\* CSS định dạng text

- Thuộc tính thiết lập font chữ, font size, line height cho text

1. font chữ

Body{

Font-family: Arial, Helvetica, sans-serif;

}

2. kích thước fint chữ

p {

font-size: 16px;

/\*

Px: pixel ( cố định)

Em: tương đối với kích thước fint của phần tử cha

Rem: phần trăm so với kích thước của font phần tử cha

%: phần trăm so với kích thước font của cha

Vw: 1% chiều rộng viewport

\*/

}

3. chiều cao dòng (line-height)

p {

line-height: 1.5;

}

4. Các thuộc tính bổ sung:

* font-weight: Độ đậm của font (normal, bold, 100-900)
* font-style: Kiểu chữ (normal, italic, oblique)
* text-transform: Biến đổi chữ (uppercase, lowercase, capitalize)
* letter-spacing: Khoảng cách giữa các ký tự
* word-spacing: Khoảng cách giữa các từ

- Căn chỉnh text, chữ hoa thường, đậm nhạt text

1. căn chỉnh text

a. text-align: căn chỉnh chiều ngang

p {

text-align: left | right | center | justify;

}

b. vertical-align:căn chỉnh theo chiều dọc

span {

vertical-align: baseline | top | middle | bottom;

}

2. chữ hoa thường (text-transform)

p {

text-transform: uppercase | lowercase | capitalize | none;

/\*

Capitalize: viết hoa chữ đầu mỗi từ

None: giữ nguyên text gốc

\*/

}

3. đậm nhạt text

p {

font-weight: normal | bold | bolder | lighter | 100 | 200 | ... | 900;

/\*

normal: Bình thường (tương đương 400)

bold:Đậm (tương đương 700)

bolder: Đậm hơn phần tử cha

lighter: Nhạt hơn phần tử cha

100 đến 900: Các mức độ đậm nhạt cụ thể (100 nhạt nhất, 900 đậm nhất)

\*/

}

- in nghiêng text, đường gạch ngang text, hủy bỏ đường gạch chân cho text

1. in nghiêng text (font-style)

p {

font-style: italic;

}

2. đường gạch nang text (text-decoration)

span {

text-decoration: line-through;

}

3. hủy bỏ đường gạch chân cho text

a {

text-decoration: none;

}

- Thiết lập màu sắc cho text, các loại mã màu thường dùng

1. thiết lập màu sắc cho text (color)

p {

color: red;

}

2. các mã màu thường dùng

a. tên màu

color: red;

color: blue;

color: green;

b. mã hex

color: #FF0000; /\* Red \*/

color: #00FF00; /\* Green \*/

color: #0000FF; /\* Blue \*/

c. RGB:

color: rgb(255, 0, 0); /\* Red \*/

color: rgb(0, 255, 0); /\* Green \*/

color: rgb(0, 0, 255); /\* Blue \*/

d. RGBA (RGB với độ trong suốt)

color: rgba(255, 0, 0, 0.5); /\* Semi-transparent red \*/

e. HSL (hue, saturation, lightness)

color: hsl(0, 100%, 50%); /\* Red \*/

color: hsl(120, 100%, 50%); /\* Green \*/

color: hsl(240, 100%, 50%); /\* Blue \*/

f. HSLA(HLS với độ trong suốt)

color: hsla(0, 100%, 50%, 0.5); /\* Semi-transparent red \*/

\*CSS định dạng khối

- Thiết lập kích thước rộng cao cho khối

.box{

Width: 200px;

Height: 100px;

}

- thiết lập màu nền khối, thiết lập kích thước hình nền khối

.box{

Background-color: ....;

Background-size: ....;

}

- Thiết lập chế độ lặp hình nền khối

.box{

Background-repeat: .....;

}

- Thiết lập cố định hình nền khi cuộn trang

.box{

Background-attachment: ....;

}

- Background viết rút gọn

.box{

Background: .... urf(image.jpg);

}

- Tạo vùng đệm cho khối

.box {

padding: 10px;

}

- Tạo đường viền cho khối

.box {

border: 1px solid black;

}

- 2 cách xác định kích thước khối

.box {

border: 1px solid black;

}

- Tạo lề, tạo khoảng cách 2 khối

.box {

margin: 10px;

}

- Thuộc tính float điều khiển nhiều phần tử

.item {

float: left;

float: right;

float: none;

float inherit;

}

- Hiển thị danh sách bài viết vs float

.post {

float: left;

width: 33.33%;

}

- TÌm hiểu về sập độ cao khi sử dụng float, dung cách nào để xử lý được việc đó

- khi sử dụng float các phần tử con sẽ nổi lên trên và thoát khỏi luồng bình thường của tài liệu. Điều này có thể dẫn đến hiện tượng sập độ cao ( collapsing) của phần tử cha, nghĩa là phần tử cho không mở rộng để bao quanh các phần tử con đang float

- một số cách xử lý vấn đề

1. clearfix hack

.clearfix::after {

content: "";

display: table;

clear: both;

}

2. overflow: thêm thuộc tính overfow vào phần tử cha

.parent {

overflow: auto;

}

3. display: flow-root

.parent{

Display: flow-root;

}

4. clear: thêm phần tử trống vào cuối phần tử cha và áp dụng clear: both cho nó

<div class="parent">

<div class="float-child">Float child 1</div>

<div class="float-child">Float child 2</div>

<div style="clear: both;"></div>

</div>

5. sử dụng flexbox hoặc grid

.parent {

Display: flex;

}

.child{

Flex:1;

}

- Ẩn mọt phần tử HTML

.hidden{

Display:none

}

- thuộc tính hiển thị : inline, inline-block, block… ý nghĩa là gì ?

Inline: trên cùng một dòng

Inline-block: inline nhưng có thể set width/height

Block: hiển thị dạng khối chiếm toàn bộ chiều rộng

- thuộc tính flex, flex-direction row/column

.container {

Display: flex;

Flex-direction: row;/\* column\*/

}

- flex wrap, justify content

.container {

Flex-wrap; wrap;

Justify-content: space-between;

}

- thay đổi vị trí hiển thị order

.item {

Order:2;

}

- flex basis

.item {

flex-basis: 200px;

}

- flex grow

.item {

flex-grow: 1;

}

- flex shrink

.item {

flex-shrink: 0;

}

- Cách viết rút gọn flex

Đây là cách viết tắt của 3 thuộc tính riêng: flex-grow, flex-shrink, flex-basic

.item {

flex: [flex-grow] [flex-shrink] [flex-basis];

}

Một số cách rút gọn phổ biến

1. flex: 1; = flex 1 1 0%;

2. flex: auto; = flex: 1 1 auto;

3. flex: none; = flex: 0 0 auto;

4. flex: 0 auto; = flex 0 1 auto;

5. flex: [number]; = flex: [number] 1 0%;

6. flex: [width/height]; = flex: 1 1 [width/height];

- căn chỉnh flex-item theo phương thẳng đứng

.container {

align-items: center;

}

- thay đổi thuộc tính display trên đối tượng

.box {

display: inline-block;

}

- thuộc tính tịnh tiến khối so với ban đầu

.box {

transform: translate(10px, 20px);

}

- điều chỉnh vị trí khối con trôi nổi trên khối cha(absolute)

.parent {

position: relative;

}

.child {

position: absolute;

top: 10px;

left: 10px;

}

- Cố định một khối trong layout(thường dung làm back top)

.fixed-box {

position: fixed;

bottom: 20px;

right: 20px;

}

- xác định vị trí hiển thị khi xếp chồng layer

.top-layer {

z-index: 10;

}

- xử lý nội dung tràn khối

.box {

overflow: auto;

}

-