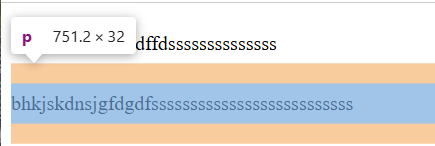
**CSS (9) - Định dạng dòng & đoạn**

* 1. Định dạng dòng & đoạn

**Định dạng dòng**

****



Đường màu xanh chính là line-height

Màu cam là đường margin

p { line-height: 2; }

p { line-height: 2em; }

p { line-height: 200%; }

Nếu giá trị của line-height là số, trình duyệt sẽ nhân số này với kích thước phông chữ của phần tử hiện tại để có được chiều cao của dòng văn bản. Tương tự, nếu giá trị của line-height là em hoặc phần trăm, thì đó là hai đơn vị đo tương đối, trình duyệt sẽ dựa trên nó để tính chiều cao của dòng văn bản.

css:

 h1 { font-size: 30px; }

    div { font-size: 15px; width: 10em; border: 1px solid red;}

    .div1 { line-height: 1.1; }

    .div2 { line-height: 1.1em }

html:

<div *class*="div1">

    <h1>Đoạn văn bản 1</h1>

 </div>

 <div *class*="div2">

    <h1>Đoạn văn bản 2</h1>

 </div>

h1 trong div1

Khi h1 k có line-height nó sẽ kế thừa từ cha là div1.

🡺 h1 có line-height : 1.1 (1.1 là giá trị số) tức là 1,1\*30

h1 trong div2

Khi h1 k có line-height nó sẽ kế thừa từ cha là div2.

🡺 h1 có line-height : 16.5(15\*1.1)em sẽ tính luôn trước khi kế thừa 🡺 Chúng ta nên sử dụng giá trị số để thiết lập cho line-height

**Thuộc tính text-indent:** sử dụng để thiết lập khoảng lùi của hàng đầu tiên trong một đoạn

**Margin & padding:** sử dụng để thiết lập khoảng lùi cho cả đoạn

css:

body {

    width: 500px;

    margin-left: 50px;

    border: 1px solid red;

}

p#doan1 { text-indent: 2em; } 🡪 2\*16 = 32 px

p#doan2 { text-indent: 25%; } 🡪 25%\*500 = 125px

p#doan3 { text-indent: -35px; }

html:

<body>

<p *id*="doan1"> </p>

<p *id*="doan2"> </p>

<p *id*="doan3"> </p>

</body>

Giá trị của thuộc tính text-indent có thể là em, % hoặc px.

Giá trị % được tính dựa trên độ rộng (width) của phần tử cha.

Giá trị -35px sẽ dịch chuyển hàng đầu tiên về phía bên trái của đoạn, phải đảm bảo là có thiết lập giá trị cho left margin, để đảm bảo phần chữ thò ra không bị che mất trên màn hình giao diện.

**Căn đoạn văn bản theo chiều ngang**

Để căn đoạn văn bản theo chiều ngang, sử dụng thuộc tính text-align.

Ý nghĩa các giá trị của thuộc tính text-align:

– Mặc định: start

– left, start: căn lề trái

– right, left: căn lề phải

– center: căn giữa

– justify: căn đều hai bên

**Gạch chân và gạch ngang**

Để gạch chân, gạch ngang, gạch trên, bỏ dấu gạch chân cho các liên kết, sử dụng thuộc tính text-decoration.

Ý nghĩa các giá trị của thuộc tính text-decoration:

– Mặc định: none

– underline: gạch dưới

– overline: gạch trên

– line-through: gạch giữa

**Định dạng chữ hoa, chữ thường**

Sử dụng thuộc tính text-transform để chuyển qua lại giữa các dạng chữ hoa và chữ thường, mà không cần phải gõ lại.

Ý nghĩa các giá trị của thuộc tính text-transform:

– capitalize: viết hoa kí tự đầu tiên của mỗi chữ

– lowercase: viết thường

– uppercase: viết hoa toàn bộ

– none: hiển thị như trong mã nguồn

**Khoảng cách giữa các kí tự, chữ**

Để thiết lập k/c giữa các kí tự, sử dụng thuộc tính letter-spacing.

Để thiết lập k/c giữa các chữ, sử dụng thuộc tính word-spacing.

Ví dụ,

p { letter-spacing: 8px; }

p { word-spacing: 1.5em; }

Lưu ý: khi giá trị của thuộc tính là em, lúc kế thừa cho phần tử con, nó sẽ chuyển cho phần tử con giá trị đã được tính ra px (chứ không phải giá trị dạng em).

### **Tạo bóng chữ**

Để tạo bóng cho chữ, sử dụng thuộc tính text-shadow.

text-shadow: ‘horizontal offset’‘vertical offset’‘blur radius’‘color’ | none

Trong đó:

– horizontal offset: là khoảng dịch của bóng sang phía phải so với văn bản, nếu giá trị này là âm, bóng sẽ được dịch sang phía trái

– vertical offset: là khoảng dịch của bóng xuống phía dưới so với văn bản, nếu giá trị này là âm, bóng sẽ được dịch lên phía trên

– color: màu của bóng, nếu bỏ qua giá trị này, bóng sẽ cùng màu với văn bản

Để bóng có tính mềm mại, cần sử dụng thêm giá trị blur-radius, đây là độ mờ của bóng, giá trị bằng 0 là không làm mờ, giá trị càng cao bóng sẽ càng mờ.

Ví dụ,

      text-shadow: -.3em -.3em .03em silver;

Ví dụ, tạo nhiều bóng,

      text-shadow: -.7em -.5em .2em silver,

                  .2em .2em .1em gray;

Ví dụ, tạo chữ nổi,

    body {background-color: thistle;}

    h1 {

      color: #ba9eba;

      text-shadow:

        -.05em -.05em .05em white,

        .03em .03em .05em purple;

    }

Ví dụ, tạo chữ khắc nổi,

    body {background-color: thistle;}

   h1 {

     color: #ba9eba;

    text-shadow:

      -.05em -.05em .05em purple,

     .03em .03em .05em white;

}

Ví dụ, làm chữ phát sáng,

h1 {

      text-shadow: 0 0 .7em purple;

    }

**Chọn kiểu**

Để chọn kiểu số hoặc dấu cho các mục của danh sách, sử dụng thuộc tính list-style-type.

Thuộc tính list-style-type:

– Giá trị: none | disc | circle | square | decimal | decimal-leading-zero | lower-alpha | upper-alpha | lower-latin | upper-latin | lower-roman | upper-roman | lower-greek

– Mặc định: disc

– Áp dụng: ul, ol, li (hoặc các phần tử có hiển thị là list-item)

Để đánh số hoặc dấu đầu dòng cho một phần tử bất kì, sử dụng thuộc tính display với giá trị là list-item, sau đó chọn một kiểu đánh số hoặc đánh dấu bằng thuộc tính list-style-type.

Ví dụ,

p.bulleted {

display: list-item;

list-style-type: upper-alpha;

}

**Xác định vị trí của số/dấu**

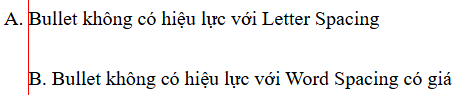
Mặc định, số/dấu sẽ nằm ngoài (outside) vùng nội dung của các mục (list-item), có thể sử dụng thuộc tính list-style-position để kéo số/dấu vào (inside) vùng nội dung của các mục.

Thuộc tính list-style-position:

– Giá trị: inside | outside

– Mặc định: outside

– Áp dụng: ul, ol, li (hoặc các phần tử có kiểu hiển thị là list-item)



Ngoài các kiểu dấu đầu dòng có sẵn, lập trình viên có thể sử dụng thuộc tính list-style-image để dùng các hình ảnh tùy ý làm dấu đầu dòng.

Thuộc tính list-style-image:

– Giá trị: url | none

– Mặc định: none

– Áp dụng: ul, ol, li (hoặc các phần tử có kiểu hiển thị là list-item)

Khi sử dụng thuộc tính list-style-image, nên sử dụng thêm thuộc tính list-style-type để dự phòng trong trường hợp không thể tải được hình ảnh của thuộc tính list-style-image. Ví dụ,

ul {

list-style-image: url(/images/happy.gif);

list-style-type: circle;

list-style-position: outside;

}