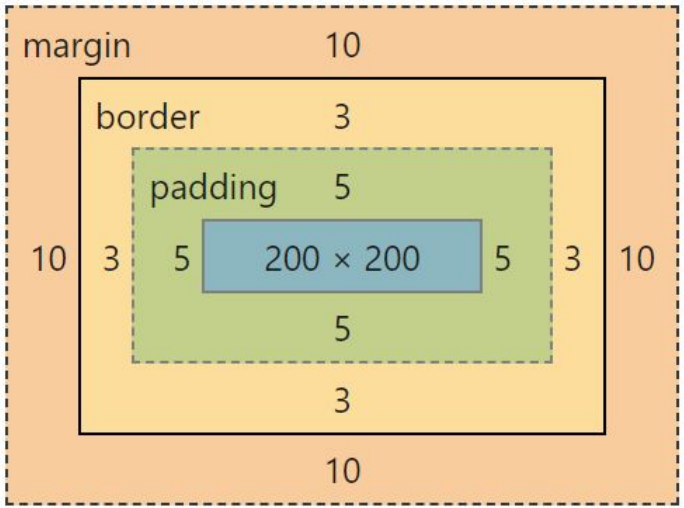
Ba Nguyễn

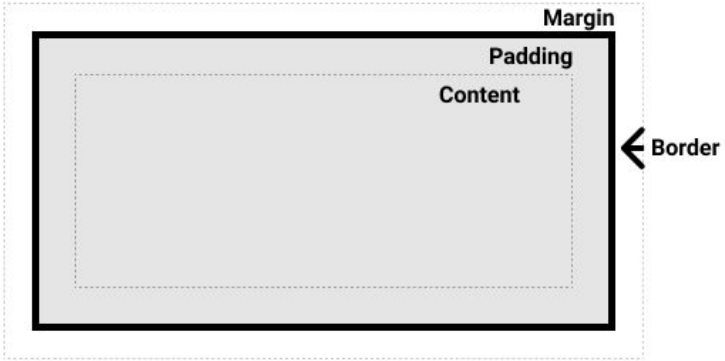
Boxes

Trong CSS, mỗi phần tử HTML đều được đặt trong 1 *cái hộp (box)* của riêng nó. Box có các thuộc tính đặc biệt, việc tạo bố cục cho trang web chủ yếu dựa trên việc sắp xếp vị trí hiển thị và kích thước các box đó.

Mỗi phần tử có một box riêng, nó có thể chứa các box khác (phần tử con), hay được đặt trong một box lớn hơn (phần tử cha)

Boxes





Box Properties

```css
/* CSS Properties thường dùng với boxes */
.box {
    box-sizing: border-box;
    display: block; /* inline, ... */
    width: 100px; /* px, %, em, ... */
    height: 100px; /* px, %, em, ... */
    padding: 15px; /* px, %, em, ... */
    border: 2px solid #ccc; /* width style color */
    margin: 30px; /* px, %, em, ... */
    /* X-axis, Y-axis, Blur, */
    box-shadow: 0 5px 20px rgba(0, 0, 0, 0.1);
    overflow: hidden;
    background: #fafafa;
}
```

Display

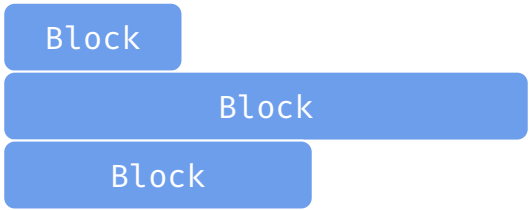
Thuộc tính `display` thay đổi cách hiển thị *(kiểu box)* của phần tử trên trang. Thuộc tính `display` có thể nhận rất nhiều giá trị với các đặc tính khác nhau. Nó rất quan trọng trong việc thiết lập bố cục cho trang web

```
.box {
    display: inline;
    display: block;
    display: inline-block;
    display: flex;
    display: none;
    /* Có một thuộc tính khác để ẩn phần tử */
    /* Phần tử không được nhìn thấy, nhưng vẫn chiếm
    kích thước trên trang */
    visibility: hidden;
}
```

Block

Các đặc tính của **block boxes**:

- Mỗi **box** này sẽ được đặt trên một dòng mới
- Mặc định **box** sẽ chiếm trọn chiều rộng vùng chứa nó
- Các thuộc tính `width` và `height` có thể áp dụng để đặt kích thước cho phần tử
- Các thuộc tính `padding` và `margin` sẽ đẩy các phần tử khác ra xa



Inline

Các đặc tính của **inline boxes**:

- Mỗi **box** này sẽ *không* được đặt trên một dòng mới
- **Box** sẽ có chiều rộng vừa với kích thước nội dung
- Các thuộc tính `width` và `height` *không có tác dụng*
- Các thuộc tính `padding` và `margin` theo hướng trái phải (chiều ngang) đẩy các phần tử khác ra xa
- Các thuộc tính `padding` và `margin` theo hướng trên dưới (chiều dọc) có tác dụng nhưng *không đẩy các phần tử khác ra xa*
- Mỗi box được coi như 1 từ trong văn bản, khoảng trắng ở giữa được xử lý giống như text (`white-space: wrap`)

Inline Inline Inline Inline

Inline-Block

Các đặc tính của **inline-block boxes**:

- Kết hợp các đặc tính của **block** và **inline**
- Mỗi **box** sẽ *không* được đặt trên một dòng mới (tương tự **inline**)
- Mặc định **Box** sẽ có chiều rộng vừa với kích thước nội dung (tương tự **inline**)
- Các thuộc tính `width` và `height` có thể áp dụng để đặt kích thước cho phần tử (tương tự **block**)
- Các thuộc tính `padding` và `margin` sẽ đẩy các phần tử khác ra xa (tương tự **block**)
- Lưu ý về khoảng cách giữa các phần tử (tương tự **inline**)
- `margin: auto;` không có tác dụng như **block**

Inline-block Inline-block Inline-block

Box Sizing

Mặc định, thuộc tính `box-sizing` có giá trị `content-box`, khi đó:

- Thuộc tính `width`, `height` sẽ chỉ áp dụng cho phần nội dung bên trong
- Kích thước tổng của phần tử sẽ bằng `width + height + padding + border`

Thuộc tính `box-sizing` nên được đặt thành `border-box`, khi đó:

- Kích thước tổng của phần tử sẽ chính bằng giá trị của `width`, `height`, việc căn chỉnh bố cục trên trang sẽ dễ dàng hơn
- Kích thước phần nội dung sẽ tự điều chỉnh bằng `width/height - padding - border`

```
.box {
    box-sizing: border-box;
    width: 100px;
    padding: 10px;
    border: 2px solid transparent;
}
```

Shorthand Properties

Một số thuộc tính là cú pháp viết tắt cho nhiều thuộc tính khác (VD: `margin`, `border`, ...), chúng có thể nhận 1 hoặc nhiều giá trị (tùy thuộc tính), các giá trị sẽ tự động được gán cho thuộc tính cụ thể nếu phù hợp.

```
.box {
    margin: 10px 20px 30px 40px;
    /* margin-top: 10px;
       margin-right: 20px;
       margin-bottom: 30px;
       margin-left: 40px; */
}
```

```
.box {
  margin: ▼ 10px 20px 30px 40px;
    margin-top: 10px;
    margin-right: 20px;
    margin-bottom: 30px;
    margin-left: 40px;
}
```

CSS Tip

```css
.box {
    /* Căn giữa phần tử theo chiều ngang */
    /* Yêu cầu kiểu hiển thị block */
    display: block;
    width: 900px; /* và giá trị width cho block */
    margin-left: auto;
    margin-right: auto;

    /* Căn giữa nội dung bên trong theo chiều dọc */
    padding: 100px;
}
```

Page Layout

Sử dụng `inline-block` tạo bố cục đơn giản

```
/* Tips */
.box {
    /* Xóa khoảng trắng giữa item */
    word-spacing: -1ch;
}

.item {
    /* Reset thuộc tính word-spacing */
    word-spacing: normal;
    display: inline-block;
}
```