



Lesson 04

Box Model, Display, Alignment, Media Queries

Module 01: WEB DESIGN



Mục tiêu

- Hiểu được khái niệm Box Model
- Sử dụng được các CSS Padding
- Sử dụng được các CSS Margin
- Sử dụng được các CSS Display
- Sử dụng được các CSS Visibility
- Sử dụng được các CSS Alignment
- Sử dụng được các CSS Border Radius
- Sử dụng được các Media Queries



Khái niệm Box Model

Giới thiệu:

- ◆ **Box Model** là một kỹ thuật cơ bản nhất trong CSS Layout và được **sử dụng để mô tả về khoảng cách** mà **mỗi phần tử trên website** được **sở hữu**.
- ◆ Là kỹ thuật tinh chỉnh khoảng cách hiển thị cho mỗi phần tử trên website.
- ◆ **Box Model** bao gồm **4 phần** quan trọng đó là: **Margin, Border, Padding, Content**

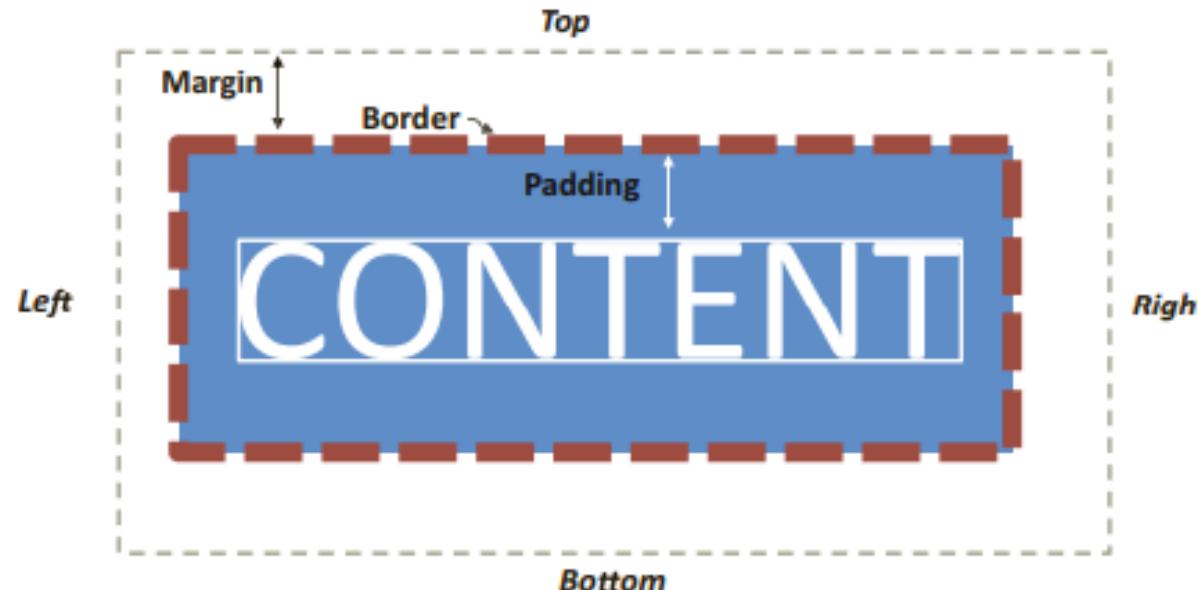


CSS Padding



☐ Một số khai báo CSS về khoảng cách thường dùng:

- ◆ **padding**: là khoảng cách từ đường viền của 1 phần tử tới nội dung nằm bên trong phần tử đó. Ví dụ: **padding: 35px;**
- ◆ Một số thuộc tính của **padding**:
 - **padding-top** (vùng đệm phía trên)
 - **padding-bottom** (vùng đệm phía dưới)
 - **padding-right** (vùng đệm phía bên phải)
 - **padding-left** (vùng đệm phía bên trái)



CSS Padding



☐ Một số khai báo CSS về khoảng cách thường dùng:

- ◆ Thuộc tính **padding** có **4 cách để canh chỉnh** như ví dụ dưới đây:

The screenshot shows a code editor and a browser developer tools panel. The code editor displays a file named 'PaddingProperty.html' with the following content:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Padding Property</title>
    <style>
      /* Cách 1 */
      div {
        padding-top: 10px;
        padding-right: 10px;
        padding-bottom: 10px;
        padding-left: 10px;
      }
      /* Cách 2 */
      div {
        padding: 20px; /* Chính padding cho cả 4 phía */
      }
      /* Cách 3 */
      div {
        padding: 10px 20px; /* 2 cặp giá trị lần lượt là top, bottom và left, right */
      }
      /* Cách 4 */
      div [
        padding: 5px 10px 15px 20px; /* Chính padding cho cả 4 phía: top, bottom, left, right */
      ]
    </style>
  </head>
  <body>
    <!-- Giới thiệu -->
    <div>
      Định nghĩa một chân dung lập trình viên mới - một lập trình viên đáp ứng ĐÚNG nhu cầu của Doanh nghiệp và thị trường CNTT – Fast Track đào tạo xoay quanh ba trụ cột Kỹ năng lập trình – Tiếng Anh giao tiếp – Tinh chuyên nghiệp trong công việc với nhiều khóa học khác nhau phù hợp với lộ trình của nhiều đối tượng học viên.
    </div>
  </body>
</html>
```

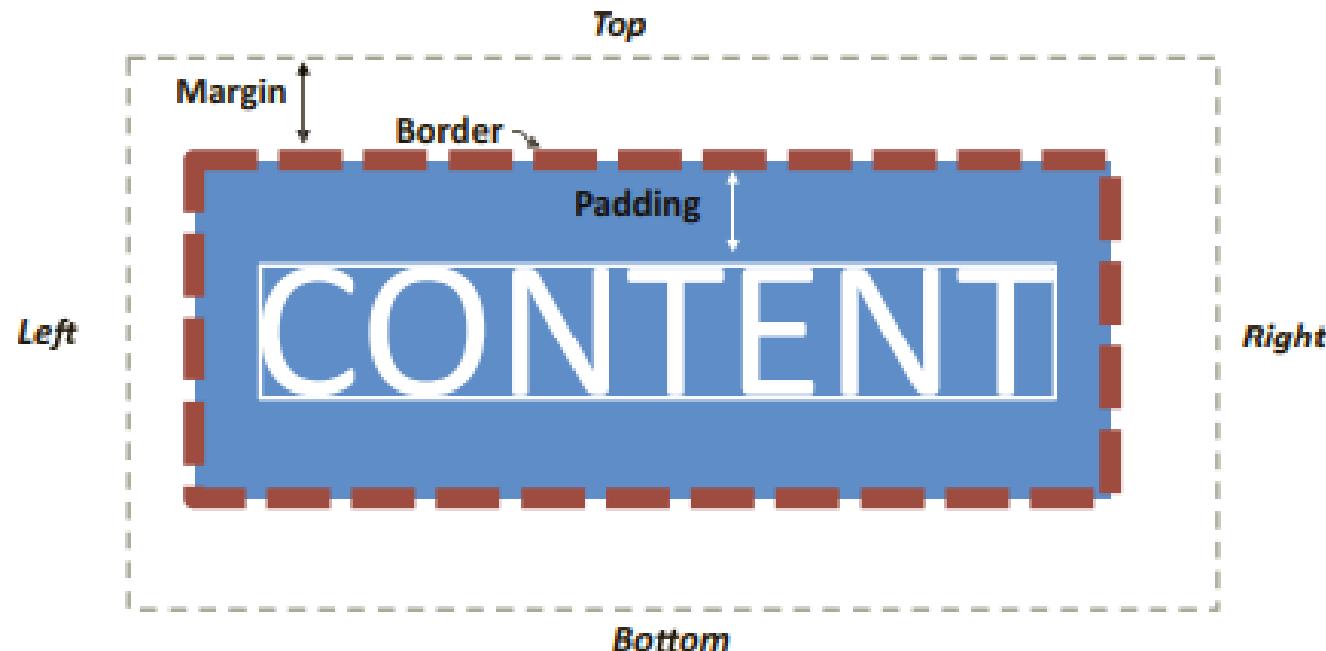
The developer tools panel shows a preview of the page with the following description:

Định nghĩa một chân dung lập trình viên mới – một lập trình viên đáp ứng ĐÚNG nhu cầu của Doanh nghiệp và thị trường CNTT – Fast Track đào tạo xoay quanh ba trụ cột Kỹ năng lập trình – Tiếng Anh giao tiếp – Tinh chuyên nghiệp trong công việc với nhiều khóa học khác nhau phù hợp với lộ trình của nhiều đối tượng học viên.

CSS Margin

☐ Một số khai báo CSS về khoảng cách thường dùng:

- ◆ **margin**: là khoảng cách giữa các phần tử HTML với nhau. Ví dụ: **margin: 15px;**
- ◆ Một số thuộc tính của **margin**:
 - **margin-top** (canh lề phía trên)
 - **margin-bottom** (canh lề phía dưới)
 - **margin-right** (canh lề phía bên phải)
 - **margin-left** (canh lề phía bên trái)



CSS Margin



☐ Một số khai báo CSS về khoảng cách thường dùng:

- ♦ Thuộc tính **margin** với giá trị **auto** sẽ tự động canh đều 2 bên của phần tử được thiết lập auto. Ví dụ: **margin: auto;**

```
MarginAuto.html
MarginAuto.html > html > head > style
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Margin Auto</title>
5          <style>
6              div {
7                  width: 300px;
8                  margin: auto;
9                  border: 1px solid red;
10             }
11         </style>
12     </head>
13     <body>
14         <h2>Dùng margin:auto</h2>
15
16         <p>
17             Bạn có thể đặt thuộc tính margin thành tự động (auto) để căn giữa phần tử trong vùng chứa của nó theo chiều ngang.
18             Sau đó, phần tử sẽ chiếm chiều rộng được chỉ định và không gian còn lại sẽ được chia đều giữa lề trái và lề phải.
19         </p>
20
21         <div>
22             Nội dung của Div này sẽ được căn giữa theo chiều ngang vì nó có margin: auto;
23         </div>
24     </body>
25 </html>
```

CSS Margin



Một số khai báo CSS về khoảng cách thường dùng:

- ♦ Thuộc tính **margin** với giá trị **auto** sẽ tự động canh đều 2 bên của phần tử được thiết lập auto. Ví dụ: **margin: auto;**

The screenshot shows a web browser window titled "Margin Auto". The address bar displays the URL "127.0.0.1:5501/MarginAuto.html". The main content area contains the following text:

Dùng margin:auto

Bạn có thể đặt thuộc tính margin thành tự động (auto) để căn giữa phần tử trong vùng chứa của nó theo chiều ngang. Sau đó, phần tử sẽ chiếm chiều rộng được chỉ định và không gian còn lại sẽ được chia đều giữa lề trái và lề phải.

Nội dung của Div này sẽ được căn giữa theo chiều ngang vì nó có margin: auto;

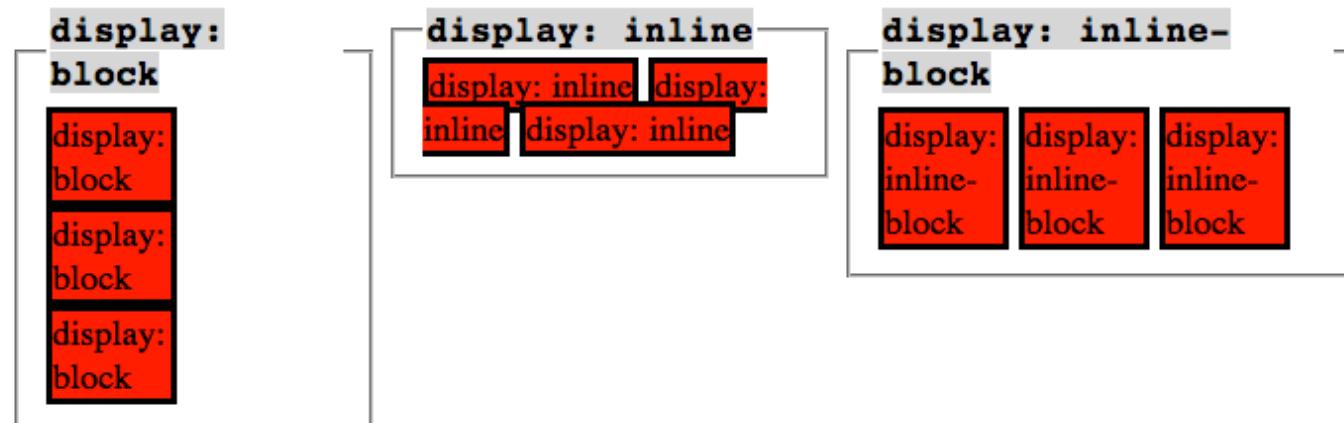
CSS Display

☐ Một số khai báo CSS về cách hiển thị (display) thường dùng:

- ◆ **block**: hiển thị thành một khối, mỗi khối 1 dòng. Ví dụ: **display: block;**
- ◆ **inline**: hiển thị nội tuyến, các phần tử đều nằm trên 1 dòng. Ví dụ: **display: inline;**
- ◆ **inline-block**: kết hợp 2 thuộc tính block và inline. Ví dụ: **display: inline-block;**
- ◆ **none**: ẩn phần tử đi, không tồn tại phần tử đó nữa. Ví dụ: **display: none;**

block vs inline vs inline-block

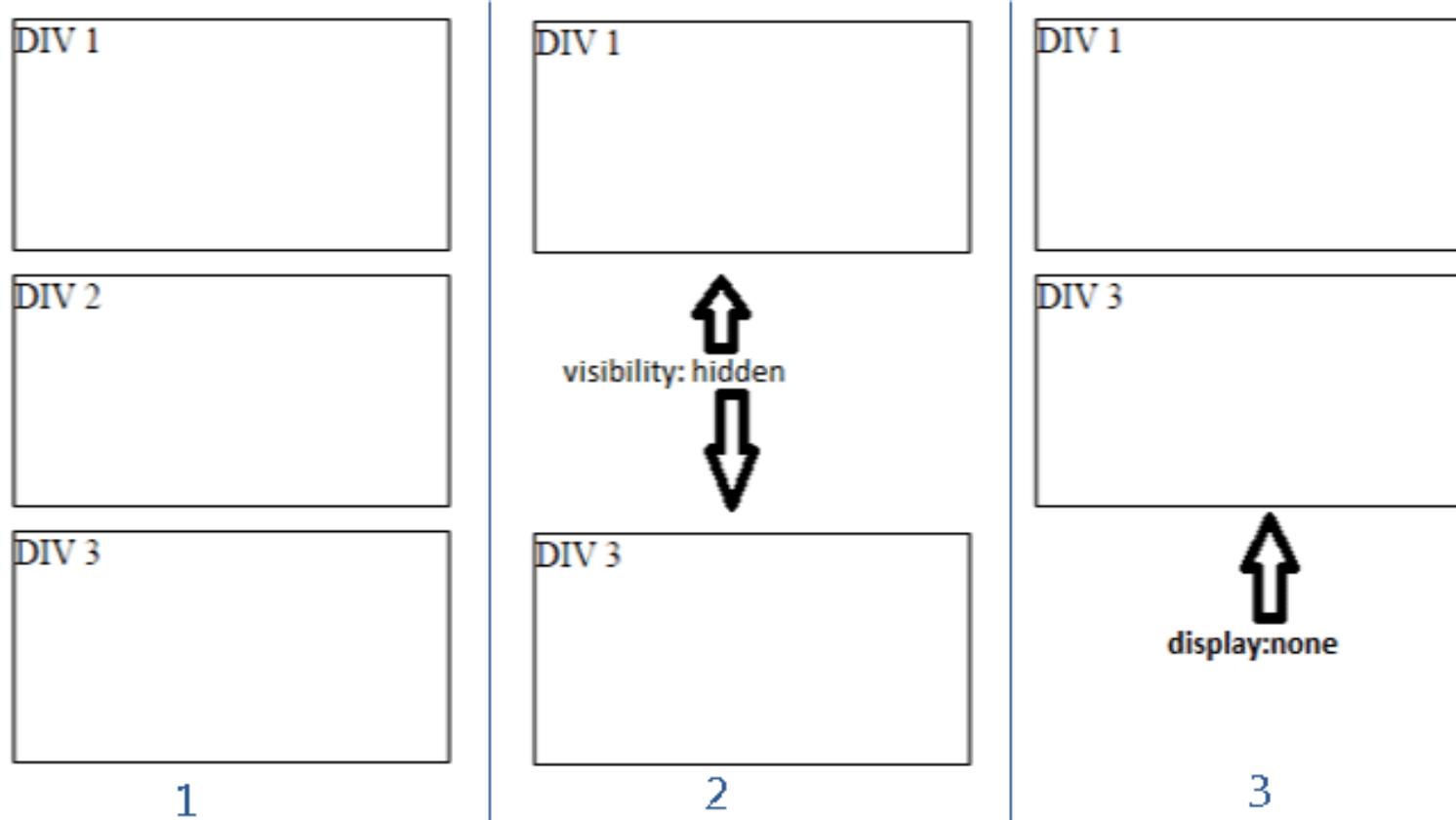
Below are a bunch of `<div style="width: 50px" ...>` with different `display:` settings.



CSS Display

Một số khai báo CSS về cách hiển thị (display) thường dùng:

- ◆ **none**: ẩn phần tử đi, không tồn tại phần tử đó nữa. Ví dụ: **display: none;**
- ◆ **visibility**: ẩn phần tử đi, vẫn tồn tại phần tử đó nữa. Ví dụ: **visibility: hidden;**



CSS Display



☐ Một số khai báo CSS về cách hiển thị (display) thường dùng:

- ◆ Ví dụ: Lần lượt sử dụng các thuộc tính display của css và quan sát kết quả:

```
<body>
    <h1>Các thuộc tính Display</h1>

    <h2>display: none</h2>
    <div>
        | Xin chào <p class="ex1">TẤT CẢ CÁC BẠN!</p>Chào mừng các bạn đến với môn Thiết kế Web tại VUS.
    </div>

    <h2>display: inline</h2>
    <div>
        | Xin chào <p class="ex2">TẤT CẢ CÁC BẠN!</p>Chào mừng các bạn đến với môn Thiết kế Web tại VUS.
    </div>

    <h2>display: block</h2>
    <div>
        | Xin chào <p class="ex3">TẤT CẢ CÁC BẠN!</p>Chào mừng các bạn đến với môn Thiết kế Web tại VUS.
    </div>

    <h2>visibility: hidden</h2>
    <div>
        | Xin chào <p class="ex4">TẤT CẢ CÁC BẠN!</p>Chào mừng các bạn đến với môn Thiết kế Web tại VUS.
    </div>
</body>
```

CSS Display



☐ Một số khai báo CSS về cách hiển thị (display) thường dùng:

- ◆ Ví dụ: Lần lượt sử dụng các thuộc tính display của css và quan sát kết quả:

The screenshot shows a browser window titled "Css Display" with the URL "127.0.0.1:5501/CssDisplay.html". The page content is as follows:

```
<style>
  p {
    color: red;
  }

  p.ex1 {
    display: none;
  }

  p.ex2 {
    display: inline;
  }

  p.ex3 {
    display: block;
  }

  p.ex4 {
    visibility: hidden;
  }
</style>
```

Các thuộc tính Display

display: none
Xin chào Chào mừng các bạn đến với môn Thiết kế Web tại VUS.

display: inline
Xin chào TẤT CẢ CÁC BẠN! Chào mừng các bạn đến với môn Thiết kế Web tại VUS.

display: block
Xin chào
TẤT CẢ CÁC BẠN!
Chào mừng các bạn đến với môn Thiết kế Web tại VUS.

visibility: hidden
Xin chào
Chào mừng các bạn đến với môn Thiết kế Web tại VUS.

CSS Display



☐ Một số khai báo CSS về cách hiển thị (list) thường dùng:

- ◆ **list-style:** dùng để thay đổi kiểu đánh dấu cho danh sách. Ví dụ: **list-style: circle;**

The image shows a split-screen view. On the left is a code editor window titled "ListStyleProperty.html" containing the following HTML and CSS code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>List Style Property</title>
5     <style>
6       ul {
7         list-style: circle;
8       }
9
10      /* ul {
11        list-style-image: url('https://w...');
12      }*/
13
14      /* ul {
15        list-style-position: outside;
16      }*/
17    </style>
18  </head>
19  <body>
20    <h2>Fast Track Course</h2>
21    <ul>
22      <li>Frontend ReactJS</li>
23      <li>Backend Java</li>
24      <li>Backend PHP</li>
25    </ul>
26  </body>
27</html>
```

On the right is a browser window titled "List Style Property" showing the rendered output:

Fast Track Course

- Frontend ReactJS
- Backend Java
- Backend PHP

CSS Display



☐ Một số khai báo CSS về cách hiển thị (list) thường dùng:

- ◆ **list-style-image:** sử dụng hình ảnh kiểu đánh dấu.

The screenshot displays a code editor and a browser window side-by-side. The code editor on the left shows the HTML and CSS code for a list style property. The browser window on the right shows the resulting page with a heading 'Fast Track Course' and an ordered list below it. The list items are represented by small purple squares followed by their respective text labels: 'Frontend ReactJS', 'Backend Java', and 'Backend PHP'. The browser's address bar shows the URL 127.0.0.1:5501/ListStyl... .

```
<!-- ListStyleProperty.html -->
<html>
  <head>
    <title>List Style Property</title>
    <style>
      /* ul {
         list-style: circle;
       } */

      ul {
        list-style-image: url('https://www.w3schools.com/cssref/sqpurple.gif');
      }

      /* ul {
         list-style-position: outside;
       } */
    </style>
  </head>
  <body>
    <h2>Fast Track Course</h2>
    <ul>
      <li>Frontend ReactJS</li>
      <li>Backend Java</li>
      <li>Backend PHP</li>
    </ul>
  </body>
</html>
```

CSS Display



☐ Một số khai báo CSS về cách hiển thị (list) thường dùng:

- ◆ **list-style-position:** vị trí đánh dấu danh sách. (inspect browser, xem tab style).

The image displays a comparison between a code editor and a web browser. On the left, a code editor window titled 'ListStyleProperty.html' shows the following HTML and CSS code:

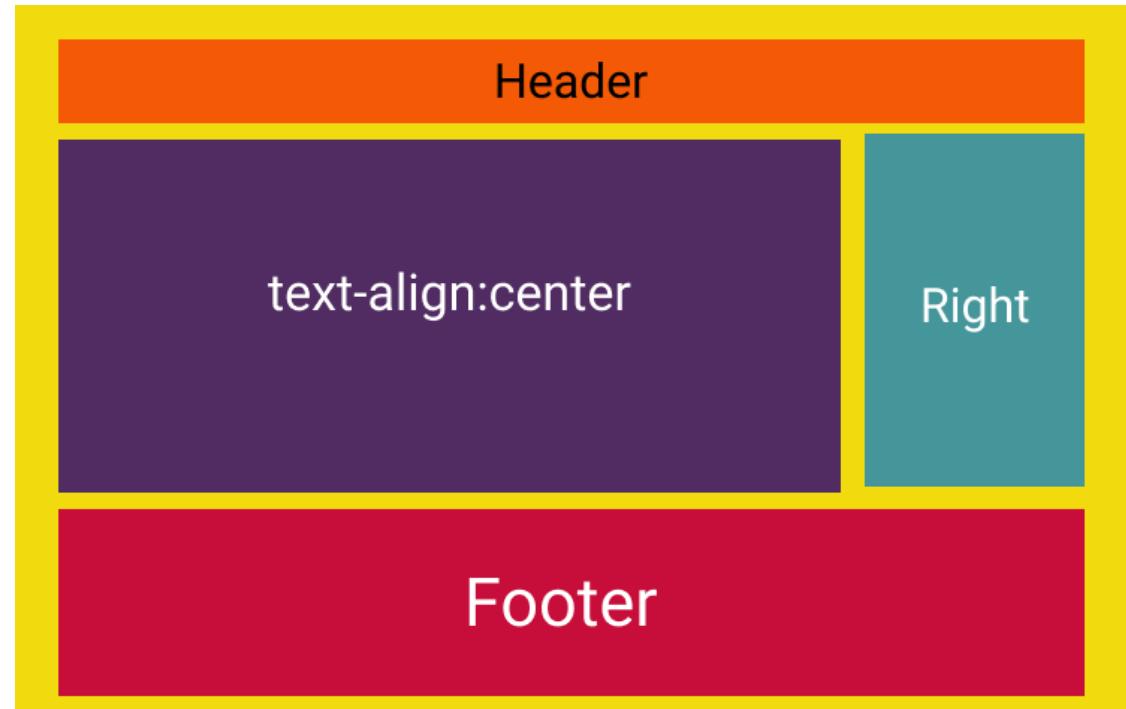
```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>List Style Property</title>
5          <style>
6              /* ul {
7                  list-style: circle;
8              } */
9
10             /* ul {
11                 list-style-image: url('https://');
12             } */
13
14             ul {
15                 list-style-position: outside;
16             }
17         </style>
18     </head>
19     <body>
20         <h2>Fast Track Course</h2>
21         <ul>
22             <li>Frontend ReactJS</li>
23             <li>Backend Java</li>
24             <li>Backend PHP</li>
25         </ul>
26     </body>
27 </html>
```

On the right, a browser window titled 'List Style Property' shows the rendered HTML. The page has a title 'Fast Track Course' and a list of three items: 'Frontend ReactJS', 'Backend Java', and 'Backend PHP'. The list items are styled with circles.

CSS Alignment

☐ Một số khai báo CSS về text, aligning text thường dùng:

- ◆ **text-align:** chỉnh vị trí của đoạn chữ theo chiều ngang.
 - Ví dụ: **text-align: center**
 - ◆ **text-indent:** chỉnh thụt đầu dòng cho đoạn chữ.
 - Ví dụ: **text-indent: 15px;**
 - ◆ **letter-spacing:** khoảng cách giữa các ký tự trong đoạn chữ.
 - Ví dụ: **letter-spacing: 2px;**



CSS Alignment



☐ Một số khai báo CSS về text, aligning text thường dùng:

- ◆ Ví dụ: **letter-spacing: 5px; text-indent: 40px; text-align: justify**

```
<style>
    .text {
        font-size: 15px;
        font-weight: bold;
        line-height: 25px;
        letter-spacing: 2px;
        text-indent: 40px;
        text-align: justify;
    }
</style>
```

<!-- Giới thiệu --></p>

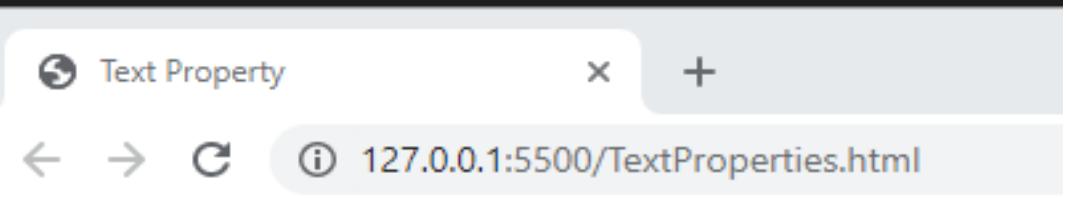
<div class="text">

Định nghĩa một chân dung

Định nghĩa một chân dung

Định nghĩa một chân dung

</div>



Định nghĩa một chân dung lập trình viên là nhu cầu của Doanh nghiệp và thị trường CNTT cột Kỹ năng lập trình - Tiếng Anh giao tiếp - nhiều khóa học khác nhau phù hợp với lộ trình

CSS Alignment



❑ Các cách dùng để căn chỉnh nội dung trang web:

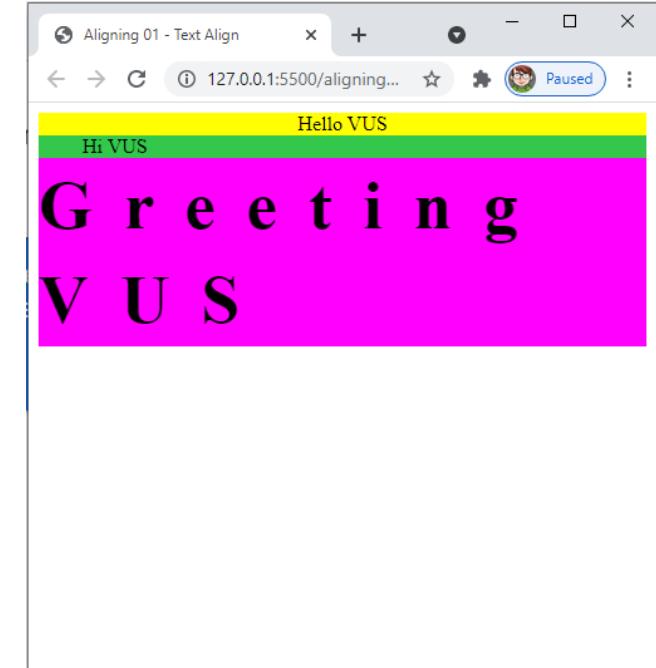
- ◆ Một cách để **căn chỉnh nội dung trang web** là sử dụng thuộc tính **text-align**.
- ◆ Sử dụng thuộc tính text-align để đặt căn lề ngang cho các dòng văn bản trong một phần tử.
- ◆ **text-align** sử dụng một trong 4 giá trị sau: **left** (mặc định), **right**, **center**, **justify**.

```
27  <body>
28      <div class="text">
29          Hello VUS
30      </div>
31
32      <div class="text-2">
33          Hi VUS
34      </div>
35
36      <div class="font">
37          Greeting VUS
38      </div>
39  </body>
```

```
<style>
    .text {
        background-color: yellow;
        text-align: center;
    }

    .text-2 {
        background-color: #rgb(51, 199, 75);
        text-indent: 35px;
    }

    .font {
        background-color: fuchsia;
        font-size: 55px;
        font-weight: bold;
        line-height: 75px;
        letter-spacing: 25px;
    }
</style>
```



CSS Alignment



☐ Một số khai báo CSS về canh chỉnh (Alignment) các phần tử thường dùng:

- ◆ float: thuộc tính float dùng để đẩy các phần tử sang bên trái hoặc phải.
- ◆ float: left để đẩy phần tử về phía bên phải và ngược lại.

```
.CssAlignment.html X
CssAlignment.html > html
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <style>
6          img {
7              float: left;
8          }
9      </style>
10     </head>
11
12     <body>
13         <h1>Thuộc tính float</h1>
14
15         <p>Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.</p>
16
17         <p>
18             Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.
19             Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.
20             Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.
21             Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.
22             Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.
23             Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.
24             Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.</p>
25     </body>
26
27     </html>
```

CSS Alignment



☐ Một số khai báo CSS về canh chỉnh (Alignment) các phần tử thường dùng:

- ◆ **float:** thuộc tính float dùng để đẩy các phần tử sang bên trái hoặc phải.
- ◆ **float: left** để đẩy phần tử về phía bên phải và ngược lại.

The screenshot shows a web browser window with the URL `127.0.0.1:5501/CssAlignment.htm`. The title of the page is "Thuộc tính float". The content of the page includes a paragraph of text and a photograph of a pineapple. The text discusses the effect of the float property on the layout. The photograph of the pineapple is positioned to the left of the text, demonstrating how floating elements affect the surrounding content.

Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.

Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.

Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình. Trong ví dụ này, chiếc hình trái dứa sẽ nằm bên phải chữ, và chữ trong đoạn văn bản sẽ bao bọc xung quanh bức hình.



CSS Alignment

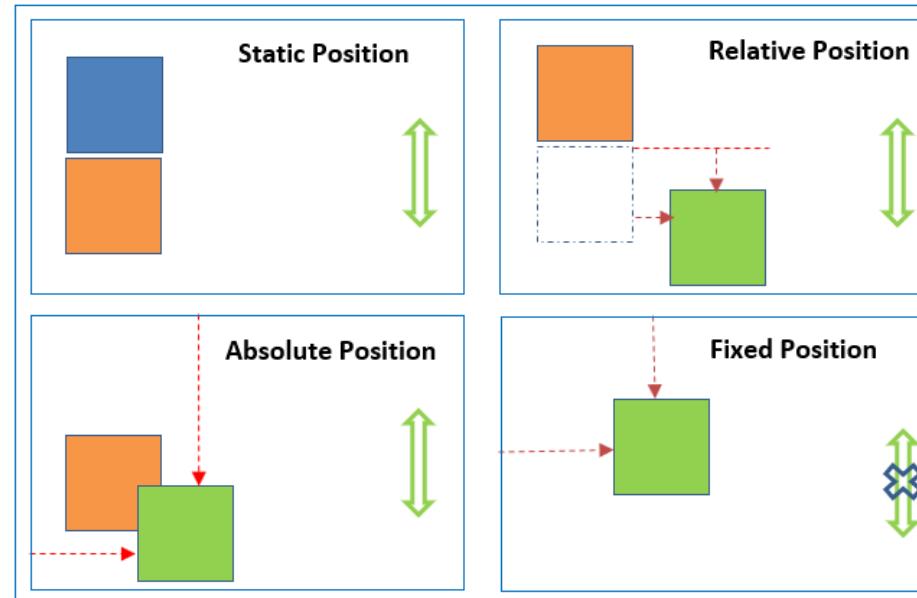
Một số khai báo CSS về canh chỉnh các phần tử thường dùng:

- ◆ **float** sẽ đẩy các phần tử block nằm trên cùng một hàng từ trái sang phải hoặc ngược lại.
 - ◆ Nếu **một hàng không đủ chỗ chứa**, các phần tử sẽ **trôi xuống hàng tiếp theo**.
 - ◆ Nếu **các phần tử có chiều cao khác nhau**, hàng tiếp theo sẽ **bắt đầu bên dưới phần tử cho chiều cao thấp nhất sau cùng**.
 - ◆ Khi sử dụng **float**, các phần tử phía sau sẽ **bị tác động rất dễ gây lỗi giao diện**. Vì vậy, ta **nên sử dụng** thuộc tính **clear:both**; ngay dưới thẻ cuối cùng của **float** để loại bỏ lỗi này.

CSS Alignment

Một số khai báo CSS về canh chỉnh các phần tử thường dùng:

- ♦ Thuộc tính **position** dùng để canh chỉnh vị trí các phần tử, bao gồm các giá trị sau:
 - **relative**: di chuyển phần tử theo vị trí tương đối hiện tại.
 - **absolute**: di chuyển phần tử theo phần tử cha chứa nó.
 - **fixed**: cố định phần tử trên trang Web.
- ♦ Các thuộc tính đi kèm **position** để căn chỉnh vị trí phần tử là: **top, left, right, bottom**

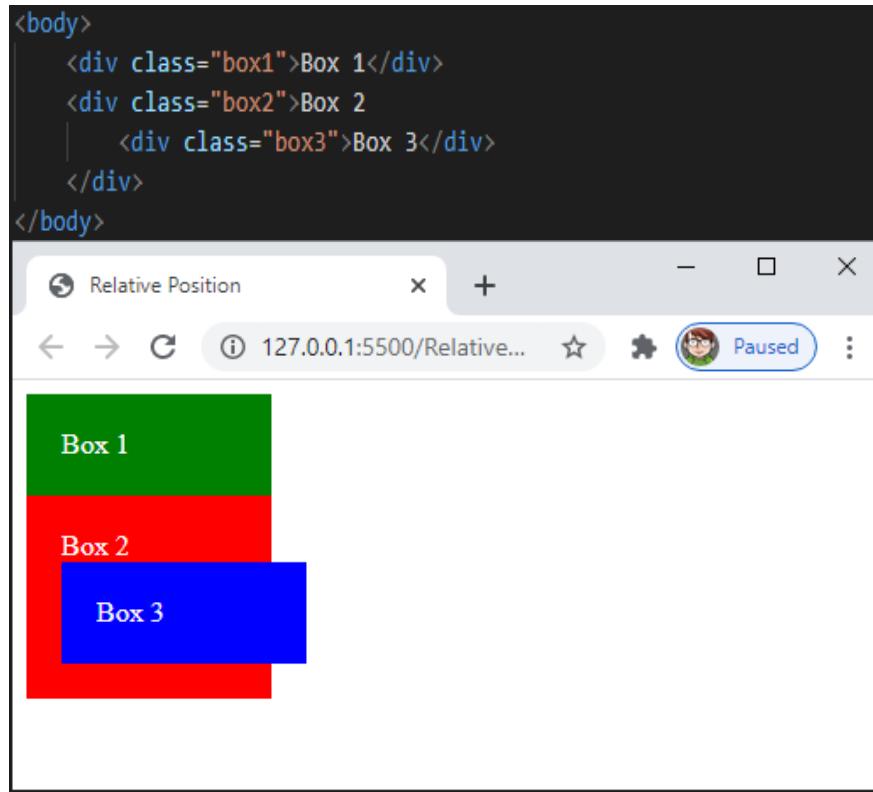


CSS Alignment

☐ Một số khai báo CSS về canh chỉnh các phần tử thường dùng:

- ♦ Thuộc tính **position** dùng để canh chỉnh vị trí các phần tử, bao gồm các giá trị sau:
 - **relative**: di chuyển phần tử theo vị trí tương đối hiện tại.

```
<style>
    .box1 {
        position: relative;
        color: white;
        background-color: green;
        padding: 20px;
        width: 100px;
    }
    .box2 {
        position: relative;
        top: 0;
        right: 0;
        color: white;
        background-color: red;
        padding: 20px;
        width: 100px;
    }
    .box3 {
        position: relative;
        color: white;
        background-color: blue;
        padding: 20px;
        width: 100px;
    }
</style>
```





CSS Alignment

☐ Một số khai báo CSS về canh chỉnh các phần tử thường dùng:

- ♦ Thuộc tính **position** dùng để canh chỉnh vị trí các phần tử, bao gồm các giá trị sau:
 - **absolute**: di chuyển phần tử theo phần tử cha chứa nó.

The screenshot shows a browser window with developer tools open. The address bar says "127.0.0.1:5500/Absolut...". The page content consists of a single parent div containing three child divs: "Box 1", "Box 2", and "Box 3". Each child div has a different background color (green, red, blue) and white text. The developer tools panel shows the CSS styles for these elements. The "Box 1" style defines a green background, white text, 20px padding, and a 100px width. The "Box 2" style defines a red background, white text, 20px padding, and a 100px width, with absolute positioning relative to its parent. The "Box 3" style defines a blue background, white text, 20px padding, and a 100px width, also with absolute positioning relative to its parent.

```
<style>
    .box1 {
        position: absolute;
        color: white;
        background-color: green;
        padding: 20px;
        width: 100px;
    }
    .box2 {
        position: absolute;
        top: 0;
        right: 0;
        color: white;
        background-color: red;
        padding: 20px;
        width: 100px;
    }
    .box3 {
        position: absolute;
        color: white;
        background-color: blue;
        padding: 20px;
        width: 100px;
    }
</style>
```

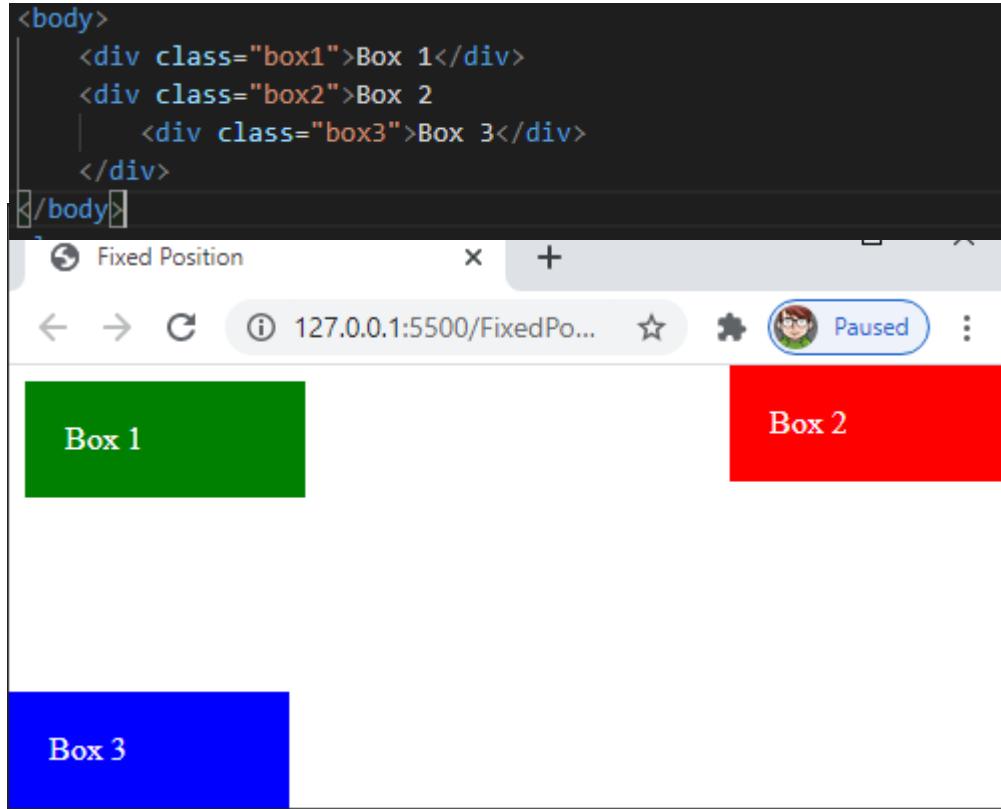
```
<body>
    <div class="box1">Box 1</div>
    <div class="box2">Box 2
        <div class="box3">Box 3</div>
    </div>
</body>
```

CSS Alignment

☐ Một số khai báo CSS về canh chỉnh các phần tử thường dùng:

- ♦ Thuộc tính **position** dùng để canh chỉnh vị trí các phần tử, bao gồm các giá trị sau:
 - **fixed**: cố định phần tử trên trang Web.

```
<style>
    .box1 {
        position: fixed;
        color: white;
        background-color: green;
        padding: 20px;
        width: 100px;
    }
    .box2 {
        position: fixed;
        top: 0;
        right: 0;
        color: white;
        background-color: red;
        padding: 20px;
        width: 100px;
    }
    .box3 {
        position: fixed;
        bottom: 0;
        left: 0;
        color: white;
        background-color: blue;
        padding: 20px;
        width: 100px;
    }
</style>
```



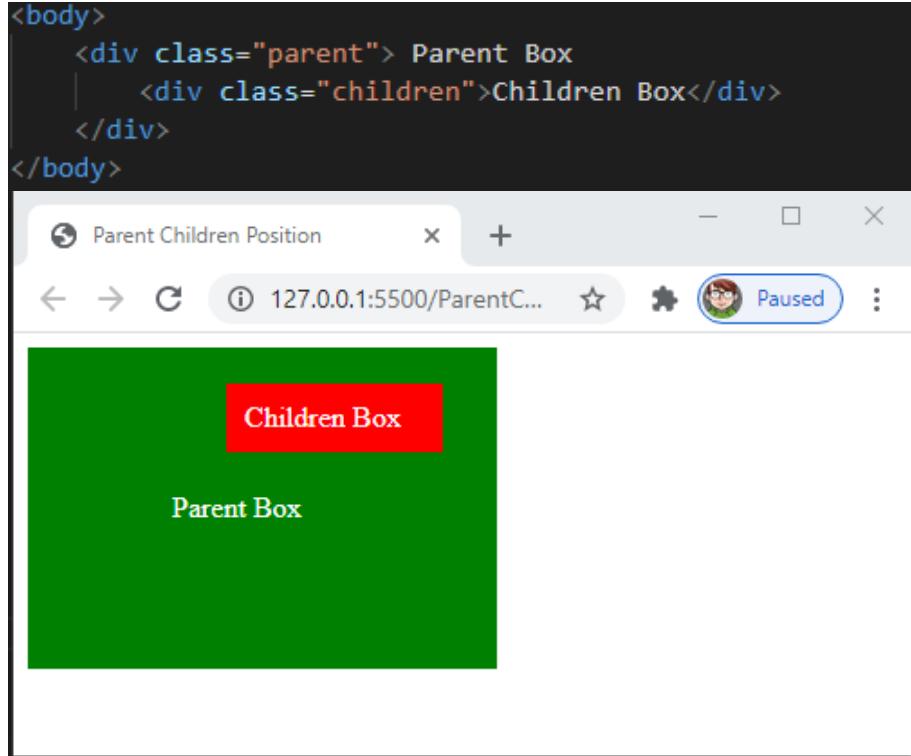
CSS Alignment



☐ Một số khai báo CSS về canh chỉnh các phần tử thường dùng:

- ◆ Position theo quan hệ cha con: **Ở thẻ cha**, ta thường dùng **relative** cho thuộc tính **position** để thẻ cha không di chuyển. **Ở thẻ con**, ta thường dùng **absolute** cho thuộc tính **position** để thẻ con có thể di chuyển bên trong thẻ cha.

```
<style>
    .parent {
        position: relative;
        color: white;
        background-color: green;
        padding: 80px;
        width: 100px;
    }
    .children {
        position: absolute;
        top: 20px;
        right: 30px;
        color: white;
        background-color: red;
        padding: 10px;
        width: 100px;
    }
</style>
```



CSS Border Radius



☐ Khai báo CSS bo tròn các góc (CSS Rounded Corners):

- ◆ Với thuộc tính **border-radius**, bạn có thể bo tròn các góc của bất kỳ phần tử nào.

```
<style>
    #rcorners1 {
        border-radius: 25px;
        background: #73AD21;
        padding: 20px;
        width: 200px;
        height: 150px;
    }

    #rcorners2 { ... }

    #rcorners3 { ... }
</style>
```

```
<h1>Các thuộc tính border-radius</h1>

<p>Các góc bo tròn của một phần tử với màu nền xác định:</p>
<p id="rcorners1">Các góc bo tròn loại 1!</p>
```

Các thuộc tính border-radius

Các góc bo tròn của một phần tử với màu nền xác định:

Các góc bo tròn loại 1!

CSS Border Radius



□ Khai báo CSS bo tròn các góc (CSS Rounded Corners):

- ◆ Với thuộc tính **border-radius**, bạn có thể bo tròn các góc của bất kỳ phần tử nào.

```
<style>
    #rcorners1 { ...
}

    #rcorners2 {
        border-radius: 25px;
        border: 2px solid #73AD21;
        padding: 20px;
        width: 200px;
        height: 150px;
    }

    #rcorners3 { ...
}
</style>
```

```
<h1>Các thuộc tính border-radius</h1>

<p>Các góc bo tròn của một phần tử với màu nền xác định:</p>
<p id="rcorners1">Các góc bo tròn loại 1!</p>
<p>Các góc bo tròn của một phần tử với một đường bao quanh:</p>
<p id="rcorners2">Các góc bo tròn loại 2!</p>
```

Các góc bo tròn của một phần tử với một đường bao quanh:

Các góc bo tròn loại 2!



CSS Border Radius

Khai báo CSS bo tròn các góc (CSS Rounded Corners):

- ◆ Với thuộc tính **border-radius**, bạn có thể bo tròn các góc của bất kỳ phần tử nào.

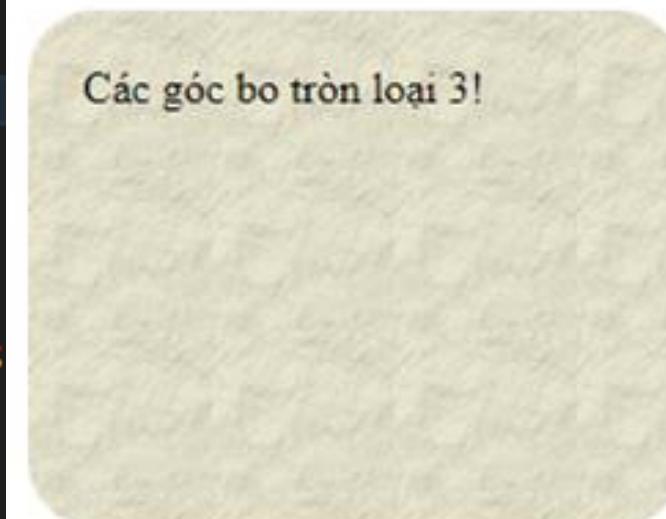
```
<style>
    #rcorners1 { ...
}

    #rcorners2 { ...
}

    #rcorners3 {
        border-radius: 25px;
        background: url(https://www.w3schools.com/css/paper.gif);
        background-position: left top;
        background-repeat: repeat;
        padding: 20px;
        width: 200px;
        height: 150px;
    }
</style>
```

Các góc bo tròn của một phần tử với một hình nền:

Các góc bo tròn loại 3!



CSS Border Radius



Khai báo CSS bo tròn các góc (CSS Rounded Corners):

- ◆ Thuộc tính **border-radius** có thể có từ 1 đến 4 giá trị, và sau đây là một số quy luật:
 - 4 giá trị - **border-radius**: **15px 50px 30px 5px**; thì ý nghĩa từng giá trị như sau:
 - Giá trị đầu tiên (**15px**): cạnh góc trái, phía bên trên (**top-left**)
 - Giá trị thứ hai (**50px**): cạnh góc phải, phía bên trên (**top-right**)
 - Giá trị thứ ba (**30px**): cạnh góc phải, phía bên dưới (**bottom-right**)
 - Giá trị đầu tiên (**5px**): cạnh góc trái, phía bên dưới (**bottom-left**)

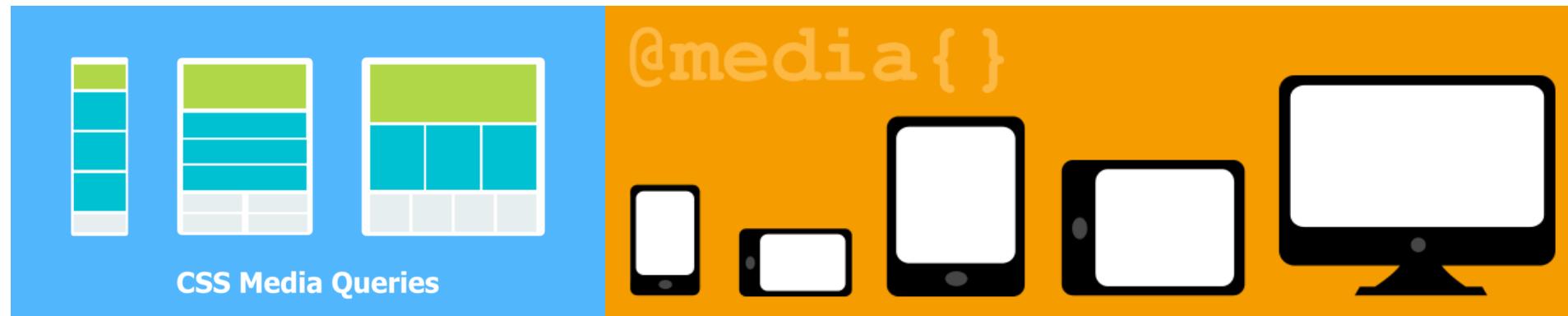


Media Queries



□ Giới thiệu:

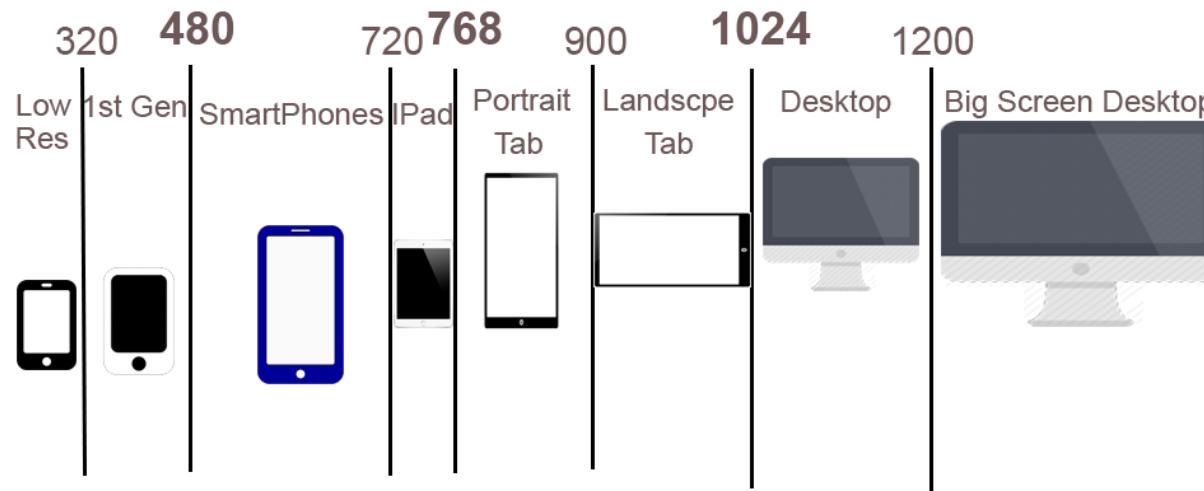
- ◆ Mobile-First **tập trung** vào việc **tạo kiểu**, **nội dung phù hợp cho** các thiết bị **mobile**.
- ◆ Mobile-First **sử dụng bố cục một cột** để **ngăn cuộn ngang** và **giảm thiểu cuộn dọc**.
-> **Bố cục này tạo ra trải nghiệm thú vị** hơn **cho người dùng** thiết bị **mobile**.
- ◆ Thiết kế cho màn hình **tablet** và **desktop** cho phép **bố cục trang web linh hoạt hơn**.
 - Thiết kế cho **tablet** có thể **sử dụng bố cục hai cột**.
 - Thiết kế cho **desktop** có thể **sử dụng bố cục nhiều cột**.
- ◆ Thiết kế cho màn hình **tablet** và **desktop** được **đơn giản hóa** thông qua việc **sử dụng** các truy vấn phương tiện **Media Queries**.



Media Queries

□ Giới thiệu:

- ◆ Quy tắc **@media** có thể áp dụng các định kiểu **CSS** để giúp:
 - Di chuyển, ẩn hoặc hiển thị nội dung trên trang web.
 - Thay đổi văn bản hoặc màu sắc hoặc thêm bất kỳ định kiểu CSS nào khác để giúp trang dễ đọc hơn trong một tình huống cụ thể.
- ◆ Nếu bạn thay đổi kích thước trang web và thấy hệ thống điều hướng thay đổi hoặc thấy bố cục nhiều cột giảm xuống một cột khi viewport thu hẹp lại bằng kích thước của điện thoại, đó là do các **media queries** đang hoạt động định kiểu lại **viewport** phù hợp.





Media Queries

Giới thiệu:

- ◆ Quy tắc **@media** có thể:
 - Được nhúng trong thẻ **link** kết nối định kiểu **CSS** bên ngoài với tệp HTML.
 - Ví dụ 1: <link rel="stylesheet" href="css/styles.css" media="screen">
 - Ý nghĩa: **styles.css** được tìm thấy trong folder CSS sẽ được áp dụng nếu trang web được hiển thị trên màn hình (screen).
 - Ví dụ 2: <link rel="stylesheet" href="css/stylesprint.css" media="print">
 - Ý nghĩa: **stylesprint.css** được tìm thấy trong folder CSS sẽ được áp dụng nếu trang web được in (print).
 - Hoặc có thể được **chèn vào** định kiểu **CSS** bên ngoài (tức file CSS ở ngoài).
 - ◆ Đây là **một kỹ thuật phổ biến** cho phép tạo kiểu CSS với màu sắc phong phú, hình ảnh nhiều màu sắc trên màn hình, nhưng text vẫn màu đen trên khi trang web được in.



Media Queries

Điểm ngắt (breakpoints):

- ♦ Media queries có thể **xác định loại thiết bị hiện tại thông qua viewport size bằng cách thiết lập một điểm ngắt (breakpoints) cho trang web.**
- ♦ **Breakpoints** là **điểm chuyển giao giữa 2 định kiểu màn hình khác nhau** giúp trang web dễ đọc hơn trên từng loại màn hình khác nhau:

| Thiết bị | Độ rộng viewport nhỏ nhất | Độ rộng viewport lớn nhất |
|---|---------------------------|---------------------------|
| Smartphones cỡ nhỏ | 320px | 480px |
| Tablets, smartphones cỡ lớn | 481px | 768px |
| Tablets (hướng ngang), laptops, màn hình desktop cỡ nhỏ | 769px | 1279px |
| Màn hình desktop cỡ lớn | 1280px | NA |



Media Queries

□ Cú pháp:

◆ Cú pháp của Media Query:

```
@media not|only mediatype and (expressions) {
```

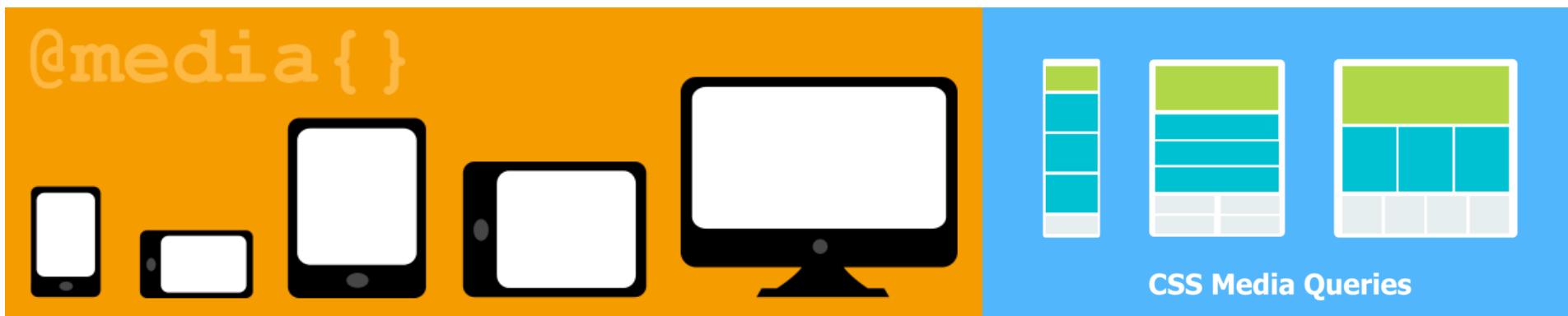
CSS-Code;

}

◆ Nếu thông số màn hình đang được **chỉ định khớp** với các biểu thức **media queries**

-> **media queries** trả về giá trị **true** -> CSS nằm trong **media queries** này sẽ **được áp dụng**.

◆ Trừ khi bạn **sử dụng** toán tử **not** hoặc **only**, **mediatype** là **tùy bạn chọn**. Còn không, **all** sẽ là mặc định và sẽ áp dụng **media queries** chọn tất cả các loại thiết bị khác nhau.





Media Queries

□ Cú pháp:

◆ Cách 1: Biểu thức truy vấn phương tiện (Media Query Expressions) **có thể được nhúng trực tiếp vào thẻ link** khi thêm đường dẫn file CSS bên ngoài như sau:

- Ví dụ 1: `<link rel="stylesheet" href="css/styles-mobile.css" media="screen and (max-width: 480px)">`

→ Mã này **hướng** các trình duyệt sử dụng biểu định kiểu **styles-mobile.css** trong thư mục CSS khi **màn hình có chiều rộng khung nhìn <= 480px**.

- Ví dụ 2: `<link rel="stylesheet" href="css/styles-tablet.css" media="screen and (min-width: 481px) and (max-width: 768px)">`

→ Mã này **hướng** các trình duyệt sử dụng biểu định kiểu **styles-tablet.css** trong thư mục CSS khi **màn hình có chiều rộng khung nhìn >= 481px và <= 768px**.

Media Queries



□ Cú pháp:

◆ Cách 2: Biểu thức truy vấn phương tiện (Media Query Expressions) **có thể được chèn thông qua** file CSS bên ngoài như sau:

▪ VD1: @media screen {
 body { width: 80%; }
}

▪ VD3: @media screen and (min-width: 481px) {
 .mobile { display: none; }
}

▪ VD2: @media print {
 body {
 width: 100%;
 color: #000000;
 background-color: #FFFFFF;
 }
}

▪ VD4: @media all and (min-width: 481px)
 and (max-width: 768px) {
 body { color: #000000; }
 }

Tóm tắt bài học



- Khái niệm Box Model
- Các CSS Padding
- Các CSS Margin
- Các CSS Display
- Các CSS Visibility
- Các CSS Alignment
- Các CSS Border Radius
- Các Media Queries