

Bài 05: Định kiểu Website với các CSS phổ biến



Nội dung

1. Định kiểu phần tử đoạn văn bản với CSS Fonts
2. Thiết lập màu sắc phần tử HTML với CSS Colors
3. Mô hình hộp khối CSS Box Model, Padding, Margin
4. Cách hiển thị phần tử HTML với CSS Display, Visibility
5. Canh chỉnh vị trí phần tử HTML với CSS Alignment
6. Đường bao phần tử HTML với Border
7. Sắp xếp hiển thị nhiều hình nền với Background



Định kiểu phần tử đoạn văn bản với CSS Fonts

❑ Giới thiệu:

- ♦ Bạn có thể dùng **định kiểu CSS để định dạng văn bản trên** một **webpage** bằng nhiều cách khác nhau.
- ♦ Danh sách **những thuộc tính** và **giá trị** của **các định kiểu CSS phổ biến**:

Thuộc tính	Mô tả	Các giá trị phổ biến
font-family	Tên phông chữ cụ thể và chung chung	font-family: Cambria, “Times New Roman”, serif; font-family: Verdana, Arial, sans-serif; font-family: Georgia, “Times New Roman”, serif;
font-size	Kích thước tuyệt đối hoặc tương đối của phông chữ	font-size: 1.5em; font-size: 50%; font-size: x-large; font-size: 14pt;
font-weight	Trọng lượng của một phông chữ	font-weight: bold; font-weight: bolder; font-weight: lighter;



Định kiểu phần tử đoạn văn bản với CSS Fonts

❑ Giới thiệu:

- ◆ Bạn có thể dùng **định kiểu CSS để định dạng văn bản trên** một **webpage** bằng nhiều cách khác nhau.
- ◆ Danh sách **những thuộc tính** và **giá trị** của **các định kiểu CSS phổ biến**:

Thuộc tính	Mô tả	Các giá trị phổ biến
font-style	Phong cách của một phong chữ	font-style: normal; font-style: italic; font-style: oblique;
line-height	Sự canh chỉnh chiều cao của chữ	line-height: 25px;
color	Màu của chữ	color: red; color: green; color: blue;



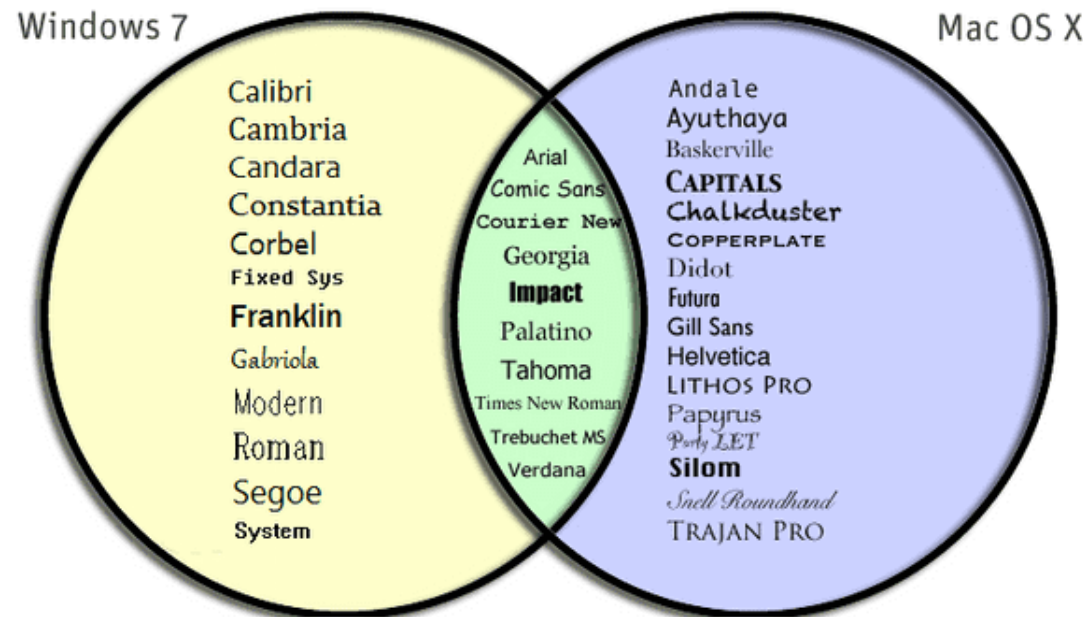
Định kiểu phần tử đoạn văn bản với CSS Fonts

❑ Tại sao một trình duyệt không hỗ trợ những phong chữ xác định?

♦ Các phong chữ được cài trên một máy tính trước khi một trình duyệt có thể hiển thị. Chúng ta **chỉ có một danh sách các phong chữ chung dành cho web**.

(Tham khảo: https://www.w3schools.com/cssref/css_websafe_fonts.asp)

♦ Vì vậy, nếu **phong chữ** bạn dùng để **thiết kế website** là một loại đặc biệt **chưa được cài trên máy tính** bạn **lúc đó**, thì trình duyệt sẽ **không đọc được**.





Định kiểu phần tử đoạn văn bản với CSS Fonts

❑ Lưu ý:

- ♦ Thuộc tính **font-family** có thể **bao gồm nhiều giá trị khác nhau**.
- ♦ Bạn **nên cung cấp nhiều hơn một giá trị cho thuộc tính này** để phòng trường hợp trình duyệt không hỗ trợ phông chữ chính, các giá trị thêm vào sẽ được gọi là **giá trị dự phòng** (fallback values).
- ♦ Trường hợp giá trị chính không được hỗ trợ, các phông chữ dự phòng sẽ được dùng.
- ♦ Ví dụ: **font-family: Cambria, "Times New Roman", serif;**

```
h1 {  
  font-family: "Times New Roman", Palatino, Georgia, Serif;  
}
```

Fallback fonts



Định kiểu phần tử đoạn văn bản với CSS Fonts

❑ Đơn vị đo kích cỡ phông chữ:

♦ CSS đo kích thước phông chữ bằng cách sử dụng **nhiều đơn vị đo lường**, bao gồm **pixel**, **points**, **ems** và **keyword** hoặc **%**. Danh sách các đơn vị đo cỡ chữ phổ biến:

Đơn vị	Định nghĩa	Ví dụ	Ghi chú
em	Liên quan đến kích thước phông chữ mặc định của phần tử	font-size: 1.5em;	Được đề xuất bởi W3C; kích thước có liên quan đến kích thước phông chữ mặc định của trình duyệt
%	Liên quan đến kích thước phông chữ mặc định của phần tử	font-size: 50%;	Được đề xuất bởi W3C; kích thước có liên quan đến kích thước phông chữ mặc định của trình duyệt
px	Số lượng pixel	font-size: 25px;	Phụ thuộc vào độ phân giải màn hình
pt	Số lượng điểm	font-size: 12pt;	Sử dụng để in các trang web
Từ khóa	Liên quan đến một số kích thước hạn chế	font-size: x-small;	Kích thước có liên quan đến kích thước phông chữ mặc định của trình duyệt, nhưng các tùy chọn kích thước bị hạn chế



Định kiểu phần tử đoạn văn bản với CSS Fonts

❑ Một số khai báo CSS về font và text thường dùng :

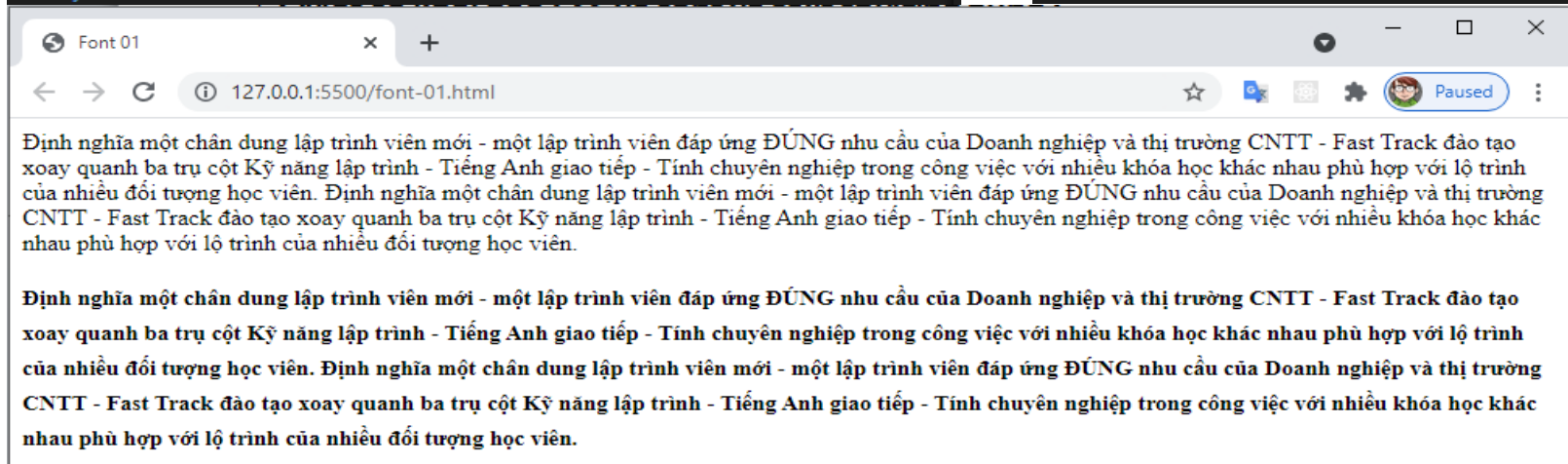
◆ Ví dụ **font-size**, **font-weight**, **line-height**:

```
<body>
  <div>
    Định nghĩa một chân dung lập trình viên mới - một lập trình viên đáp ứng
    khóa học khác nhau phù hợp với lộ trình của nhiều đối tượng học viên. Đ
    lập trình - Tiếng Anh giao tiếp - Tính chuyên nghiệp trong công việc vớ
  </div>

  <br>

  <div class="font">
    Định nghĩa một chân dung lập trình viên mới - một lập trình viên đáp ứng
    khóa học khác nhau phù hợp với lộ trình của nhiều đối tượng học viên. Đ
    lập trình - Tiếng Anh giao tiếp - Tính chuyên nghiệp trong công việc vớ
  </div>
</body>
```

```
<head>
  <title>Font 01</title>
  <style>
    .font {
      font-size: 15px;
      font-weight: bold;
      line-height: 25px;
    }
  </style>
</head>
```





Thiết lập màu sắc phần tử HTML với CSS Colors

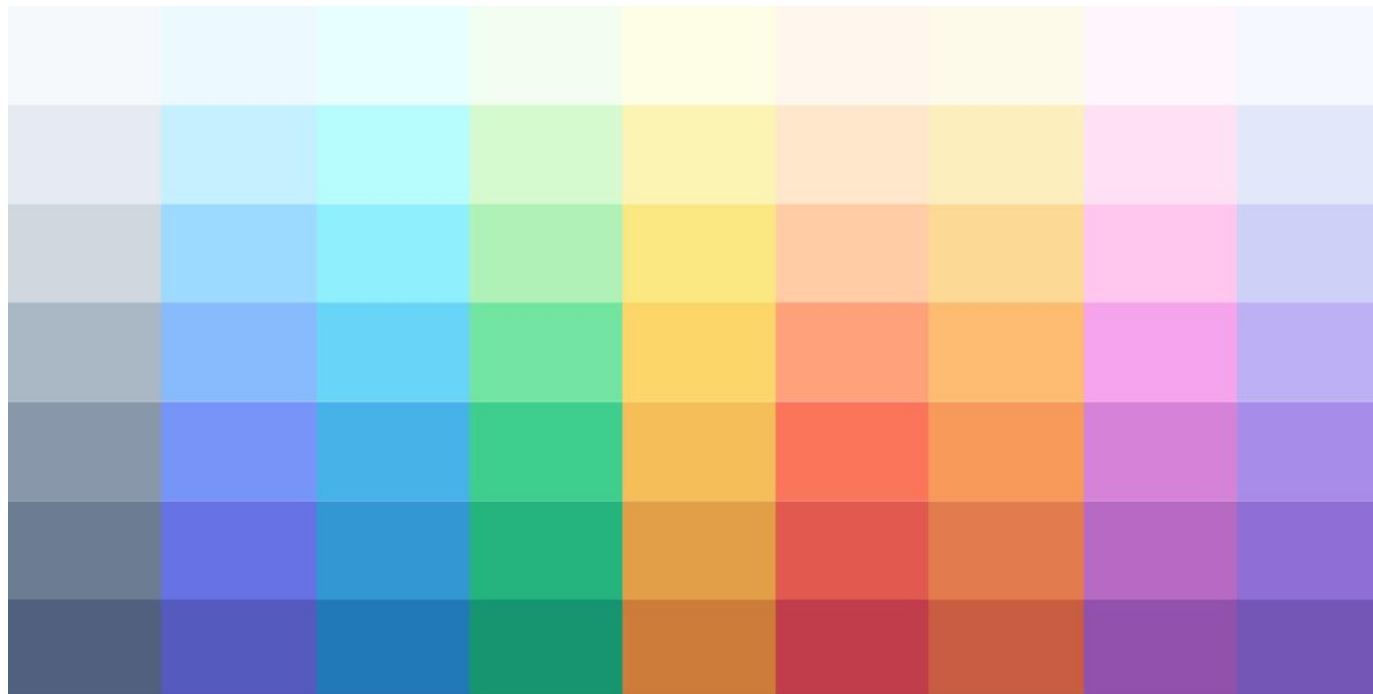
❑ Một số khai báo CSS về màu sắc thường dùng:

◆ **color**: màu chữ

▪ **color name**: màu sắc được xác định bằng **tên màu**. Ví dụ: green, red, purple,...

▪ **HEX value**: màu sắc được xác định bằng **trị số hexa** theo **cấu trúc #rrggbb**.

Trong đó: **rr** (red) **gg** (green) **bb** (blue) ứng với giá trị hexa từ **00** -> **ff**





Thiết lập màu sắc phần tử HTML với CSS Colors

❑ Một số khai báo CSS về màu sắc thường dùng:

◆ Ví dụ:

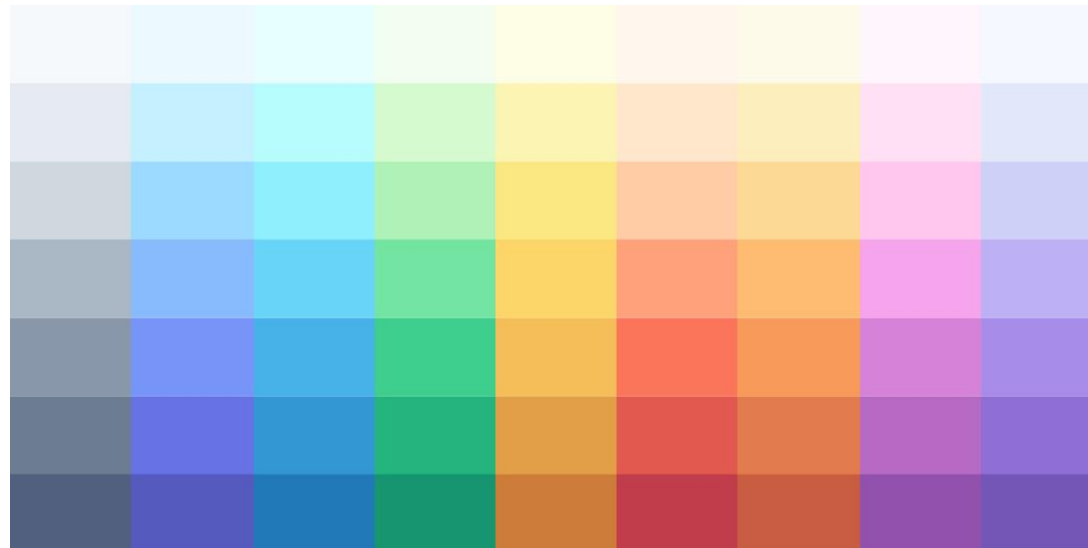
#ff0000 là màu đỏ tươi do phần **red** là cao nhất, các phần còn lại thấp nhất.

#00ff00 là màu xanh lục tươi do phần **green** là cao nhất, các phần còn lại thấp nhất.

#0000ff là màu xanh dương tươi do phần **blue** là cao nhất, các phần còn lại thấp nhất.

▪ **RGB value:** màu sắc được xác định theo **cấu trúc: rgb(red, green, blue)**

Trong đó: **red, green, blue** ứng với giá trị từ **0 -> 255**

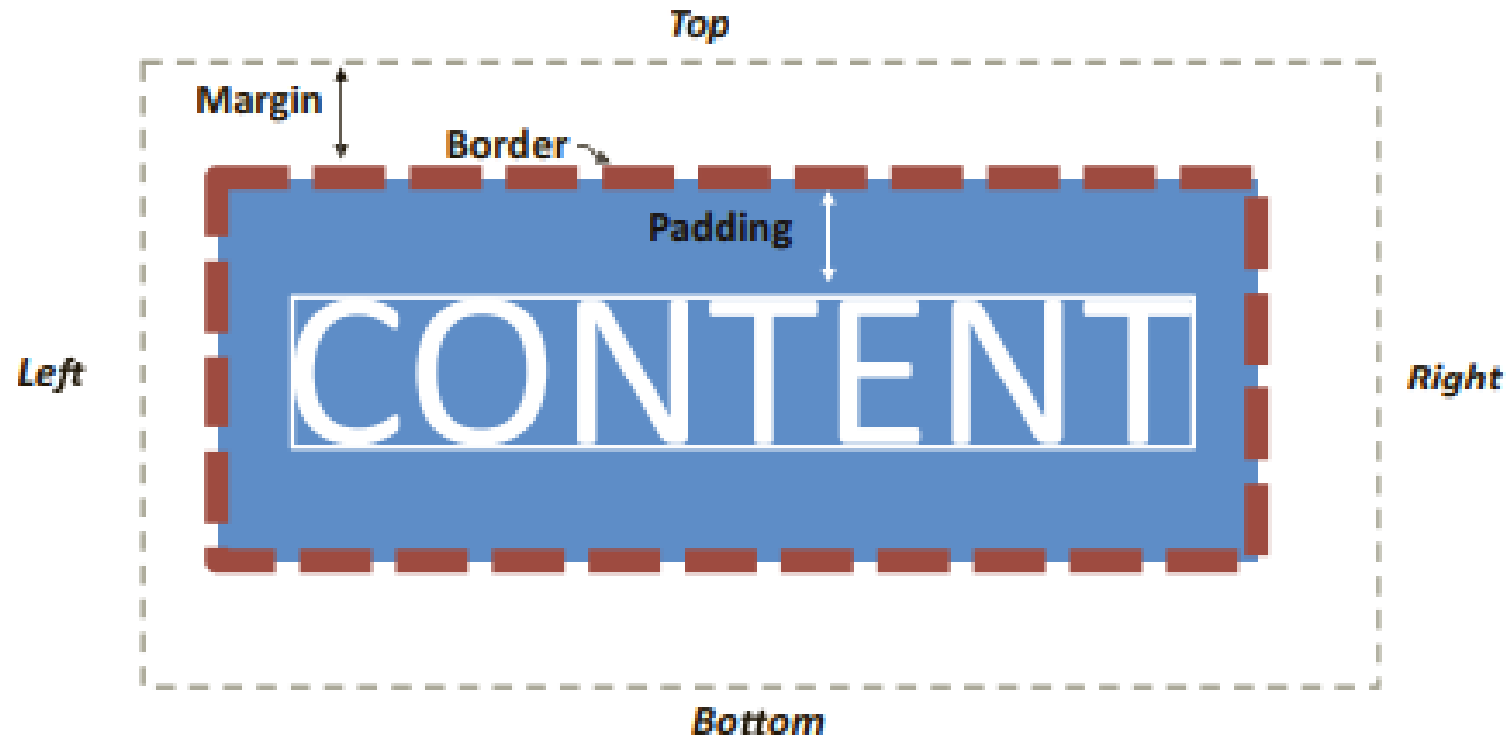




Mô hình hộp khối CSS Box Model, Padding, Margin

❑ 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;**
- ♦ **margin**: là khoảng cách giữa các phần tử HTML với nhau. Ví dụ: **margin: 15px;**

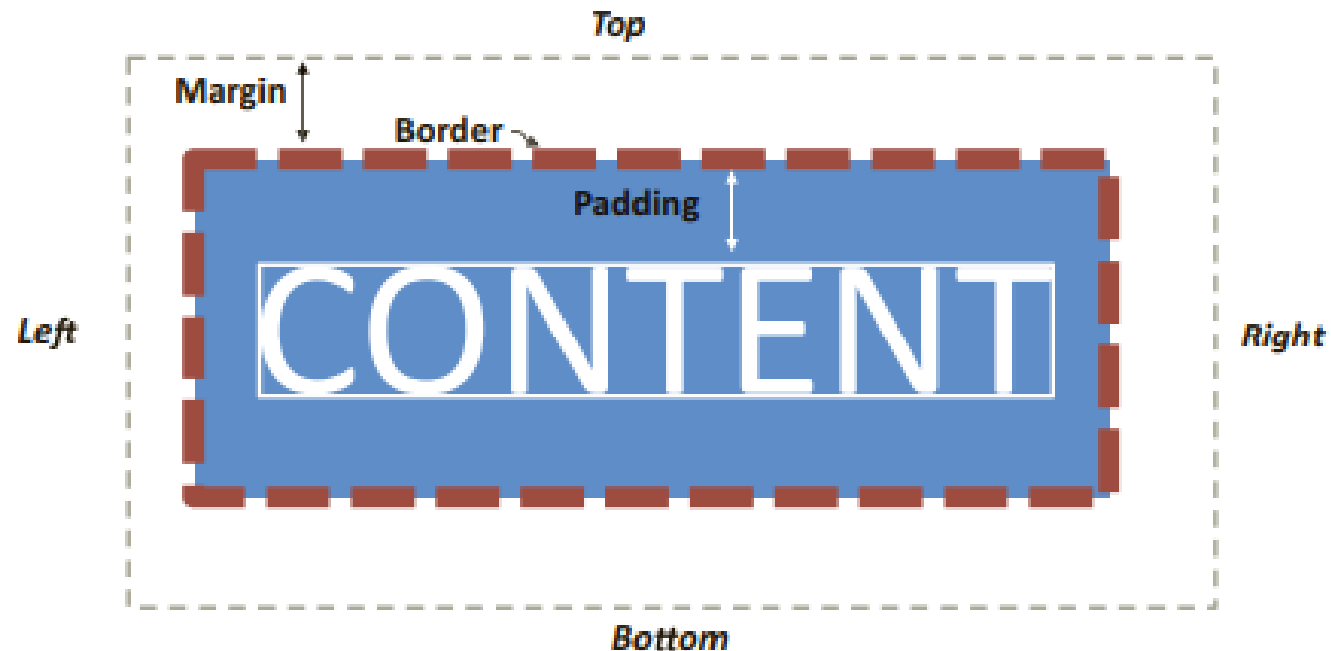




Mô hình hộp khối CSS Box Model, Padding, Margin

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

- ♦ Một số thuộc tính của **padding**:
 - **padding-top** (vùng đệm phía trên)
 - **padding-right** (vùng đệm phía bên phải)
 - **padding-bottom** (vùng đệm phía dưới)
 - **padding-left** (vùng đệm phía bên trái)

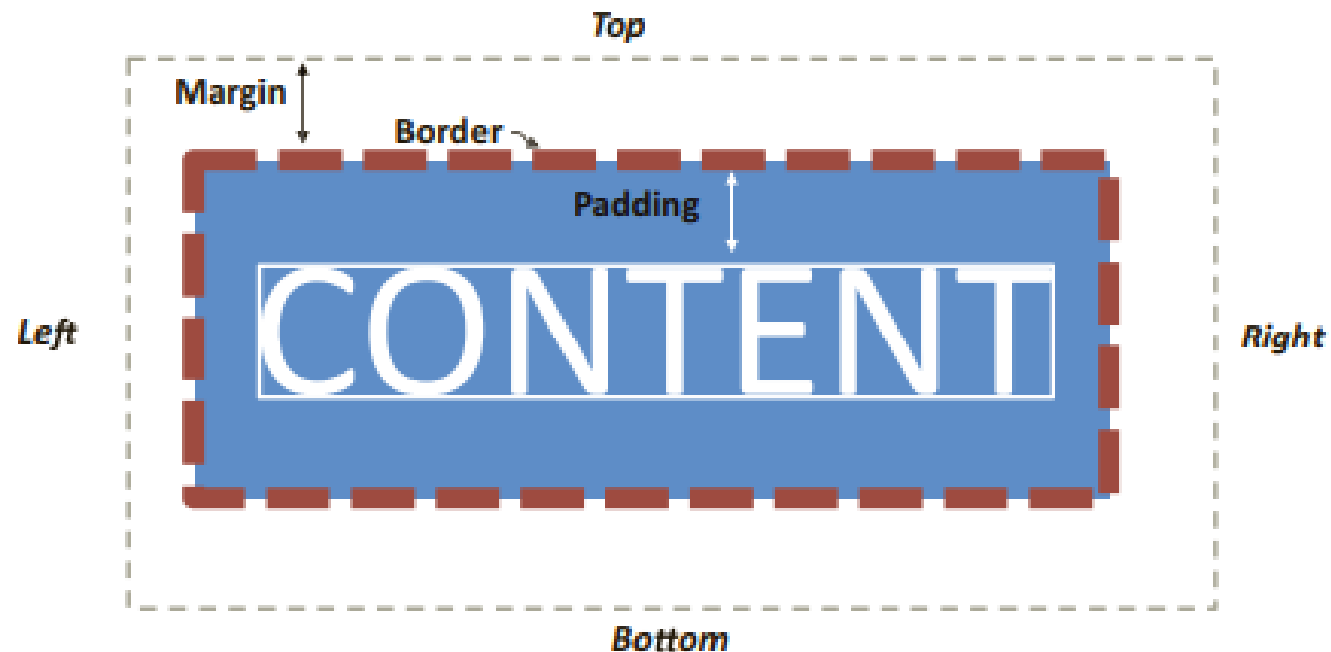




Mô hình hộp khối CSS Box Model, Padding, Margin

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

- ♦ Một số thuộc tính của **margin**:
 - **margin-top** (canh lề phía trên)
 - **margin-right** (canh lề phía bên phải)
 - **margin-bottom** (canh lề phía dưới)
 - **margin-left** (canh lề phía bên trái)





Mô hình hộp khối CSS Box Model, Padding, 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;**

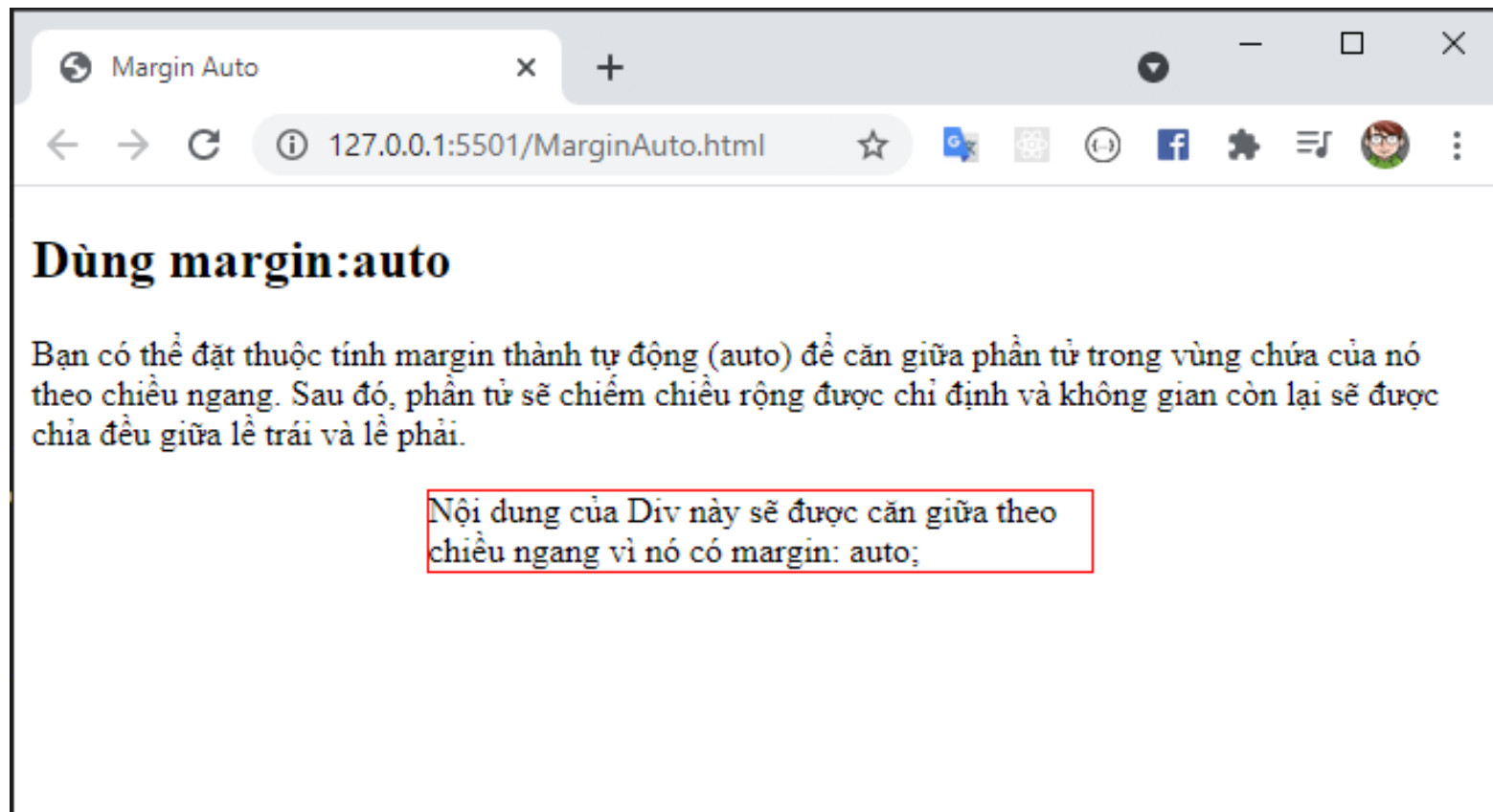
```
MarginAuto.html X
MarginAuto.html > html > head > style
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Margin Auto</title>
5     <style>
6       div {
7         width: 300px;
8         margin: auto;
9         border: 1px solid red;
10      }
11    </style>
12  </head>
13  <body>
14    <h2>Dùng 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>
```



Mô hình hộp khối CSS Box Model, Padding, 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;**





Mô hình hộp khối CSS Box Model, Padding, Margin

❑ 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 with a file named 'PaddingProperty.html'. The code defines four different ways to set the padding property for a `div` element:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Padding Property</title>
5     <style>
6       /* Cách 1 */
7       div {
8         padding-top: 10px;
9         padding-right: 10px;
10        padding-bottom: 10px;
11        padding-left: 10px;
12      }
13      /* Cách 2 */
14      div {
15        padding: 20px; /* Chỉnh padding cho cả 4 phía */
16      }
17      /* Cách 3 */
18      div {
19        padding: 10px 20px; /* 2 cặp giá trị lần lượt là top, bottom và left, right */
20      }
21      /* Cách 4 */
22      div {
23        padding: 5px 10px 15px 20px; /* Chỉnh padding cho cả 4 phía: top, bottom, left, right */
24      }
25    </style>
26  </head>
27  <body>
28    <!-- Giới thiệu -->
29    <div>
30      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.
31    </div>
32  </body>
33 </html>
```

Overlaid on the code editor is a browser window titled 'Padding Property' at the address '127.0.0.1:5500/Padding...'. The browser displays the rendered content of the `div` element, which is a paragraph of text: '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.'



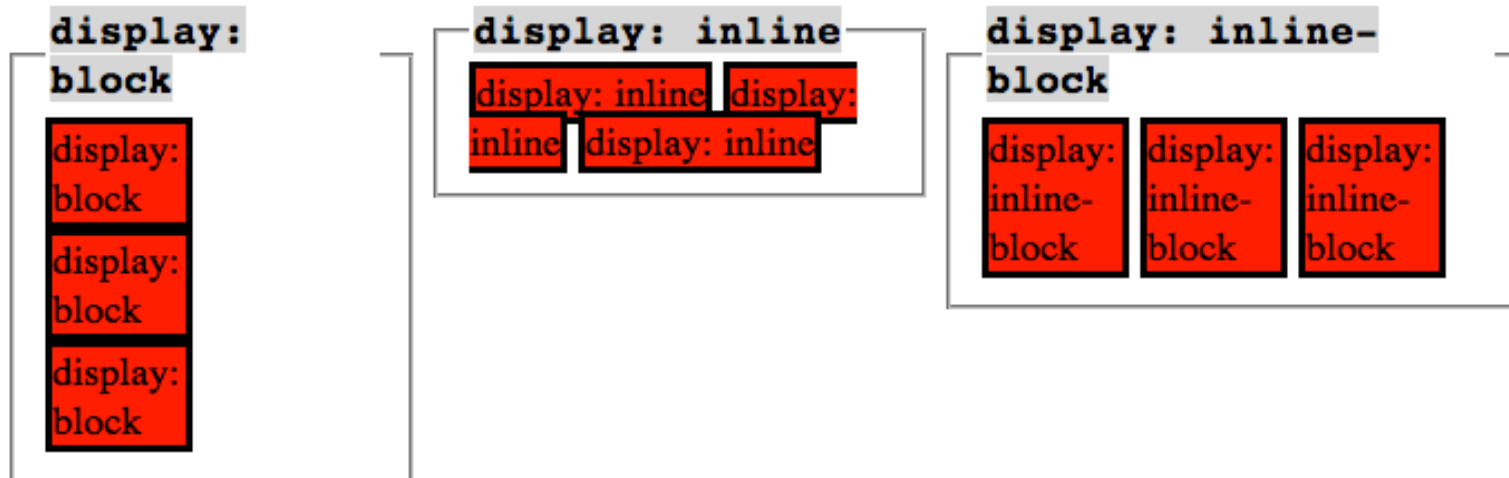
Cách hiển thị phần tử HTML với CSS Display, Visibility

❑ 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**;

block vs inline vs inline-block

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

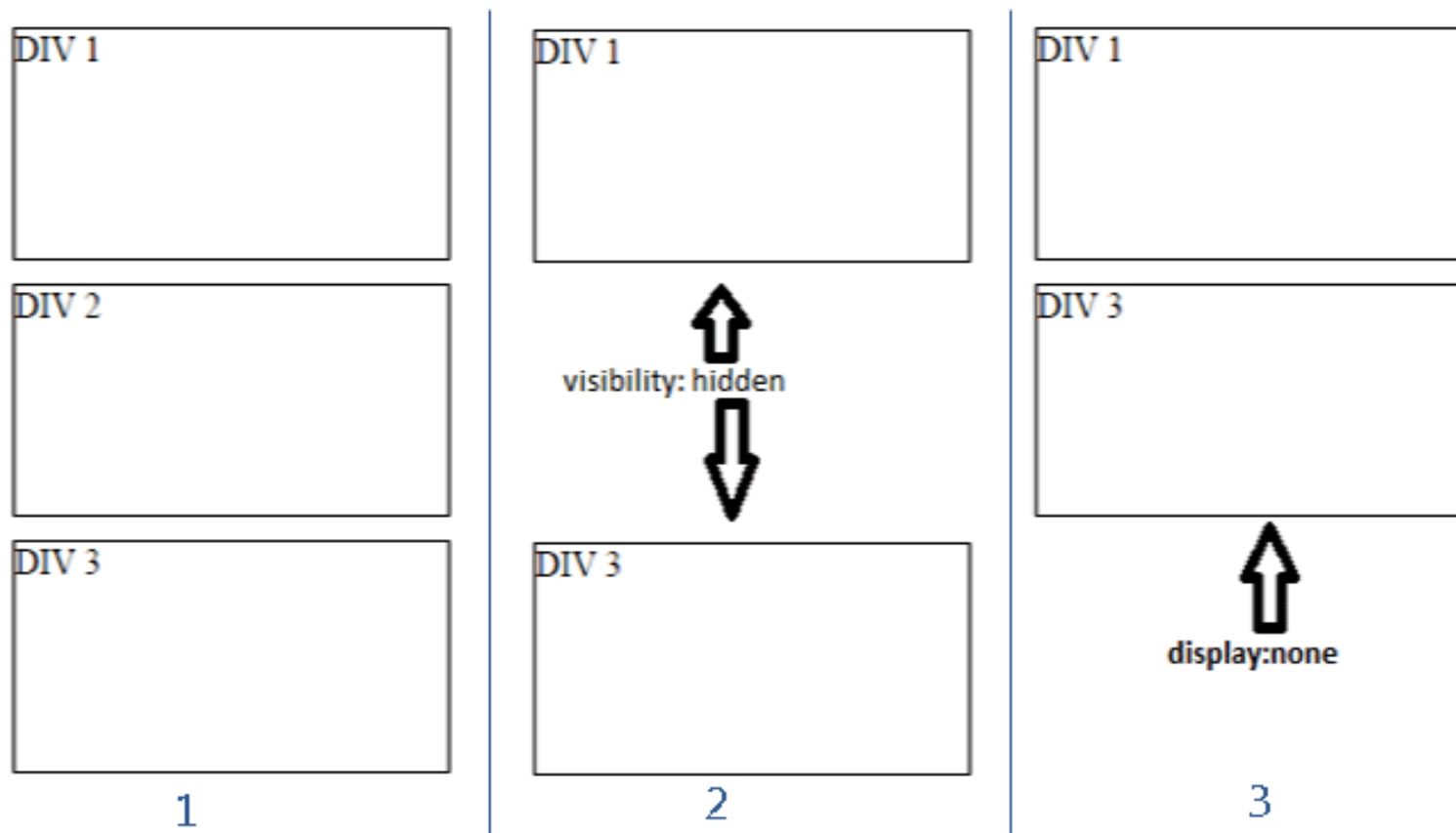




Cách hiển thị phần tử HTML với CSS Display, Visibility

❑ 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;**





Cách hiển thị phần tử HTML với CSS Display, Visibility

❑ Một số khai báo CSS về cách hiển thị 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>
```



Cách hiển thị phần tử HTML với CSS Display, Visibility

❑ Một số khai báo CSS về cách hiển thị thường dùng:

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

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

  p.ex1 {
    display: none;
  }

  p.ex2 {
    display: inline;
  }

  p.ex3 {
    display: block;
  }

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

Css Display

127.0.0.1:5501/CssDisplay.html

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.



Cách hiển thị phần tử HTML với CSS Display, Visibility

❑ Một số khai báo CSS về cách hiển thị thường dùng:

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

The screenshot shows a code editor on the left and a web browser on the right. The code editor displays the HTML and CSS for a file named 'ListStyleProperty.html'. The CSS includes a rule for the 'ul' element that sets 'list-style: circle;'. The browser window shows the rendered page with the title 'List Style Property' and a heading 'Fast Track Course'. Below the heading is an unordered list with three items: 'Frontend ReactJS', 'Backend Java', and 'Backend PHP', each preceded by a circular bullet point.

```
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>
```



Cách hiển thị phần tử HTML với CSS Display, Visibility

❑ Một số khai báo CSS về cách hiển thị thường dùng:

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

The screenshot shows a code editor on the left and a web browser on the right. The code editor displays the HTML and CSS for a file named 'ListStyleProperty.html'. The CSS includes a comment for a circle list style and a rule for 'list-style-image' pointing to a purple square GIF. The browser window shows the rendered page with the title 'Fast Track Course' and a list of three items: 'Frontend ReactJS', 'Backend Java', and 'Backend PHP', each preceded by a purple square icon.

```
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://www.w3schools.com/cssref/sqpurple.gif');
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>
```



Cách hiển thị phần tử HTML với CSS Display, Visibility

❑ Một số khai báo CSS về cách hiển thị thường dùng:

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

The screenshot shows a code editor on the left and a web browser on the right. The code editor displays the HTML and CSS for a file named 'ListStyleProperty.html'. The CSS includes two commented-out rules for 'ul' (one with 'list-style: circle;' and one with 'list-style-image: url('https://...')') and an active rule 'ul { list-style-position: outside; }'. The HTML structure includes a title 'List Style Property', a heading 'Fast Track Course', and an unordered list with three items: 'Frontend ReactJS', 'Backend Java', and 'Backend PHP'. The browser window on the right shows the rendered page with the heading 'Fast Track Course' and the list items. The list items are displayed with a square bullet point, which is a visual result of the 'list-style-position: outside;' CSS rule.

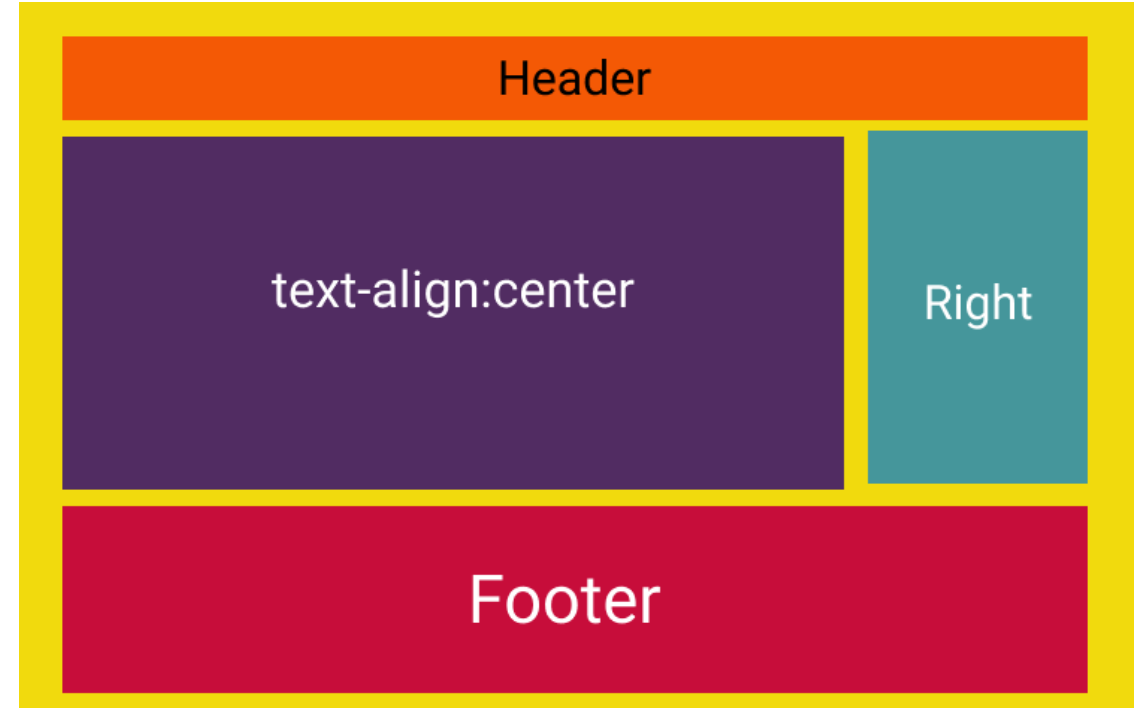
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>List Style Property</title>
5     <style>
6       /* ul {
7         list-style: circle;
8       } */
9
10      /* ul {
11        list-style-image: url('https://
12      } */
13
14      ul {
15        list-style-position: outside;
16      }
17    </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>
```



Canh chỉnh vị trí phần tử HTML với 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;**





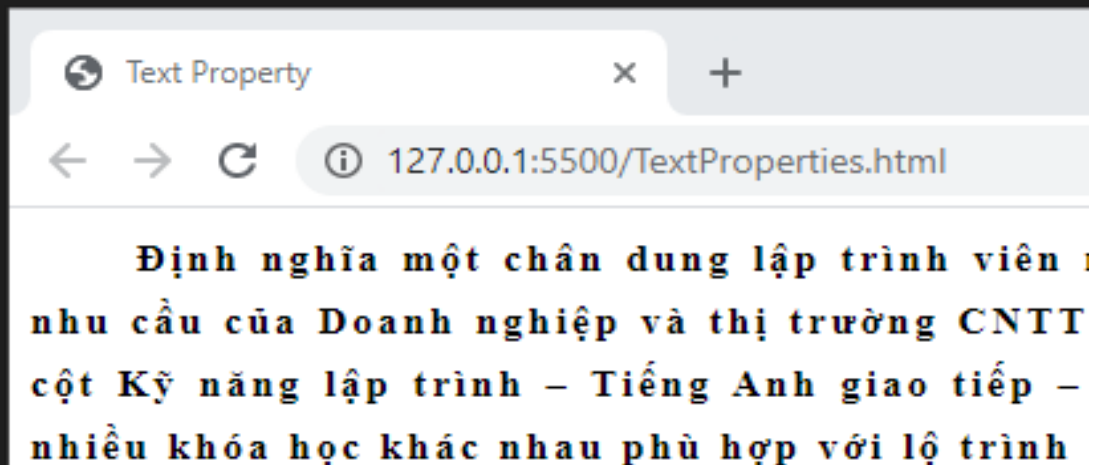
Canh chỉnh vị trí phần tử HTML với CSS Alignment

❑ 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 -->
<div class="text">
  Định nghĩa một chân dung
  Định nghĩa một chân dung
  Định nghĩa một chân dung
</div>
```





Canh chỉnh vị trí phần tử HTML với CSS Alignment

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

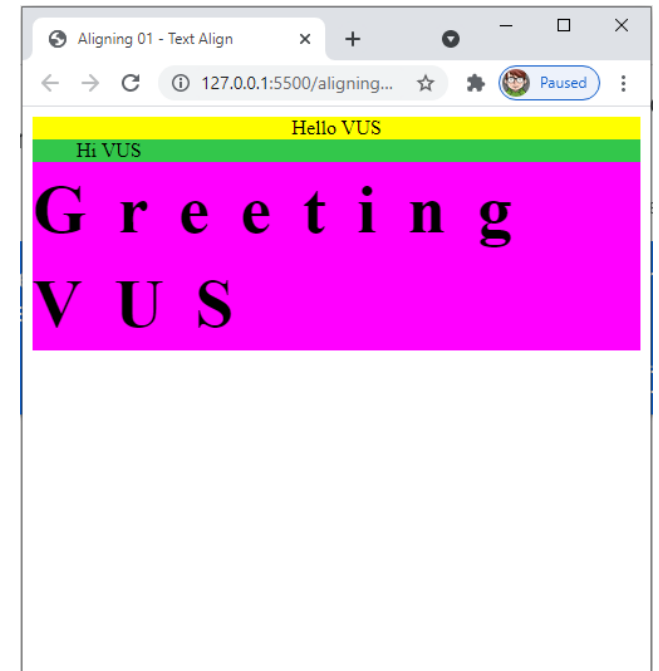
- ♦ Một cách để **căn chỉnh nội dung trang web** là sử dụng thuộc tính **text-align**.
- ♦ Sử dụng thuộc tính text-align để đặt căn lề ngang cho các dòng văn bản trong một phần tử.
- ♦ **text-align** sử dụng một trong 4 giá trị sau: **left** (mặc định), **right**, **center**, **justify**.

```
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>
```





- ◆ **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.

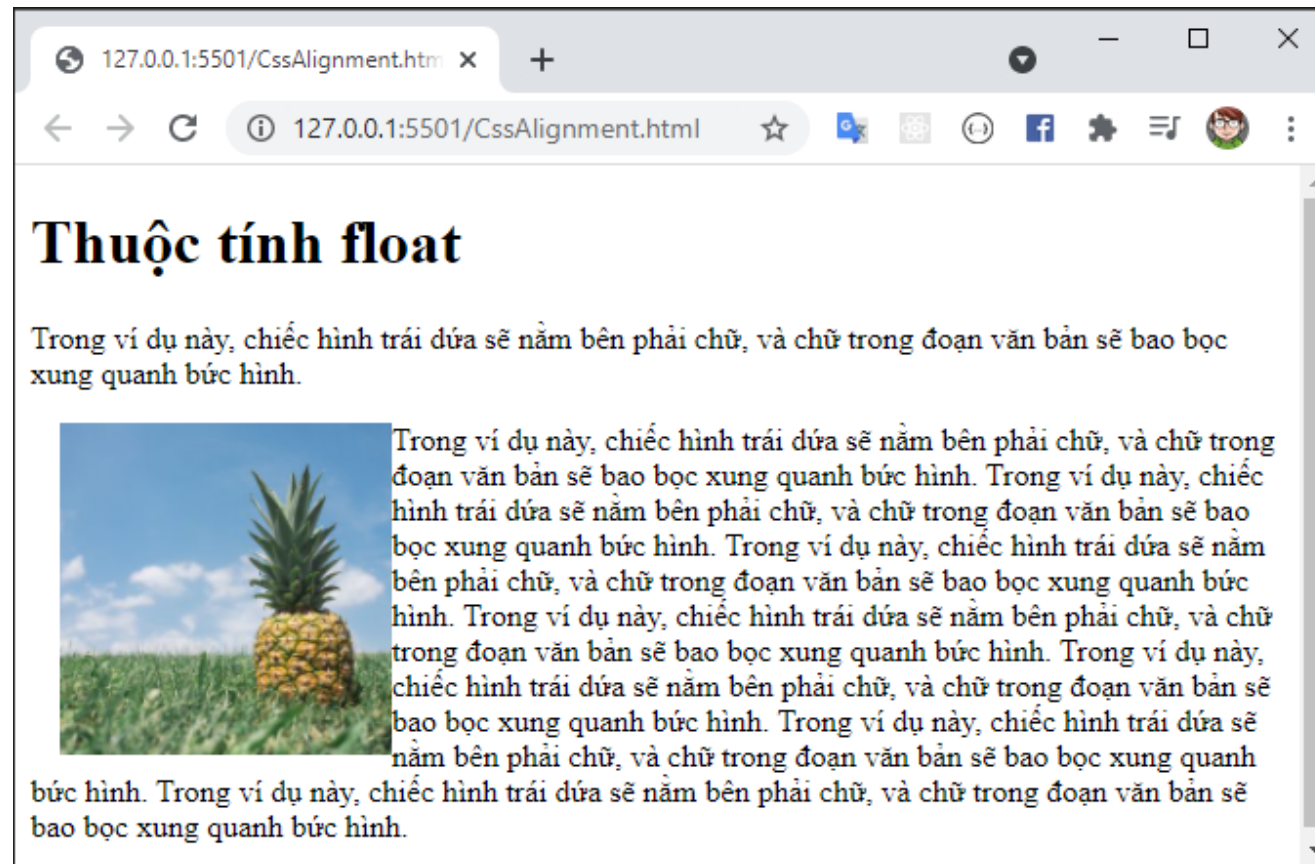
daynghevietuc.com



Canh chỉnh vị trí phần tử HTML với 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.





Canh chỉnh vị trí phần tử HTML với CSS Alignment

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

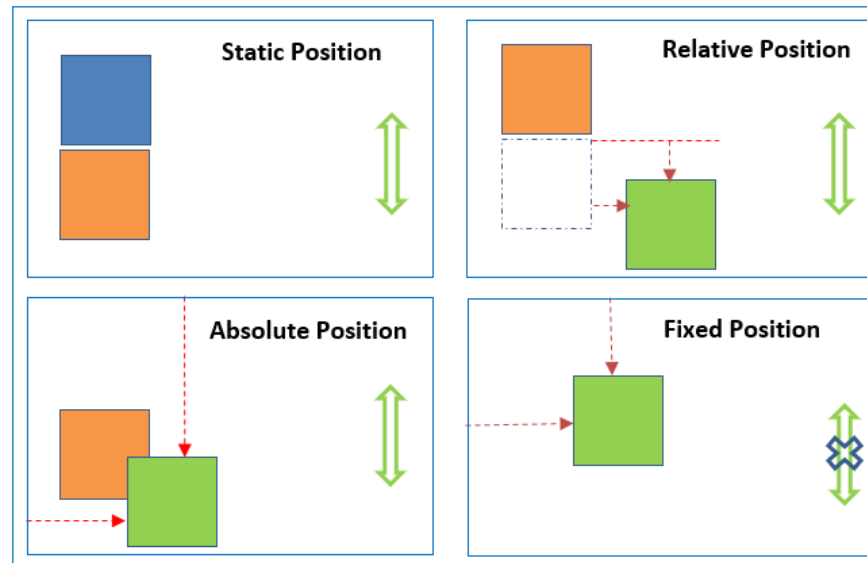
- ♦ **float** sẽ đẩy các phần tử block nằm trên cùng một hàng từ trái sang phải hoặc ngược lại.
- ♦ Nếu một hàng **không đủ chỗ chứa**, các phần tử sẽ **trôi xuống hàng tiếp theo**.
- ♦ Nếu các phần tử có **chiều cao khác nhau**, hàng tiếp theo sẽ **bắt đầu bên dưới phần tử cho chiều cao thấp nhất sau 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.



Canh chỉnh vị trí phần tử HTML với 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 với **position** để căn chỉnh vị trí phần tử là:
 - top, left, right, bottom, z-index



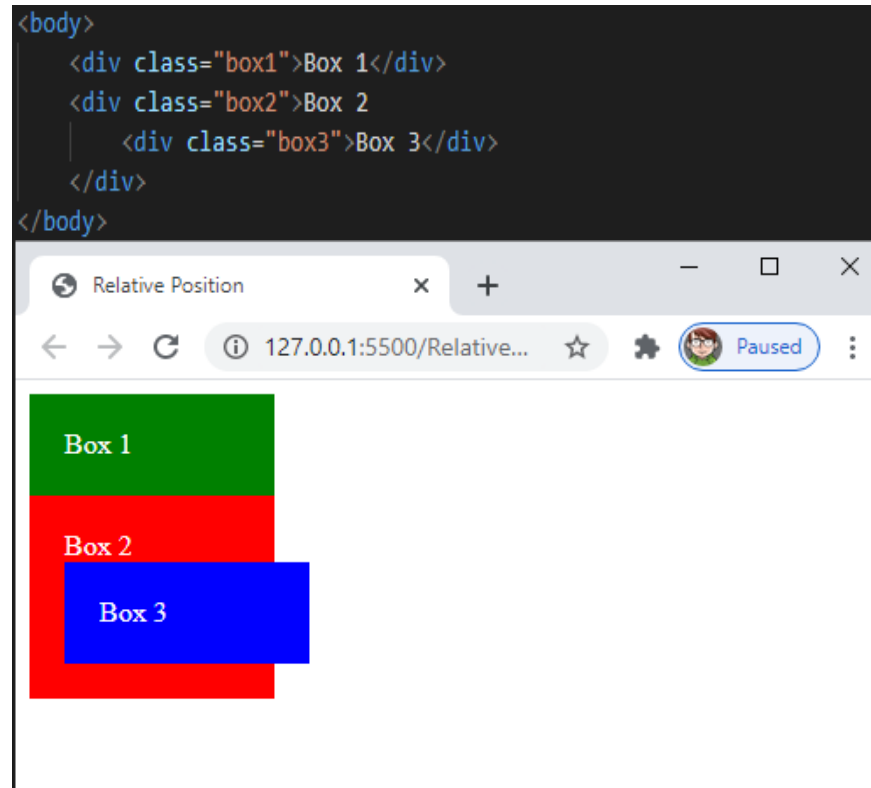


Canh chỉnh vị trí phần tử HTML với CSS Alignment

❑ 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>
```



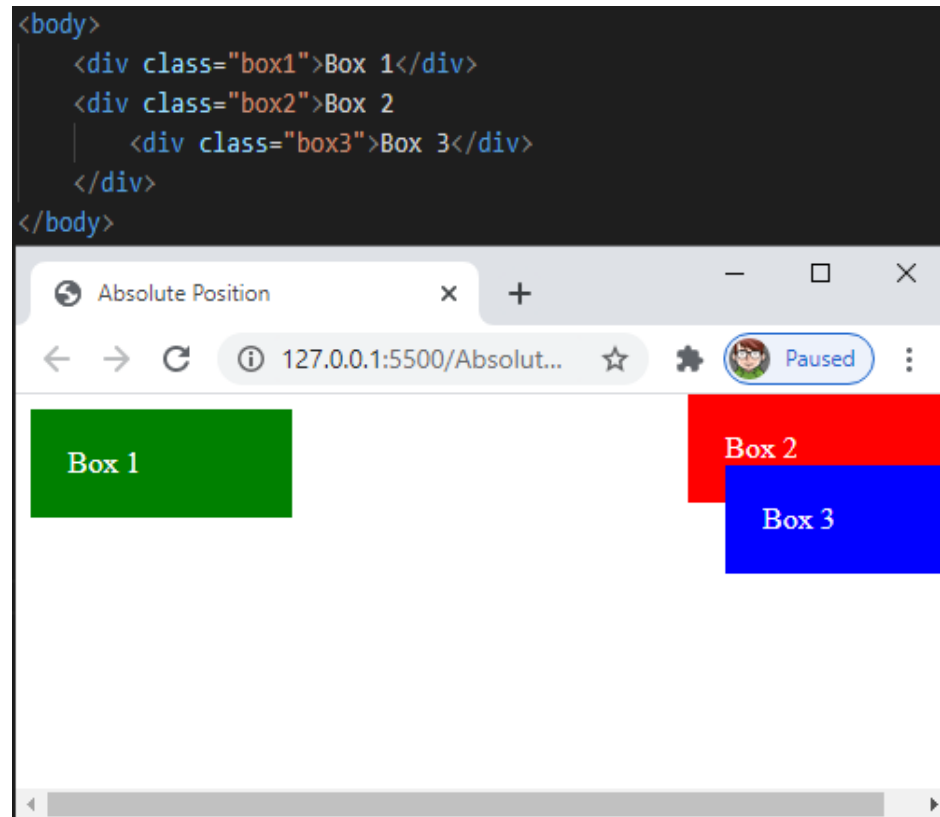


Canh chỉnh vị trí phần tử HTML với CSS Alignment

❑ 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.

```
<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>
```





Canh chỉnh vị trí phần tử HTML với 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>
```

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

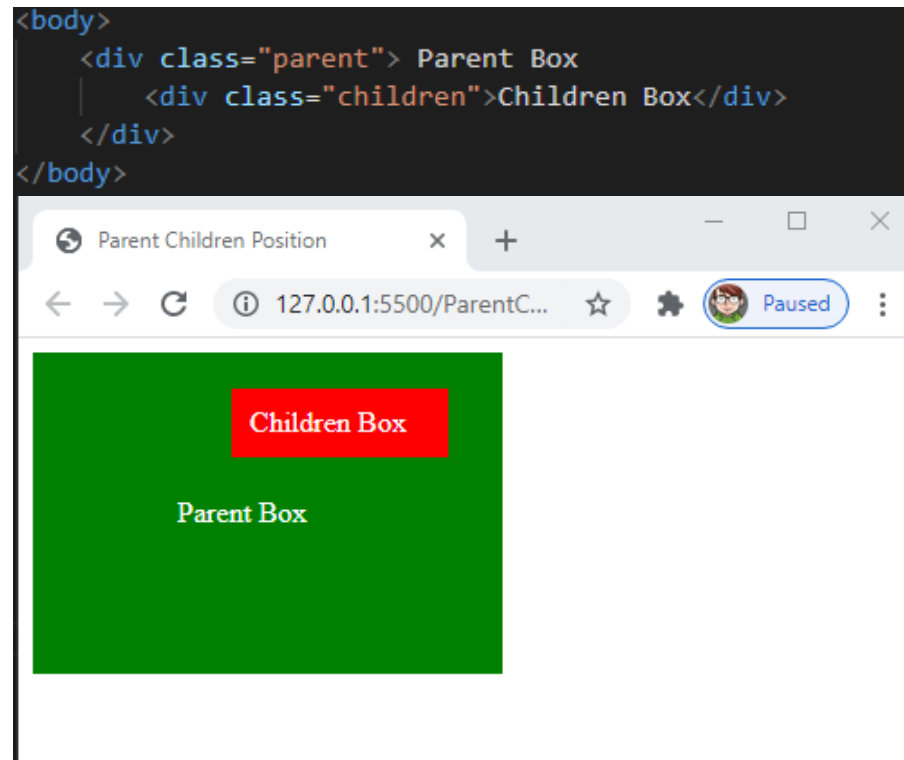


Canh chỉnh vị trí phần tử HTML với CSS Alignment

❑ 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>
```





Đường bao phần tử HTML với Border

❑ 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!



Đường bao phần tử HTML với Border

❑ 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 { ...
}

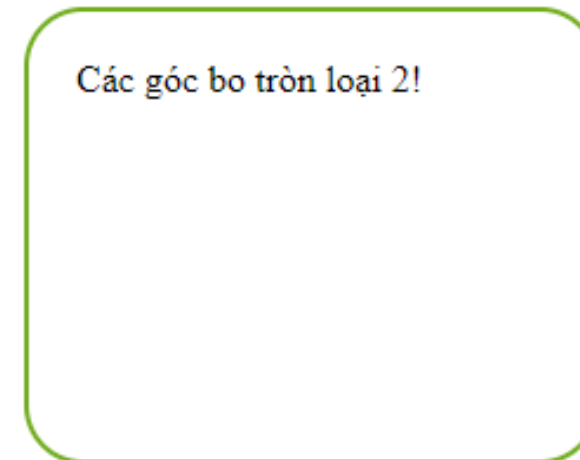
#rcorners2 {
border-radius: 25px;
border: 2px solid #73AD21;
padding: 20px;
width: 200px;
height: 150px;
}

#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:





Đường bao phần tử HTML với Border

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

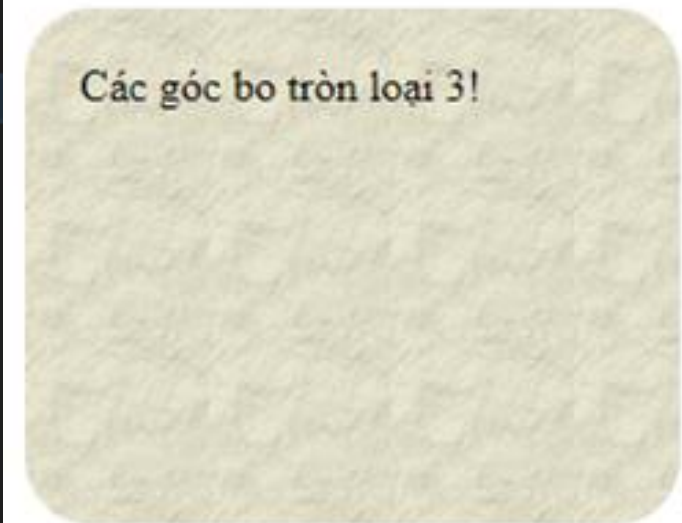
- ♦ Với thuộc tính **border-radius**, bạn có thể bo tròn các góc của bất kỳ phần tử nào.

```
<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:





Đường bao phần tử HTML với Border

❑ 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**)





Sắp xếp hiển thị nhiều hình nền với Background

❑ Khai báo CSS nhiều hình nền:

- ◆ CSS cho phép bạn thêm nhiều hình nền cho một phần tử, thông qua thuộc tính **background-image**.
- ◆ Nhiều hình nền ngăn cách nhau bởi dấu phẩy, và các hình ảnh được đặt trên các hình ảnh còn lại, nơi mà hình ảnh đầu tiên sẽ gần với người xem nhìn nhất.
- ◆ Ví dụ dưới đây có 2 hình nền:
 - Hình đầu tiên là một chùm hoa (đặt ở bên dưới góc phải màn hình), có đường dẫn là: https://www.w3schools.com/css/img_flwr.gif
 - Hình thứ 2 là một nền giấy báo (đặt ở bên trên góc trái màn hình), có đường dẫn là: <https://www.w3schools.com/css/paper.gif>



Sắp xếp hiển thị nhiều hình nền với Background

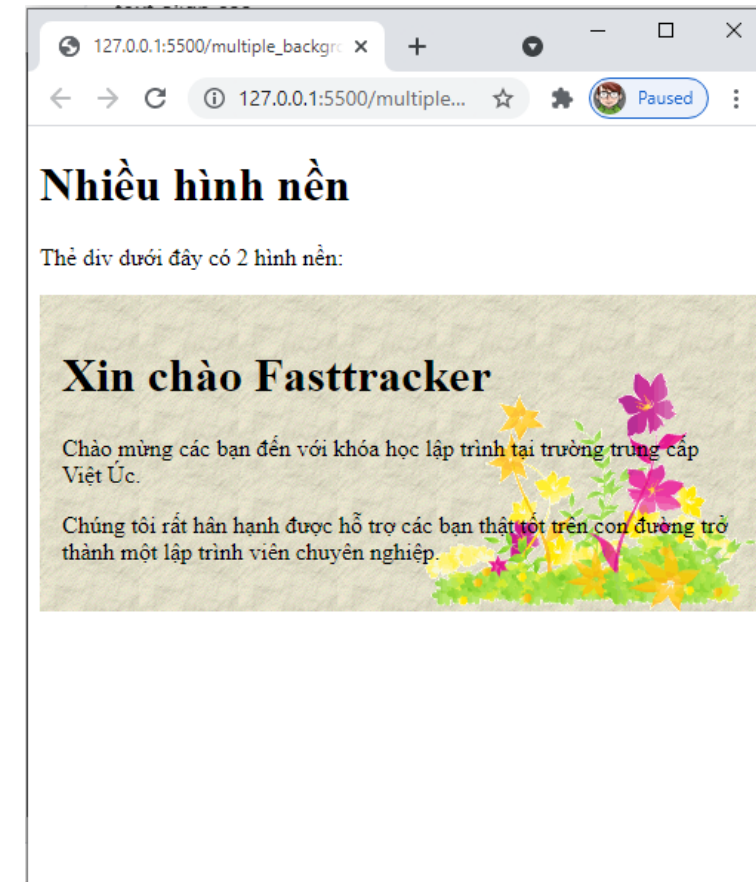
❑ Khai báo CSS nhiều hình nền:

◆ Ví dụ 2 hình nền chồng lên nhau:

```
<body>
  <h1>Nhiều hình nền</h1>
  <p>Thẻ div dưới đây có 2 hình nền:</p>

  <div id="example1">
    <h1>Xin chào Fasttracker</h1>
    <p>Chào mừng các bạn đến với khóa học lập trình tại trường trung cấp Việt Úc.</p>
    <p>Chúng tôi rất hân hạnh được hỗ trợ các bạn thật tốt trên con đường trở thành
      một lập trình viên chuyên nghiệp.</p>
  </div>
</body>
```

```
<style>
  #example1 {
    background-image: url(https://www.w3schools.com/css/img_flwr.gif),
                     url(https://www.w3schools.com/css/paper.gif);
    background-position: right bottom, left top;
    background-repeat: no-repeat, repeat;
    padding: 15px;
  }
</style>
```





Sắp xếp hiển thị nhiều hình nền với Background

❑ Khai báo CSS nhiều hình nền:

◆ Ví dụ 3 hình nền chồng lên nhau:

```
<body>
  <h1>Nhiều hình nền</h1>
  <p>Thẻ div dưới đây có 3 hình nền:</p>

  <div id="example1">
    <h1>Xin chào Fasttracker</h1>
    <p>Chào mừng các bạn đến với khóa học lập trình tại trường trung cấp Việt Úc.</p>
    <p>Chúng tôi rất hân hạnh được hỗ trợ các bạn thật tốt trên con đường trở thành
      một lập trình viên chuyên nghiệp.</p>
  </div>
</body>
```

```
<style>
  #example1 {
    background: url(https://www.w3schools.com/css/img_tree.gif) left top no-repeat,
               url(https://www.w3schools.com/css/img_flwr.gif) right bottom no-repeat,
               url(https://www.w3schools.com/css/paper.gif) left top repeat;
    padding: 15px;
    background-size: 50px, 130px, auto;
  }
</style>
```





Tổng kết nội dung bài học

- ☐ Định kiểu phần tử đoạn văn bản với CSS Fonts
- ☐ Thiết lập màu sắc phần tử HTML với CSS Colors
- ☐ Mô hình hộp khối CSS Box Model, Padding, Margin
- ☐ Cách hiển thị phần tử HTML với CSS Display, Visibility
- ☐ Canh chỉnh vị trí phần tử HTML với CSS Alignment
- ☐ Đường bao phần tử HTML với Border
- ☐ Sắp xếp hiển thị nhiều hình nền với Background

