

CSS

Cascading Style Sheets



Khoa CNTT – HvKTMM
Giảng viên: ThS. Lê Đức Thuận



CSS - Cascading Style Sheets

1

Giới thiệu CSS

2

Sử dụng CSS

3

Sử dụng các thuộc tính trong CSS

4

Model-box và xếp chồng đối tượng

5

Sử dụng background



CSS - Cascading Style Sheets

6

Các thuộc tính cho văn bản

7

Một số thuộc tính và selector hay dung khác

8

Sắp xếp vị trí các khối

9

Tương thích website đa màn hình

10

Bài tập



1. Giới thiệu CSS

- ❖ Là ngôn ngữ định kiểu, đưa ra các chỉ thị để mô tả cách trình bày nội dung trong HTML
- ❖ CSS giúp tách biệt phần nội dung (HTML) và phần định dạng của một trang web
- ❖ Phần định dạng có thể nằm trong thẻ HTML, trong file hoặc riêng biệt thành file .css
- ❖ CSS giúp tái sử dụng mã => tiết kiệm thời gian làm việc
- ❖ Giảm mã HTML
- ❖ Độc lập thiết bị

2. Sử dụng CSS

❖ CSS được dung ngay trong thẻ HTML

```
<body>
  <!-- -->
  <p style="color: #0f0;">Sử dụng CSS trong thẻ HTML (CSS inline)</p>
</body>
```

❖ CSS sử dụng bên trong file HTML (trong thẻ style nằm trong thẻ head)

```
<head>
  <meta charset="UTF-8">
  <title>Bắt đầu với CSS</title>
  <style type="text/css">
    p {
      color: #f00;
    }
  </style>
</head>
```

**Thứ tự ưu
tiên???**

❖ CSS được viết thành file .css riêng và được liên kết

```
<head>
  <meta charset="UTF-8">
  <title>Bắt đầu với CSS</title>
  <link rel="stylesheet" type="text/css" href="./css/exampleCSS.css">
</head>
```



2. Sử dụng CSS (tiếp) Cú pháp của CSS

- ❖ CSS tương tác với tên thẻ HTML
- ❖ Tương tác với id của thẻ HTML
- ❖ Tương tác với tên class của thẻ html
- ❖ Tương tác với tên thuộc tính



Bộ chọn

Đơn vị tính

Mã màu



3. Bộ chọn thuộc tính

Bộ chọn	Ví dụ	Mô tả
*	*	Chọn tất cả các thành phần html
Tag	P	Chọn tất cả các thành phần cùng loại
tang01 tang02	Div p	Chọn tất cả thành phần <tang02> bên trong thành phần <tag 01>
Tag01, tang02	Div, p	Chọn tất cả thành phần <tang01> và <tang02>
Tag01+tag02	div + p	Chọn tất cả thành phần <tag02> được đặt kế và sau thành phần <tag01>
Tag01~tag02	div ~ p	Chọn tất cả thành phần <tag02> được sau thành phần <tag01>
Tag01>tag02	Div>p	Chọn tất cả thành phần <tag02> có thành phần cha là <tag01>
.class	Classname	Chọn tất cả các thành phần có cùng tên “class”
#id	#idName	Chọn tất cả các thành phần có cùng tên id
:active	A:active	Chọn các liên kết được kích hoạt
:after	P:affer	Thêm nội dung ngay sau thành phần

3. Bộ chọn thuộc tính (tiếp)

Bộ chọn	Ví dụ	Mô tả
:first-child	P:first-child	Chọn thành phần đầu tiên của thành phần cha chứa nó
:first-letter	P:first-letter	Chọn ký tự đầu tiên của một thành phần
:first-line	P:first-line	Chọn dòng đầu tiên của một thành phần
:focus	Input:focus	Thành phần sẽ focus khi được chọn
:hover	A: hover	Chọn các liên kết được hover (di chuyển chuột lên thành phần)
:lang(mã nn)	P:lang(Vi)	Chọn thành phần với thuộc tính ngôn ngữ có giá trị bắt đầu với “mã ngôn ngữ”
:link	A:link	Định dạng cho tất cả liên kết khi chưa được click
Visited	A:visited	Định dạng cho các thành phần đã được click
[Thuộc tính]	[target]	Chọn tất cả các thành phần có cùng thuộc tính (attribute)
[Thuộc tính = ngôn ngữ]	[lang =vi]	Được sử dụng để chọn tất cả các thành phần với thuộc tính có giá trị = “ngôn ngữ”
[Thuộc tính ~= giá trị]	[title~=myWeb	Chọn tất cả các thành phần với thuộc tính chứa một giá trị
[Thuộc tính = giá trị]	[target=_blank]	Chọn tất cả các thành phần với thuộc tính bằng giá trị

3. Bộ chọn thuộc tính (tiếp)

Bộ chọn	Ví dụ	Mô tả
Tag01~tag02	ul~p	Chọn tất cả thành phần 02 khi có thành phần 01 ở trước
[attribute^=value]	Img[src^="bnr_"]	Chọn tất cả thành phần với thuộc tính có giá trị bắt đầu bằng "value"
[attribute\$=value]	Img[src&=".gif"]	Chọn tất cả thành phần với thuộc tính có giá trị kết thúc bằng "value"
[attribute*=value]	Img[src*=""]	Chọn tất cả thành phần với thuộc tính có giá trị đặc biệt bằng "value"
:first-of-type	P:first-of-type	Chọn thành phần con đầu tiên hoặc duy nhất trong các thành phần cha
:last-of-type	P:last-of-type	Chọn thành phần con cuối cùng hoặc duy nhất trong các thành phần cha
:only-of-type	P:only-of-type	Chọn thành phần con trong các thành phần cha, khi thành phần cha có một thành phần con là chính nó. Trong thành phần cha có thể chứa nhiều thành phần con, tuy nhiên thành phần con được chọn phải là duy nhất, không được có từ 2 trở lên
:only-child	P:only-child	Chọn thành phần con trong các thành phần cha, khi thành phần cha có mỗi thành phần con là chính nó, không được chứa thành phần con khác. Chỉ chọn thành phần có thành phần cha, những thành phần độc lập không được chọn.
:nth-child(n)	P:nth-child(3)	Chọn thành phần thứ "n" trong thành phần cha. Chỉ chọn thành phần có thành phần cha, những thành phần độc lập không được chọn
:nth-last-child(n)	P:nth-last-child(3)	Chọn thành phần thứ "n" tính từ thành phần cuối trong thành phần cha. Chỉ chọn thành phần có thành phần cha.

3. Bộ chọn thuộc tính (tiếp)

Bộ chọn	Ví dụ	Mô tả
:nth-of-type(n)	P:nth-of-type(3)	Chọn thành phần thứ “n”
:nth-last-of-type(n)	P::nth-last-of-type(3)	Chọn thành phần thứ “n” từ phần phần cuối cùng của cha
:last-child	P:last-child	Chọn thành phần cuối cùng trong thành phần cha. Chỉ chọn thành phần có thành phần cha, những thành phần độc lập không được chọn
:root	:root	Chọn thành phần gốc của văn bản
:empty	P:empty	Chọn thành phần không chứa thành phần khác
:target	#event:target	Sử dụng trong liên kết anchor name (link neo)
:enabled	Input:enabled	Chọn thành phần được kích hoạt (enabled) thường sử dụng cho các thành phần của form
:disabled	Input:disabled	Chọn thành phần bị vô hiệu hóa (disable), thường sử dụng cho các thành phần của form
:checked	Input:checked	Chọn thành phần được check (checked) thường sử dụng cho các thành phần của form
:not (bộ chọn)	:not(p)	Chọn tất cả ngoại trừ bộ chọn trong ngoặc
::selection	::selection	Chọn phần tử được người dùng chọn

3. Bộ chọn thuộc tính (tiếp)

Thuộc tính	Ví dụ	Mô tả
Background	Background:#ff0000;	Định dạng màu nền
Border	Border: 1px solid #ff0000;	Định dạng đường viền cho thành phần
Border-collapse	Border-collapse: collapse;	Xác định viền của table có tách biệt ra hay không
Border-spacing	Border-spacing: 10px;	Xác định khoảng cách giữa các đường viền của các cột lân cận
Bottom	bottom: 10px;	Xác định vị trí dưới cùng của thành phần được định vị trí.
Caption-side	caption-side: bottom;	Xác định vị trí một chú thích của table .
Clear	clear: both;	Xác định 2 bên của phần tử (left, right), nơi mà phần tử float không được cho phép.
Clip	clip: rect(0,0,50px,10px);	Xác định đoạn cho thành phần khi sử dụng thuộc tính position có giá trị "absolute".
Color	color: #ff0000;	Xác định màu sắc cho text.
content	content: "."	Sử dụng kèm với bộ chọn ":before", ":after" để chèn nội dung được tạo.
counter-increment	counter-increment: section;	Gia tăng một hoặc nhiều counter (sắp xếp có thứ tự, có hiển thị số)

3. Bộ chọn thuộc tính (tiếp)

Thuộc tính	Ví dụ	Mô tả
counter-reset	counter-reset: subsection;	Tạo hoặc reset một hoặc nhiều counter.
cursor	cursor: pointer;	Xác định kiểu con trỏ chuột được hiển thị.
direction	direction: ltr;	Xác định hướng cho văn bản.
display	display: inline;	Xác định loại hiển thị của thành phần.
empty-cells	empty-cells: hide;	Xác định có hay không có đường viền và nền trong một cột rỗng của table
float	float: left;	Xác định có hay không một thành phần được float.
font	font: 12px arial,sans-serif;	Thiết lập font cho thành phần, bao gồm font chữ, độ rộng, ...
height	height: 50px;	Thiết lập chiều cao của thành phần.
left	left: 10px;	Xác định vị trí bên trái của thành phần định vị trí (như position)
letter-spacing	letter-spacing: 2px;	Tăng hoặc giảm khoảng cách giữa các ký tự trong đoạn text.
line-height	line-height: 1.5;	Thiết lập chiều cao giữa các dòng.

3. Bộ chọn thuộc tính (tiếp)

Thuộc tính	Ví dụ	Mô tả
list-style	list-style: decimal;	Thiết lập kiểu cho một danh sách.
margin	margin: 15px;	Canh lề cho thành phần.
max-height	max-height: 200px;	Thiết lập chiều cao tối đa của thành phần.
max-width	max-width: 900px;	Thiết lập chiều rộng tối đa của thành phần.
min-height	min-height: 100px;	Thiết lập chiều cao tối thiểu của thành phần.
min-width	min-width: 600px;	Thiết lập chiều rộng tối thiểu của thành phần.
outline	outline: dotted;	Định dạng các đường viền bao ngoài
overflow	overflow: scroll;	Xác định điều gì sẽ xảy ra nếu một thành phần box tràn nội dung.
padding	padding: 15px;	Thiết lập các thuộc tính padding trong một khai báo.
page-break-after	page-break-after: alway;	Xác định các phân chia văn bản ngay sau thành phần.
page-break-before	page-break-before: alway;	Xác định các phân chia văn bản ngay trước thành phần.

3. Bộ chọn thuộc tính (tiếp)

Thuộc tính	Ví dụ	Mô tả
page-break-inside	page-break-inside: always;	Xác định các phân chia văn bản ngay bên trong thành phần.
position	position: absolute;	Xác định loại của phương pháp định vị trí cho thành phần.
quotes	" " " "	Thiết lập các loại dấu ngoặc bao ngoài khi nhúng một trích dẫn.
right	right: 10px;	Xác định vị trí bên phải của thành phần định vị trí (như position)
table-layout	table-layout: fixed;	Thiết lập các thuật toán layout được sử dụng cho table.
text-align	text-align: center;	Sắp xếp các nội dung theo chiều ngang.
text-decoration	text-decoration: underline;	Xác định các trang trí thêm cho text.
text-indent	text-indent: 10px;	Ghi rõ thụt đầu dòng của dòng đầu tiên trong một khối văn bản.
text-transform	text-transform: uppercase;	Thiết lập các ký tự viết hoa cho văn bản.
top	top: 10px;	Xác định vị trí bên trên của thành phần định vị trí (như position)
vertical-align	vertical-align: middle;	Sắp xếp các nội dung theo chiều dọc.



3. Bộ chọn thuộc tính (tiếp)

[illegible]

3. Bộ chọn thuộc tính (tiếp)

Thuộc tính	Mô tả
animation	Xác định một chuyển động của một thành phần.
appearance	Định dạng cho thành phần trông như giao diện chuẩn gần với người dùng.
backface-visibility	Xác định bề mặt sau của thành phần khi thực hiện một chuyển động xoay.
background-clip	Xác định vùng background được cắt bớt theo vùng được giới hạn.
background-origin	Xác định giá trị tương đối của background giới hạn theo vùng giới hạn.
background-size	Xác định lại chiều rộng và chiều cao cho background.
background gradient	Tạo màu sắc cho background theo biên độ giảm dần.
Nhiều background	Sử dụng để khai báo nhiều dạng background khác nhau trong cùng một tag.
border-image	Dùng để định dạng các dạng border bằng hình ảnh.
border-radius	Dùng để định dạng các dạng bo góc của border.
box-align	Xác định vị trí cho thành phần theo chiều dọc hoặc theo chiều thẳng đứng.
box-direction	Xác định hướng cho thành phần.
box-flex	Xác định sự ưu tiên linh hoạt theo các thành phần khác.
box-ordinal-group	Cho biết thứ tự ưu tiên của các thành phần.

3. Bộ chọn thuộc tính (tiếp)

Thuộc tính	Mô tả
box-orient	Xác định thành phần dọc theo phương hướng khối hoặc theo trục.
box-pack	Định vị trí của thành phần theo mép rìa của thành phần.
box-sizing	Xác định lại chiều rộng và chiều cao của thành phần.
box-shadow	Định dạng bóng cho thành phần.
column	Dùng để chia nội dung thành phần thành nhiều cột khác nhau.
@font-face	Định dạng các dạng font chữ khác nhau theo các dạng font riêng.
font-size-adjust	Dùng để định dạng điều chỉnh cho font chữ, độ lớn của chữ được thể hiện bởi phép nhân.
@keyframes	Dùng để điều khiển diễn biến một hoạt động của thành phần, được dùng kèm với thuộc tính animation .
nav	Di chuyển qua lại giữa các thành phần điều hướng (navigate) bằng cách di chuyển các phím mũi tên.
opacity	Hiển thị cấp độ trong suốt cho thành phần.
perspective	Cho ta thấy được chiều sâu của thành phần trong khai báo 3D.
perspective-origin	Định nghĩa trục quay cho thành phần có sử dụng perspective.
resize	Định dạng cho vùng nội dung mà người dùng có thể thay đổi được kích thước.

3. Bộ chọn thuộc tính (tiếp)

Thuộc tính	Mô tả
text-justify	Tăng hoặc giảm khoảng cách giữa các từ và giữa các ký tự sao cho dàn đều thành phần.
text-overflow	Xác định vùng text được cắt bớt.
text-shadow	Xác định bóng đổ cho text.
transform	Xác định một chuyển đổi 2 chiều, 3 chiều, có thể là xoay, tỷ lệ, di chuyển, nghiêng, ...
transform-origin	Xác định trục cho một chuyển đổi 2 chiều, 3 chiều.
transform-style	Các thành phần bên trong sẽ giữ vị trí 3D của nó.
transition	Xác định một quá trình chuyển đổi khi có một hành động.
word-break	Sẽ làm cho những chữ trong một từ không còn là một thể thống nhất, nghĩa là có thể xuống hàng bất cứ vị trí nào trong từ.
word-wrap	Sẽ làm cho những từ dài xuống hàng mà không làm vỡ layout.



Đơn vị tính trong html và css

Đơn vị	Mô tả
%	Phần trăm
Cm	centimeter
in	inch (1inch = 2.54 cm)
mm	millimeter
em	1em ngang bằng với kích cỡ font hiện hành, 2em = 2 lần kích cỡ font hiện hành.
ex	1 ex bằng chiều cao của chữ x thường của font hiện hành. Đơn vị này phụ thuộc vào kích cỡ font chữ và loại font chữ
pt	point (1pt = 1/72inch)
pc	pica (1pc = 12pt)
px	pixels (điểm ảnh trên màn hình máy tính)

- ❖ % thường dùng để định dạng phần trăm cho chiều rộng hay khoảng cách, font size.
- ❖ Em: Dùng cho những nội dung có khả năng co giãn khi zoom
- ❖ Px: dùng để định dạng khoảng cách, chiều rộng, chiều cao



Giá trị màu trong HTML CSS

- ❖ Màu theo hệ thập lục phân – HEX
- ❖ Màu theo kết hợp màu RGB
- ❖ Màu theo tên tiếng anh



Giá trị màu trong HTML CSS

❖ Màu theo hệ thập lục phân – HEX

- Cách viết: gồm dấu “#” kết hợp với dãy ký tự màu (chữ hoặc số thập lục phân – hoa và thường giống nhau)
- Ví dụ: #00Aa67
- 6 ký tự màu chia thành 3 cặp: nếu 2 ký tự trong 1 cặp giống nhau có thể viết thành 1 ký tự:
 - #224488 => #248



Giá trị màu trong HTML CSS

- ❖ **Màu RGB:** xác định màu theo sự kết hợp của 3 màu cơ bản: đỏ (red), xanh lá (green) và xanh dương (blue)
- ❖ **Cách viết:** `rgb(giá trị 1, giá trị 2, giá trị 3)`
 - Giá trị từ 0 – 255
 - Ví dụ: `rgb(100,50,1)`
- ❖ **Giá trị màu đặc biệt:**
 - Đen: 0,0,0 đỏ: 255,0,0 xanh lá: 0,255,0
 - Xanh dương: 0,0,255 vàng: 255,255,0
 - Trắng: 255,255,255



Giá trị màu trong HTML CSS

- ❖ Màu theo tên tiếng anh:
- ❖ Cách viết: theo phiên âm tiếng anh
 - Ví dụ: blue, red
- ❖ Một số màu hay dùng:
 - Đen: black
 - Đỏ: red
 - Xanh lá cây: green
 - Xanh dương: blue
 - Vàng: yellow
 - Trắng: white

CSS phần 3

BOX MODEL

Border

Margin

Padding

Xếp chồng các đối tượng

Display

Opacity

4. BOX MODEL và các thuộc tính

❖ **Box Model:** Mô tả khoảng cách mà mỗi phần tử trên website được sở hữu

- Margin: Khoảng cách tính từ bên ngoài của phần tử
- Border: Đường viền phần tử
- Padding: Khoảng cách tính từ bên trong của phần tử
- Content: Nội dung trong phần tử



```
01 | margin: top right bottom left;  
02 |  
03 | border: top right bottom left;  
04 |  
05 | padding: top right bottom left;
```

4. Border (tiếp)

❖ Đường viền của phần tử

- Màu sắc
- Độ dày đường viền
- Loại nét vẽ đường viền (nét liền, nét đứt, ...)

4. Border (tiếp)

❖ Thuộc tính **border-color**:

- Chỉ thị màu của đường viền.
- Giá trị của thuộc tính này có thể là một từ (tiếng anh) chỉ định màu hoặc mã màu
 - ❖ Ví dụ: **border-color:blue;** HOẶC **border-color:#666666;**

4. Border (tiếp)

❖ Thuộc tính **border-width**:

- Chỉ thị độ dày của đường viền
- Giá trị của thuộc tính này là một giá trị tính theo đơn vị px, pt, cm, em, ... phổ biến nên dùng px
 - ❖ Ví dụ: **border-width:1px;**

4. Border (tiếp)

❖ Thuộc tính **border-style**:

- Chỉ định loại nét vẽ của đường viền
- Giá trị của thuộc tính này là một trong danh sách sau: **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset**, **outset**, **none**, **hidden**.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

❖ Ví dụ: **border-style:solid;**

Border

❖ **Border:** Tạo viền cho phần tử

■ Cú pháp: `border: [size] [type] [color];`

■ Ví dụ: `border: 1px solid red;`

■ Border hỗ trợ một số kiểu như:

- Solid
- Dotted
- double
- Groove
- Ridge
- Inset
- Outset

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

■ Thuộc tính con: `border-top`, `border-right`, `border-bottom`, `border-left`.

4. Border (tiếp)

❖ Thuộc tính **border**:

❖ Là thuộc tính viết tắt trong việc chỉ định đường viền. Do 3 giá trị tương ứng màu sắc, loại nét vẽ và độ dày có định dạng hoàn toàn không trùng nhau nên chỉ cần liệt kê ra thành giá trị của thuộc tính **border** đơn lẻ cho nhanh.

❖ Ví dụ: **border : solid #524521 2px;**

❖ **border-left-color** dùng để chỉ định màu cho đường viền cạnh trái

❖ **border-top-style** dùng để chỉ định loại nét vẽ cho đường viền cạnh trên

❖ **border-right** là lối viết tắt dùng để chỉ ra các giá trị cho đường viền cạnh phải

❖ **border-bottom** là lối viết tắt cho đường viền cạnh dưới

4. Border (tiếp)

❖ Thuộc tính **border-radius** trong CSS3:

❖ Được bổ sung vào CSS3 nhằm thực hiện việc bẻ cong đường viền tại 4 góc nhọn của cạnh của khối. Giá trị của thuộc tính này là số đường kính đường tròn tạo đường cong. Nguyên tắc như sau:

- 4 bộ giá trị: tương ứng với góc trái-trên, phải trên, phải dưới, trái dưới.
- 3 bộ giá trị: tương ứng với góc trái trên, phải trên + phải dưới, trái dưới.
- 2 bộ giá trị: tương ứng với góc trái trên + phải trên, phải dưới + trái dưới
- 1 giá trị: áp dụng cho cả 4 góc cùng 1 đường kính cong

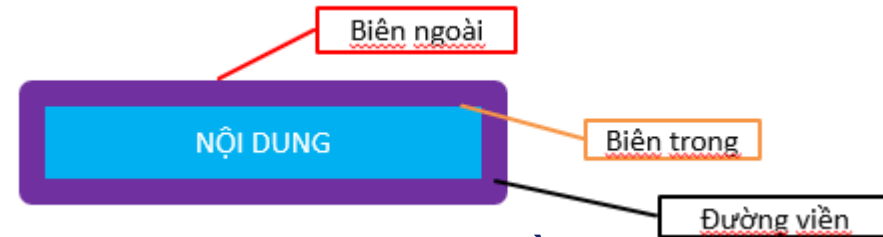
❖ Ví dụ: `border-radius : 5px 10px 3px 15px;` HOẶC `border-radius : 5px 10px;`

4. Margin và Padding (tiếp)

❖ Thuộc tính **margin-top**, **margin-right**, **margin-bottom**, **margin-left** của CSS

- Dùng để chỉ định khoảng cách từ biên ngoài của phần tử này với giới hạn của phần tử liền kề theo các hướng lần lượt là **TRÊN, PHẢI, DƯỚI, TRÁI**.

Ví dụ: **margin-top:10px** hoặc **margin-left:5%**



❖ Thuộc tính **margin**:

- Là cách viết tắt để định khoảng cách từ biên ngoài của phần tử này với giới hạn của phần tử liền kề. Có những cách khai báo giá trị sau:

- ❖ Viết 1 giá trị: áp dụng cả 4 hướng cùng 1 giá trị như nhau → **margin:10px;**
- ❖ Viết 2 giá trị: giá trị 1 áp dụng cho trên và dưới, giá trị 2 áp dụng cho trái và phải → **margin:10px 15px;**
- ❖ Viết 4 giá trị: các giá trị lần lượt áp dụng theo hướng trên, phải, dưới, trái → **margin:10px 15px 20px 25px;**

4. Margin và Padding (tiếp)

❖ Giá trị **auto** của thuộc tính **margin**

- Giá trị này dùng để định vị phần tử này nằm ở giữa theo chiều ngang bên trong phần nội dung của phần tử chứa nó.
- ❖ Ví dụ: **margin:0px auto;** hoặc **margin:auto;**

4. Margin và Padding (tiếp)

❖ Vấn đề tự thu gọn giá trị margin

- Đôi lúc, giá trị **margin-bottom** và **margin-top** của 2 phần tử liền kề có thể bị lược bỏ 1 giá trị và lấy giá trị lớn nhất. Điều này không xảy ra với left và right.
- Ví dụ 1: thẻ h2 đặt sau thẻ h1 thì 2 thẻ này sẽ chỉ cách nhau 30px chứ ko phải $20px + 30px = 50px$
- Ví dụ 2: thẻ h1 đặt đầu tiên và trong thẻ div1 thì thẻ div1 sẽ thay giá trị margin-top thành 30px.

```
h1 { margin: 0 0 20px 0;
}
h2 { margin: 30px 0 0 0;
}
```

```
#div1 { margin: 20px 0 0 0; }
h1 { margin: 30px 0 0 0; }
```


4. Margin và Padding (tiếp)

❖ Padding tương tự như Margin



4. Điều khiển hiển thị trong HTML

Xếp chồng các đối tượng

It.kma

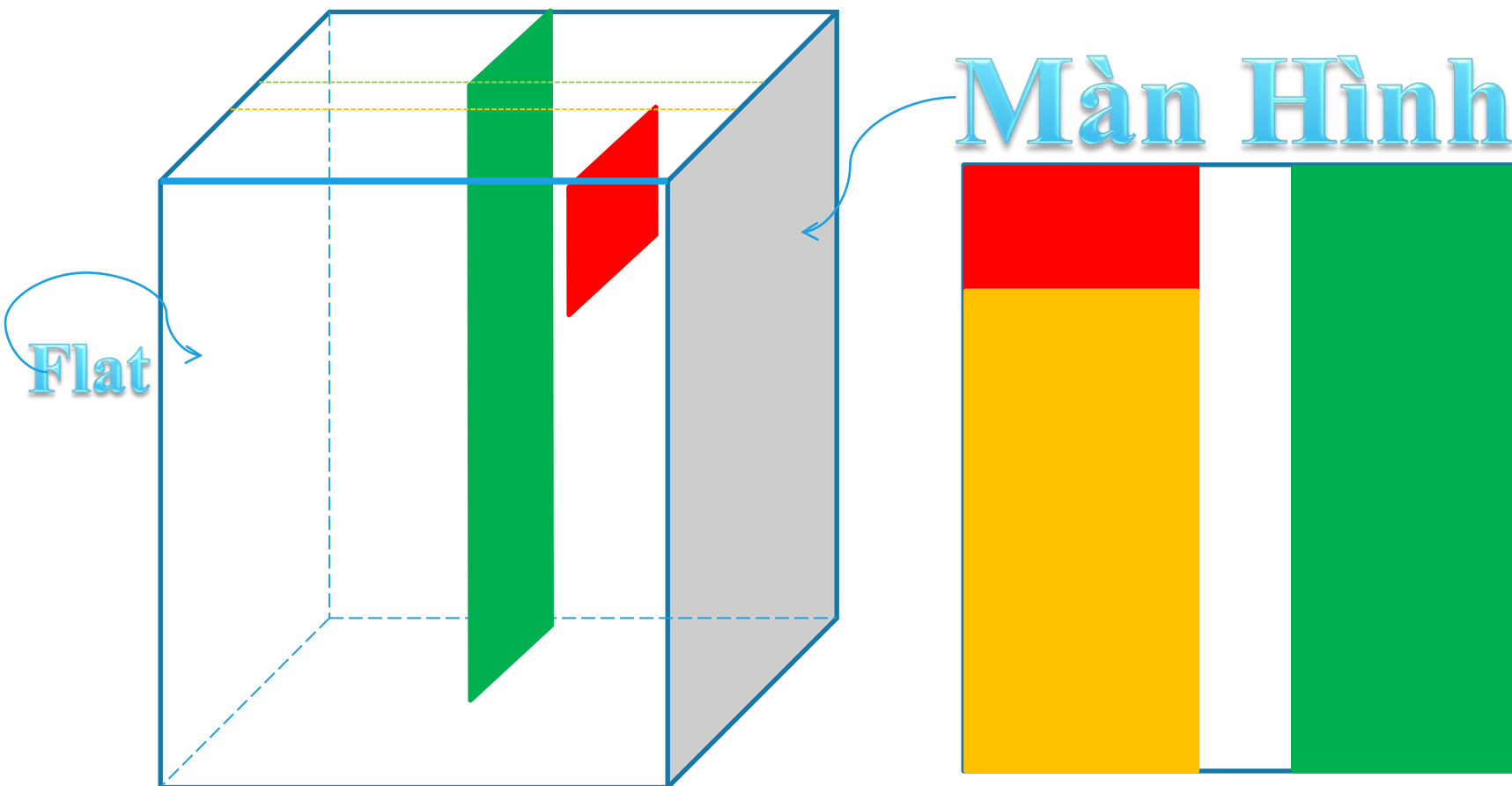
- ❖ Khung nhìn của các đối tượng xếp chồng.
- ❖ Thuộc tính display
- ❖ Thuộc tính Opacity



4. Điều khiển hiển thị trong HTML

Xếp chồng các đối tượng

It.kma





4. Điều khiển hiển thị trong HTML

Xếp chồng các đối tượng

It.kma

❖ Thuộc tính **z-index** của CSS

- Thuộc tính này dùng để chỉ định độ sâu (chiều Z trong không gian 3 chiều). Phần tử có z-index nhỏ hơn sẽ nằm phía sau và sẽ bị che ở những chỗ mà đã bị những phần tử có z-index cao hơn chiếm lấy.



4. Điều khiển hiển thị trong HTML It.kma

Xếp chồng các đối tượng

❖ Thuộc tính **left**, **right**, **top**, **bottom** của CSS (chỉ định giá trị tọa độ 2D)

- left để chỉ định tọa độ chiều ngang (tính từ mép biên bên trái của hệ tọa độ)
- right để chỉ định tọa độ chiều ngang (tính từ mép biên bên phải của hệ tọa độ)
- top để chỉ định tọa độ chiều dọc (tính từ mép biên bên trên của hệ tọa độ)
- bottom để chỉ định tọa độ chiều dọc (tính từ mép biên bên dưới của hệ tọa độ)



4. Điều khiển hiển thị trong HTML It.kma

Xếp chồng các đối tượng

❖ Thuộc tính **left**, **right**, **top**, **bottom** của CSS (chỉ định giá trị tọa độ 2D)

- Chỉ được và phải sử dụng 2 thuộc tính khác chiều nhau. Một thuộc tính chiều ngang kết hợp một thuộc tính chiều dọc.
 - left : 20px ; top : 10px Nghĩa là cách trái 20px và cách trên 10px
 - left : 20px ; bottom : 10px Nghĩa là cách trái 20px và cách dưới 10px
 - right : 10px ; top : 10px Nghĩa là cách phải 10px và cách trên 10px



4. Điều khiển hiển thị trong HTML It.kma

Xếp chồng các đối tượng

❖ Thuộc tính **position** - Chỉ một thẻ html sẽ hiển thị như thế nào?

- **static** : giá trị mặc định, thẻ html hiển thị bình thường theo nguyên tắc văn bản và tờ giấy (hay còn gọi là document flow)
- **absolute**: thẻ html sẽ là một viewBox và lấy tọa độ 2D dựa vào phần tử viewBox cha của nó (nếu có). Nếu không có phần tử cha nào là viewBox thì tọa độ 2D sẽ là viewBox gốc của tài liệu html.
- **fixed** : thẻ html sẽ là một viewBox, lấy tọa độ 2D theo màn hình. Có nghĩa là hiển thị sẽ luôn bị cố định dù người dùng có scroll chuột đi đâu cũng thấy như nhau.
- **relative**: thẻ html sẽ là một viewBox và lấy tọa độ 2D dựa vị trí tĩnh của chính nó hiện tại.



Kiểu hiển thị

❖ Kiểu hiển thị

- **INLINE** : như từng chữ trong một bài văn, kiểu hiển thị này sắp liền kề nhiều phần tử vào nhau cho tới khi hết hàng (line) thì xuống hàng mới. Những thẻ mặc định kiểu này như `` `<label>`
- **BLOCK** : phần tử được xác định là 1 khối (hình chữ nhật) và sẽ chiếm trọn 1 dòng. Tức là khi gặp phần tử kiểu BLOCK thì phải xuống dòng và kết thúc phần tử này cũng xuống dòng 1 lần nữa. Những thẻ có mặc định hiển thị kiểu này như: `<p>` `<div>`
- **INLINE-BLOCK** : thức. kiểu này chỉ định một phần tử sẽ có dạng khối nhưng không được ưu tiên xuống hàng, xuống 1 dòng khi xuất hiện và kết thúc.



4. Điều khiển hiển thị trong HTML It.kma

Thuộc tính Display

❖ Thuộc tính **DISPLAY** qui định hiển thị

- Dùng để chỉ định kiểu hiển thị của phần tử HTML này trên mặt phẳng nó đang thuộc về.
- Giá trị cơ bản bao gồm:
 - ✓ **inline** : tương ứng hiển thị kiểu inline
 - ✓ **block** : tương ứng hiển thị kiểu block
 - ✓ **inline-block** : tương ứng hiển thị kiểu kết hợp giữa inline và block
 - ✓ **none** : không hiển thị phần tử



4. Điều khiển hiển thị trong HTML It.kma

Thuộc tính Opacity

❖ Thuộc tính **OPACITY** qui định % hiển thị

- Thuộc tính này chỉ định độ trong suốt của phần tử HTML dựa trên tỷ lệ sẽ hiển thị so với ẩn đi. Tạo thành hiệu ứng trong suốt có thể nhìn thấu xuống các phần tử bị che bên dưới theo chiều z-index
- Giá trị từ 0 → 1 tương ứng với 0% và 100%. Ví dụ: opacity:0.5; nghĩa là chỉ hiển thị 50% và trong suốt 50%
- **Lưu ý:** khi opacity có giá trị 0 thì phần tử trong suốt 100% và không nhìn thấy nữa, nhưng hoàn toàn khác với không hiển thị. Ở đây chỉ có ý nghĩa là trong suốt 100% thôi chứ phần tử này vẫn ở đó và vẫn chiếm diện tích chỗ đó.

5. Sử dụng background

- ❖ **background-color**
- ❖ **background-image**
- ❖ **background-repeat**
- ❖ **background-attachment**
- ❖ **background-position**
- ❖ **CSS3**
 - background-size
 - background-origin
 - background-clip



5. Sử dụng background (tiếp)

❖ Thuộc tính **background-color**

- Chỉ định màu nền cho thẻ html được chọn
- Giá trị màu được sử dụng theo RGB, thập lục phân hoặc bằng tên tiếng anh

5. Sử dụng background (tiếp)

❖ Thuộc tính **background-image**

- Chỉ định ảnh nền cho thẻ html được chọn
- Giá trị có cú pháp như sau: **url('đường dẫn hình ảnh')**
- Đường dẫn đến ảnh có thể là tương đối hoặc tuyệt đối (nếu viết trong file .css thì đường dẫn tương đối được tính từ file .css đó)

5. Sử dụng background (tiếp)

❖ Thuộc tính **background-repeat**

- Sử dụng kèm với background-image. Dùng để chỉ định cách ảnh nền lặp lại chính nó như thế nào? Giá trị bao gồm:
 - No-repeat: không lặp lại
 - Repeat-x: ảnh nền lặp lại theo chiều ngang
 - Repeat-y: ảnh nền lặp lại theo chiều dọc
 - Repeat: ảnh nền lặp lại theo cả hai chiều (mặc định)
 - Inherit: kế thừa lại từ thẻ HTML cha của nó

5. Sử dụng background (tiếp)

❖ Thuộc tính **background-attachment**

- Sử dụng kèm với background-image. Dùng để chỉ định cách ảnh nền hiển thị cố định hay scroll theo nội dung? Giá trị bao gồm:
 - scroll : ảnh nền sẽ scroll theo thẻ html (thường là mặc định)
 - fixed : ảnh nền bị cố định và không scroll theo nội dung
 - local : ảnh nền sẽ scroll theo nội dung
 - inherit : kế thừa lại từ HTML cha của thẻ HTML

5. Sử dụng background (tiếp)

❖ Thuộc tính **background-position**

- Sử dụng kèm với background-image. Dùng để chỉ định vị trí ảnh nền được hiển thị như thế nào trong thẻ HTML? Giá trị bao gồm một cặp dữ liệu thể hiện tọa độ X và Y
 - px px : ảnh nền sẽ hiển thị từ vị trí cách biên trái Xpx, cách biên trên Ypx
 - key key : Với key tương ứng → top, right, bottom, left, center

5. Sử dụng background (tiếp)

❖ CSS3

- background-size
- background-origin
- background-clip



5. Sử dụng background (tiếp)

❖ Thuộc tính **background-size**

- Thuộc tính này có trong CSS3 và được sử dụng cùng với background-image có tác dụng chỉ định kích thước ảnh nền, giá trị bao gồm:
 - width height : Cặp số chỉ định chiều rộng và cao của ảnh nền (px)
 - contain : càng lớn càng tốt và phải đảm bảo nhìn đủ ảnh
 - cover : càng lớn càng tốt và phải đảm bảo bao hết nội dung



5. Sử dụng background (tiếp)

❖ Thuộc tính **background-origin** (hình nền)

- Hoạt động gần giống như background-position (vị trí bắt đầu có ảnh nền) nhưng chỉ có 3 trường hợp giá trị đặc biệt chứ không được sử dụng tọa độ cụ thể:
 - border-box : Ảnh nền được tính từ đường viền (border)
 - padding-box : Ảnh nền được tính từ khoảng cách biên (mặc định)
 - content-box : Ảnh nền được tính từ chỗ chứa nội dung



5. Sử dụng background (tiếp)

❖ Thuộc tính **background-clip** (màu nền)

- Hoạt động gần giống như background-origin nhưng tính cả màu nền và cũng chỉ có 3 trường hợp giá trị đặc biệt chứ không được sử dụng tọa độ cụ thể:
 - border-box : Nền được tính từ đường viền (border) (mặc định)
 - padding-box : Nền được tính từ khoảng cách biên
 - content-box : Ảnh nền được tính từ chỗ chứa nội dung

5. Sử dụng background (tiếp)

❖ Nhiều thuộc tính background viết cùng một hàng (shorthand)

```
background: #666 url('actvn.jpg') left top no-repeat;
```

❖ Nhiều ảnh nền chồng lên nhau:

- Có thể khai báo sử dụng cùng lúc nhiều bộ shorthand và kết hợp nhịp nhàng với thuộc tính tạo vị trí sẽ có tác dụng hiển thị nhiều ảnh nền cho cùng một phần tử HTML. Mỗi bộ giá trị chỉ cần cách nhau dấu phẩy “,” như ví dụ:

```
background: url('bg1.jpg'), url('bg2.jpg'), url('bg3.jpg');  
background-position: left top, right top, center bottom;
```

```
background: url('bg1.jpg') left top, url('bg2.jpg') right top;
```



3. Các thuộc tính trong CSS hay dùng

It.kma

background: Định dạng nền cho thành phần

Thuộc tính	giá trị	Ví dụ	Mô tả
background-attachment	fixed scroll	background-attachment :fixed;	Xác định thành phần nền được cố định hoặc cuộn so với trang. Được sử dụng kèm với giá trị background-image
background-color	mã màu tên màu Giá trị rgb	background-color :#ff0000; background-color: red; background-color: rgb(255,0,0);	Xác định màu nền cho thành phần.
background-image	url(đường dẫn hình)	background-image: url(ico_arrow.gif);	Xác định hình ảnh nền cho thành phần.
background-position	left right top bottom px %	background-position: left top;	Xác định vị trí hình ảnh nền cho thành phần. Được sử dụng kèm với giá trị background-image.
background-repeat	repeat-x repeat-y repeat no-repeat	background-repeat: repeat-x;	Xác định hình ảnh nền được lặp như thế nào.
background	Một hoặc nhiều giá trị của các thuộc tính trên	background: url(ico_arrow.gif) repeat-x left top;	Có giá trị riêng lẻ hoặc tổng hợp các thuộc tính trên.



6. Các thuộc tính trong CSS hay dùng It.kma

Thuộc tính cho văn bản (text)

- ❖ **Color**: Định màu cho văn bản.
- ❖ **Text-align**: căn lề cho văn bản
- ❖ **Text-decoration**: Trang trí văn bản
- ❖ **Text-indent**: Thêm khoảng trống trước văn bản theo chiều ngang từ trái qua phải
- ❖ **Text-shadow**: Thêm độ bóng cho văn bản
- ❖ **Text-transform**: Tùy chỉnh việc hiển thị chữ in hoa
- ❖ **font-family**: Chọn kiểu chữ cho văn bản
- ❖ **font-style, font-size**:
- ❖ **Letter-spacing**: khoảng cách giữa 2 chữ cái trong từ
- ❖ **Line-height**: độ cao của 1 line trong đoạn.
- ❖ **Word-spacing**: định khoảng cách giữa hai từ trong đoạn



6. Các thuộc tính trong CSS hay dùng It.kma Thuộc tính cho văn bản (text)

- ❖ **Direction** và **Unicode-bidi**: hướng và cách viết văn bản
- ❖ **Vertical-align**: Cách căn văn bản theo chiều dọc trong cùng 1 line
- ❖ **Text-overflow**: Quy định cách xử lý văn bản khi bị tràn khỏi khung chứa.
- ❖ **User-select**: có cho phép người dùng chọn văn bản.
- ❖ **Visibility**: có cho phép hiển thị văn bản hay không?
- ❖ **Writing-mode**: định hướng viết văn bản





6. Các thuộc tính trong CSS hay dùng It.kma Thuộc tính cho văn bản (tiếp)

- ❖ **Color**: Định màu cho văn bản được chọn.
 - Màu bằng mã hex, RGB, tiếng anh
- ❖ **Text-align**: căn lề cho văn bản, gồm giá trị:
 - Text-align : left; //căn lề từ bên trái
 - Text-align : right; //căn lề từ bên phải
 - Text-align : center; //căn lề từ chính giữa
 - Text-align : justify; //căn lề đều hai bên





6. Các thuộc tính trong CSS hay dùng

Thuộc tính cho văn bản (tiếp)

It.kma

```
<style>
h1 {
  text-align: center;
}
h2 {
  text-align: left;
}
h3 {
  text-align: right;
}
div {
  border: 1px solid black;
  padding: 10px;
  width: 200px;
  height: 200px;
  text-align: justify;
  color: red;
}
</style>
</head>
<body>
<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>
<div>
```

Heading 1 (center)

Heading 2 (left)

Heading 3 (right)

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'



6. Các thuộc tính trong CSS hay dùng

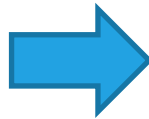
Thuộc tính cho văn bản (tiếp)

❖ Text-decoration: trang trí văn bản:

- Text-decoration: underline; //gạch trên chữ
- Text-decoration : underline; //gạch dưới chữ
- Text-decoration : line-through; //gạch ngang chữ
- Text-decoration : none; //không gạch gì

```
<style>
h1 {
  text-decoration: underline;
}
h2 {
  text-decoration: line-through;
}
h3 {
  text-decoration: underline;
}
h3 {
  text-decoration: none;
}
</style>
</head>
<body>
```

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
```



This is heading 1

~~This is heading 2~~

This is heading 3

This is heading 4



The text-indent Property

text-indent: 50px:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

text-indent: -2em:

em ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

text-indent: 30%:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

```
<style>
div.a {
  text-indent: 50px;
}
div.b {
  text-indent: -2em;
}
div.c {
  text-indent: 30%;
}
</style>
</head>
<body>
```

```
<h1>The text-indent Property</h1>
```

```
<h2>text-indent: 50px:</h2>
```

```
<div class="a">
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
```

```
</div>
```

```
<h2>text-indent: -2em:</h2>
```

```
<div class="b">
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
```

```
</div>
```

```
<h2>text-indent: 30%:</h2>
```

```
<div class="c">
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.</p>
```

```
</div>
```

❖ Text-indent:

thêm khoảng
trống trước văn
bản:

- Text-indent: 15px
(giá trị là đơn vị đo lường)



6. Các thuộc tính trong CSS hay dùng It.kma Thuộc tính cho văn bản (tiếp)

❖ Text-shadow: tạo độ bóng cho văn bản

■ Text-shadow: [tọa độ x] [y] [độ mờ] [màu sắc] ;

```
<style>
h1 {
  text-shadow: 2px 2px 8px #FF0000;
}
</style>
</head>
<body>
```

```
<h1>Text-shadow with blur effect</h1>
```

```
<p><b>Note:</b> Internet Explorer 9 and earlier do not support the text-shadow property.</p>
```

```
</body>
</html>
```



Text-shadow with blur effect

```
h1 {
  text-shadow: 2px 2px #FF0000;
}
```



Text-shadow with blur effect

Note: Internet Explorer 9 and earlier do not support the text-shadow property.



6. Các thuộc tính trong CSS hay dùng

Thuộc tính cho văn bản (tiếp)



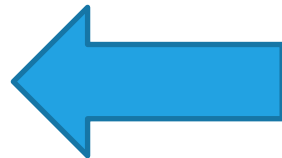
❖ Text-transform: Tùy chỉnh văn bản in hoa hay thường

- Capitalize: chữ hoa ở đầu tiên mỗi từ
- Uppercase: chữ hoa toàn văn bản
- Lowercase: chữ thường
- None: không làm gì cả

THIS IS SOME TEXT.

this is some text.

This Is Some Text.



```
<style>
p.uppercase {
  text-transform: uppercase;
}

p.lowercase {
  text-transform: lowercase;
}

p.capitalize {
  text-transform: capitalize;
}
</style>
</head>
<body>
```

```
<p class="uppercase">This is some text.</p>
<p class="lowercase">This is some text.</p>
<p class="capitalize">This is some text.</p>
```





6. Các thuộc tính trong CSS hay dùng

Trang trí font chữ cho văn bản

It.kma

❖ **font-family**: Chọn kiểu chữ cho văn bản

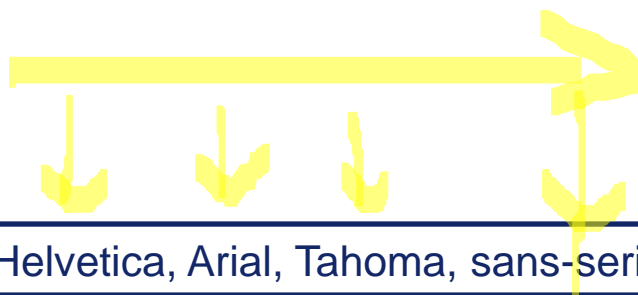
- Cú pháp: font-family: tên-font, tên-font-backup,...;
- Hai giá trị font chữ: serif (font có chân), sans-serif (font không có chân) – dựa theo font cơ bản trên MT.

■ Serif

- Palatino
- Time new roman
- Georgia

■ Sans Serif

- Arial
- Helvetica
- Verdana
- Tohoma



font-family: Helvetica, Arial, Tahoma, sans-serif;





6. Các thuộc tính trong CSS hay dùng

It.kma

Trang trí font chữ cho văn bản

❖ **font-style**: Thiết lập in nghiêng, bình thường

- Normal: bình thường
- Italic: In nghiêng
- Oblique: in nghiêng

`font-style: italic;`

Font

Font

❖ **Font-weight**: In đậm cho chữ

- Font-weight: giá trị số [hoặc normal or bold];
- Nếu phần tử cha đã được thiết lập thì ta có thể dùng giá trị lighter và bolder để thiết lập tương đối.

`font-weight: 300;`
`Font-weight: bold;`

❖ **Color**: thiết lập màu chữ

❖ **Font-size**: Thiết lập cỡ chữ

- **Font-size**: giá trị + đơn vị đo lường

`color: #fff;`
`Font-size: bold;`





6. Các thuộc tính trong CSS hay dùng

It.kma

Trang trí font chữ cho văn bản

❖ **Letter-spacing**: Thiết lập khoảng cách giữa hai chữ cái trong từ

```
<style>
h1 {
  letter-spacing: 3px;
}
```

```
h2 {
  letter-spacing: 2px;
}
```

```
h3 {
  letter-spacing: -1px;
}
```

```
</style>
</head>
<body>
```

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

This is heading 1

This is heading 2

This is heading 3



6. Các thuộc tính trong CSS hay dùng

Trang trí font chữ cho văn bản

It.kma

❖ Line-height: độ cao của 1 line trong đoạn.

```
<style>
div.a {
  line-height: normal;
}

div.b {
  line-height: 1.6;
}

div.c {
  line-height: 80%;
}
div.d {
  line-height: 200%;
}
</style>
</head>
<body>
<h1>The line-height Property</h1>
<h2>line-height: normal (default):</h2>
<div class="a">This is a paragraph with a standard line-height.<br>
The standard line height in most browsers is about 110% to 120%.</div>

<h2>line-height: 1.6 (recommended):</h2>
<div class="b">This is a paragraph with the recommended line-height.<br>
The line height is here set to 1.6. This is a unitless value;<br>
meaning that the line height will be relative to the font size.</div>

<h2>line-height: 80%:</h2>
<div class="c">This is a paragraph with a smaller line-height.<br>
The line height is here set to 80%.</div>

<h2>line-height: 200%:</h2>
<div class="d">This is a paragraph with a bigger line-height.<br>
The line height is here set to 200%.</div>
```

The line-height Property

line-height: normal (default):

This is a paragraph with a standard line-height.
The standard line height in most browsers is about 110% to 120%.

line-height: 1.6 (recommended):

This is a paragraph with the recommended line-height.
The line height is here set to 1.6. This is a unitless value;
meaning that the line height will be relative to the font size.

line-height: 80%:

This is a paragraph with a smaller line-height.
The line height is here set to 80%.

line-height: 200%:

This is a paragraph with a bigger line-height.
The line height is here set to 200%.





6. Các thuộc tính trong CSS hay dùng It.kma

Trang trí font chữ cho văn bản

❖ **Word-spacing**: định khoảng cách giữa hai từ trong đoạn

```
<style>
p.a {
  word-spacing: normal;
}
p.b {
  word-spacing: 30px;
}
p.c {
  word-spacing: 1cm;
}
</style>
</head>
<body>

<h1>The word-spacing Property</h1>

<h2>word-spacing: normal:</h2>
<p class="a">This is some text. This is some text.</p>

<h2>word-spacing: 30px:</h2>
<p class="b">This is some text. This is some text.</p>

<h2>word-spacing: 1cm:</h2>
<p class="c">This is some text. This is some text.</p>
```

The word-spacing Property

word-spacing: normal:

This is some text. This is some text.

word-spacing: 30px:

This is some text. This is some text.

word-spacing: 1cm:

This is some text. This is some text.



6. Các thuộc tính trong CSS hay dùng

It.kma

Trang trí font chữ cho văn bản

❖ **Direction** và **Unicode-bidi**: hướng và cách viết văn bản

- **direction**: chỉ định hướng ghi văn bản, từ **trái sang phải (ltr)** hoặc ngược lại (**rtl**)
- **unicode-bidi**: dùng kết hợp với direction để phục vụ cho một số cách viết ngược trong một số ngôn ngữ. Có 2 giá trị quan trọng là **normal** (bình thường) và **bidirectional-override** (viết ngược).

```
<style>  
p.rtl {  
  direction: rtl;  
}
```

The direction Property

This text goes from left to right. This is default.

```
div.ex1 {  
  direction: rtl;  
  unicode-bidi: bidi-override;  
}  
</style>  
</head>  
<body>
```

.This text goes from right to left

.noitcerid tfel-ot-thgiR .txet emoS

```
<h1>The direction Property</h1>
```

```
<p>This text goes from left to right. This is default.</p>
```

```
<p class="rtl">This text goes from right to left.</p>
```

```
<div class="ex1">Some text. Right-to-left direction.</div>
```



6. Các thuộc tính trong CSS hay dùng

Trang trí font chữ cho văn bản

It.kma

❖ Vertical-align: Cách căn văn bản theo chiều dọc trong cùng 1 line

The vertical Property

vertical-align: baseline (default):

An image with a default alignment.

vertical-align: text-top:

An image with a text-top alignment.

vertical-align: text-bottom:

An image with a text-bottom alignment.

vertical-align: sub:

An image with a sub alignment.

vertical-align: sup:

An image with a super alignment.

<The vertical Property>

<vertical-align: baseline (default):>

<An image with a default alignment.>

<vertical-align: text-top:>

<An image with a text-top alignment.>

<vertical-align: text-bottom:>

<An image with a text-bottom alignment.>

<vertical-align: sub:>

<An image with a sub alignment.>

<vertical-align: sup:>

<An image with a super alignment.>

```
<style>
img.a {
  vertical-align: baseline;
}

img.b {
  vertical-align: text-top;
}

img.c {
  vertical-align: text-bottom;
}

img.d {
  vertical-align: sub;
}

img.e {
  vertical-align: super;
}
```





6. Các thuộc tính trong CSS hay dùng

It.kma

Trang trí font chữ cho văn bản

- ❖ **White-space**: Xác định khoảng trắng có bên trong thành phần được xử lý như thế nào.

```
tag {  
  white-space: giá trị;  
}
```

Thuộc tính	giá trị	Ví dụ	Mô tả
white-space	normal	white-space: normal;	Khoảng trắng sẽ thể hiện bình thường, đây là dạng mặc định.
	nowrap	white-space: nowrap;	Văn bản sẽ hiển thị trên cùng một hàng, chỉ xuống hàng khi gặp thẻ <code>
</code> .
	pre	white-space: pre;	Khoảng trắng sẽ do trình duyệt điều khiển, văn bản sẽ hiển thị trên cùng một hàng, chỉ ngắt dòng tại đoạn văn bản sử dụng thẻ <code><pre></code> .
	pre-line	white-space: pre-line;	Văn bản sẽ tự động bao lại khi cần thiết, và xuống hàng.
	pre-wrap	white-space: pre-wrap;	Khoảng trắng sẽ do trình duyệt điều khiển, văn bản sẽ tự động bao lại khi cần thiết, và xuống hàng.
	inherit	white-space: inherit;	Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài).



6. Các thuộc tính trong CSS hay dùng

Trang trí font chữ cho văn bản

It.kma

❖ **Text-overflow**: Quy định cách xử lý văn bản khi bị tràn khỏi khung chứa.

❖ Có 1 giá trị quan trọng:

✓ **ellipsis**: tạo ra đoạn cuối văn bản dấu ... trước phần văn bản bị mất

```
p {  
  border: 1px solid #cc0000;  
  overflow: hidden;  
  white-space: nowrap;  
  width: 200px;  
}
```

→ Đây là đoạn text có sử dụng te

```
p {  
  border: 1px solid #cc0000;  
  overflow: hidden;  
  text-overflow: clip;  
  white-space: nowrap;  
  width: 200px;  
}
```

← Đây là đoạn text có sử dụng...

```
p {  
  border: 1px solid #cc0000;  
  overflow: hidden;  
  text-overflow: ellipsis;  
  white-space: nowrap;  
  width: 200px;  
}
```

```
p:hover {  
  overflow: visible;  
  white-space: pre-line;  
  width: 500px;  
}
```

```
<html>  
<head></head>  
<body>
```

```
<p>Đây là đoạn text có sử dụng text overflow đây là đoạn text có sử dụng text overflow đây là đoạn text có sử dụng text overflow đây là đoạn text có sử dụng text overflow đây là đoạn text có sử dụng text overflow đây là đoạn text có sử dụng text overflow</p>
```

```
</body>  
</html>
```

Nếu ellipsis kết hợp với hover thì khi di chuột vào sẽ thấy toàn bộ nội dung



6. Các thuộc tính trong CSS hay dùng It.kma Trang trí font chữ cho văn bản

❖ **User-select:** có cho phép người dùng chọn văn bản.

- **Auto:** mặc định, text có thể được chọn nếu trình duyệt cho phép
- **None:** Không cho phép chọn text
- **Text:** text có thể được chọn bởi user
- **All:** text được chọn bằng một lần click thay vì double-click

`user-select: auto|none|text|all;`



6. Các thuộc tính trong CSS hay dùng

Trang trí font chữ cho văn bản

It.kma

❖ **Visibility**: có cho phép hiển thị văn bản hay không?

Thuộc tính	giá trị	Ví dụ	Mô tả
visibility	collapse	visibility: collapse;	Làm "sụp đổ" các hàng và cột của thành phần table , giá trị này chỉ sử dụng cho thành phần table.
	hidden	visibility: hidden;	Thành phần sẽ không được nhìn thấy.
	visible	visibility: visible;	Hiển thị thành phần, đây là dạng mặc định.
	inherit	visibility: inherit;	Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài).

```
<html>
<head></head>
<body>
<p>HỌC WEB CHUẨN</p>
<p class="visibility">Toán Học</p>
<p>HTML</p>
<p>CSS</p>
</body>
</html>
```



HỌC WEB CHUẨN

Toán Học

HTML

CSS





6. Các thuộc tính trong CSS hay dùng

Trang trí cho văn bản

It.kma

❖ **Writing-mode:** định hướng viết văn bản

The writing-mode Property

```
<style>
p.test1 {
  writing-mode: horizontal-tb;
}
```

```
p.test2 {
  writing-mode: vertical-rl;
}
```

```
span.test2 {
  writing-mode: vertical-rl;
}
```

```
</style>
</head>
<body>
```

```
<h1>The writing-mode Property</h1>
```

```
<p class="test1">Some text with default writing-mode.</p>
```

```
<p>Some text with a span element with a <span class="test2">vertical-rl</span> writing-mode.</p>
```

```
<p class="test2">Some text with writing-mode: vertical-rl.</p>
```

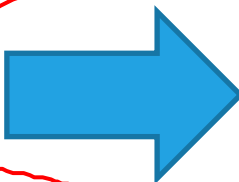
```
</body>
```

Some text with default writing-mode.

Some text with a span element with a writing-mode.

Some text with writing-mode: vertical-rl.

vertical-rl





6. Các thuộc tính trong CSS hay dùng

Trang trí cho văn bản

❖ Kết thúc các thuộc tính trang trí cho văn bản



7. Một số thuộc tính và selector hay sử dụng khác

- ❖ Thuộc tính cố định box-model
- ❖ Tùy biến hiển thị danh sách (list)
- ❖ Selector với trạng thái (thẻ a)
- ❖ Selector theo quan hệ cha con
- ❖ Selector với thẻ input trong form.

7. Thuộc tính cố định box-model

- ❖ Length: px, pt, em ,rem
- ❖ Thiết lập cho các phần tử bock.

Tên thuộc tính	Mô tả	Các kiểu giá trị
Height	Thiết lập chiều cao cho một phần tử	Auto, length, %, inherit
Max-height	Thiết lập chiều cao tối đa của một phần tử	None, length, %, inherit
Max-width	Thiết lập chiều rộng tối đa của một phần tử	None, length, %, inherit
Min-width	Thiết lập chiều rộng tối thiểu cho một phần tử	length, %, inherit
Min-height	Thiết lập chiều cao tối thiểu cho 1 phần tử	length, %, inherit
Width	Thiết lập chiều rộng cho một phần tử	Auto, length, %, inherit

7. Thuộc tính cố định box-model

❖ Sử dụng Box Model:

- Khi box có thuộc tính padding và border
 - Kích thước box biến đổi nếu có thêm width và height
 - Ví dụ: box = 500 x 500 (w x h), padding:15px => box: 530x530 (tương tự với border)
- Sử dụng **box-sizing** nếu muốn box giữ nguyên kích thước (dù có thêm padding và border)
 - box-sizing: border-box;
 - -moz-box-sizing: border-box;
 - -webkit-box-sizing: border-box;
- Giá trị trong box-sizing:
 - **Content-box**: mặc định, W + H chỉ áp dụng cho nội dung bên trong (không bao gồm padding, border, margin)
 - **Border-box**: bao gồm padding, border không margin
 - **Padding-box**: bao gồm padding không border, margin (firefox).



Tùy biến hiển thị danh sách (list)

- ❖ Trong HTML dùng thẻ ol, ul để định danh sách
- ❖ Trong CSS có thể tùy biến li theo sở thích
 - Sử dụng thuộc tính list-style
 - Cú pháp: list-style: <type> <position> <image>;
 - List-style: none /*Bỏ type */
 - **List-style-type**: thay đổi loại hiển thị danh sách
 - Disc, circle, square, decimal (số thứ tự), lower-roman (la mã thường), upper-roman, none.
 - **List-style-Position**: Tùy chỉnh vị trí hiển thị các dấu nằm trong hay ngoài danh sách.
 - **Inside**: Hiển thị dấu đầu dòng bên trong block
 - **Outside**: Hiển thị dấu đầu dòng bên ngoài block
 - **List-style-Image**: Sử dụng ảnh để làm các dấu đầu dòng cho danh sách.
 - url('link ảnh');
- ❖ Sử dụng thuộc tính **display: list-item**: chuyển các phần tử về dạng

7. Selector với trạng thái

❖ Các trạng thái với thẻ <a>

- **a:link{}** → chọn tất cả thẻ a chưa từng bị tác động đến (click)
- **a:active{}** → chọn tất cả thẻ a đã bị click và chuẩn bị chuyển trang
- **a:visited{}** → chọn tất cả thẻ a đã từng bị click và chuyển trang

❖ **div:hover** → css có tác dụng khi rê chuột lên đối tượng thẻ div

- Ví dụ: *#banner:hover{opacity:0.5}*
- Ý nghĩa: Làm mờ 50% đối tượng có id là banner khi con trỏ chuột rê lên nó.

7. Selector chọn theo quan hệ cha con

- ❖ **p:first-child** → chọn mọi thẻ <p> mà bản thân nó là thẻ con đầu tiên của thẻ cha của nó.
- ❖ **p:last-child** → chọn mọi thẻ <p> mà bản thân nó là thẻ con cuối cùng của thẻ cha của nó.
- ❖ **p:first-of-type** → chọn mọi thẻ <p> mà bản thân nó là thẻ <p> con đầu tiên của thẻ cha của nó.
- ❖ **p:last-of-type** → chọn mọi thẻ <p> mà bản thân nó là thẻ <p> con cuối cùng của thẻ cha của nó.
 - Child và of-type: child thẻ đó phải là thẻ con đầu tiên nghĩa là bên trên nó không có thẻ nào khác (của một thẻ cha nào đó). Of-type: trên nó có thể có nhiều thẻ khác nhưng nó là đầu tiên của loại thẻ đó.
 - Ví dụ: trong <body> có nhiều thẻ <p>. **Child** đó là thẻ <p> nằm ngay cạnh <body>, trên nó không có bất cứ 1 thẻ nào khác (kể cả khác loại). Còn **of-type**: nó có thể có các thẻ div, a,.. Khác nhưng nó là thẻ <p> đầu tiên.

7. Selector chọn theo quan hệ cha con

- ❖ **p:nth-child(3)** → chọn mọi thẻ <p> mà bản thân nó là thẻ con thứ 3 của thẻ cha của nó.
- ❖ **p:nth-last-child(2)** → chọn mọi thẻ <p> mà bản thân nó là thẻ con thứ 2 (từ dưới lên) của thẻ cha của nó.
- ❖ **p:nth-of-type(4)** → chọn mọi thẻ <p> mà bản thân nó là thẻ <p> con thứ 4 của thẻ cha của nó.
- ❖ **p:nth-last-of-type(3)** → chọn mọi thẻ <p> mà bản thân nó là thẻ <p> con thứ 3 (từ dưới lên) của thẻ cha của nó.



7. Selector với thẻ input trong form

- ❖ **input:checked** → chọn mọi thẻ `<input>` đã được chọn (radio hoặc checkbox)
- ❖ **input:disabled** → chọn mọi thẻ `<input>` bất khả dụng
- ❖ **input:enabled** → chọn mọi thẻ `<input>` khả dụng
- ❖ **input:valid** → chọn mọi thẻ `<input>` có giá trị đúng định dạng (GT?)
- ❖ **input:invalid** → chọn mọi thẻ `<input>` có giá trị sai định dạng (GT?)
- ❖ **input:in-range** → chọn mọi thẻ `<input>` có giá trị nằm trong vùng giới hạn (GT?)
- ❖ **input:out-of-range** → chọn mọi thẻ `<input>` có giá trị ngoài vùng giới hạn (GT?)



7. Selector với thẻ input trong form

- ❖ **input:read-only** → chọn mọi thẻ `<input>` không được thay đổi giá trị
- ❖ **input:read-write** → chọn mọi thẻ `<input>` được phép thay đổi giá trị
- ❖ **input:required** → chọn mọi thẻ `<input>` không được phép bỏ trống giá trị
- ❖ **input:optional** → chọn mọi thẻ `<input>` có thể bỏ trống giá trị



7. Selector tạo nội dung ngoài luồng

- ❖ **p::after{content:"nội dung"[,css]}** → thêm "nội dung" vào sau mọi thẻ p. Kèm theo định dạng css cho nội dung ngoài luồng (nếu có)
- ❖ **p::before{content:"nội dung"[,css]}** → thêm "nội dung" vào trước mọi thẻ p. Kèm theo định dạng css cho nội dung ngoài luồng (nếu có)



8. Sắp xếp vị trí các khối

- ❖ Sử dụng thuộc tính float và clear float để tạo cột
- ❖ Sử dụng flexbox
- ❖ Sử dụng grid-layout



8. Sắp xếp vị trí các khối

Thuộc tính float và clear float

- ❖ Bước 1: Các cột luôn có 1 container (phần tử mẹ bao bọc)
- ❖ B2: Thiết lập chiều rộng của container (width)
- ❖ B3: Thiết lập chiều rộng cho 2 hoặc nhiều cột (tổng bằng hoặc ít hơn container)
- ❖ B4: Sử dụng box-sizing: border-box để tính toán kích thước chính xác
- ❖ B5: Sử dụng thuộc tính float với giá trị left và right để đẩy phần tử sang trái hoặc phải
- ❖ B6: Tiến hành clear float.



8. Sắp xếp vị trí các khối Thuộc tính float và clear float

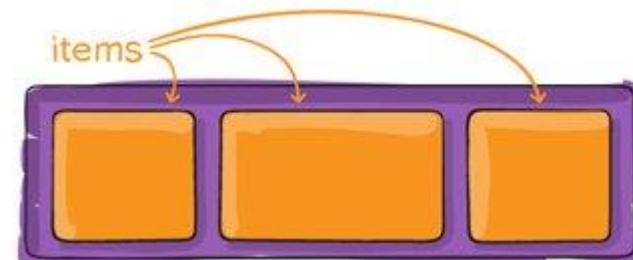
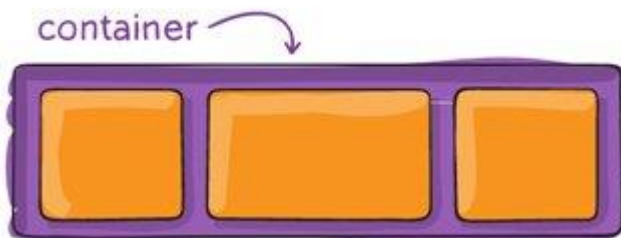
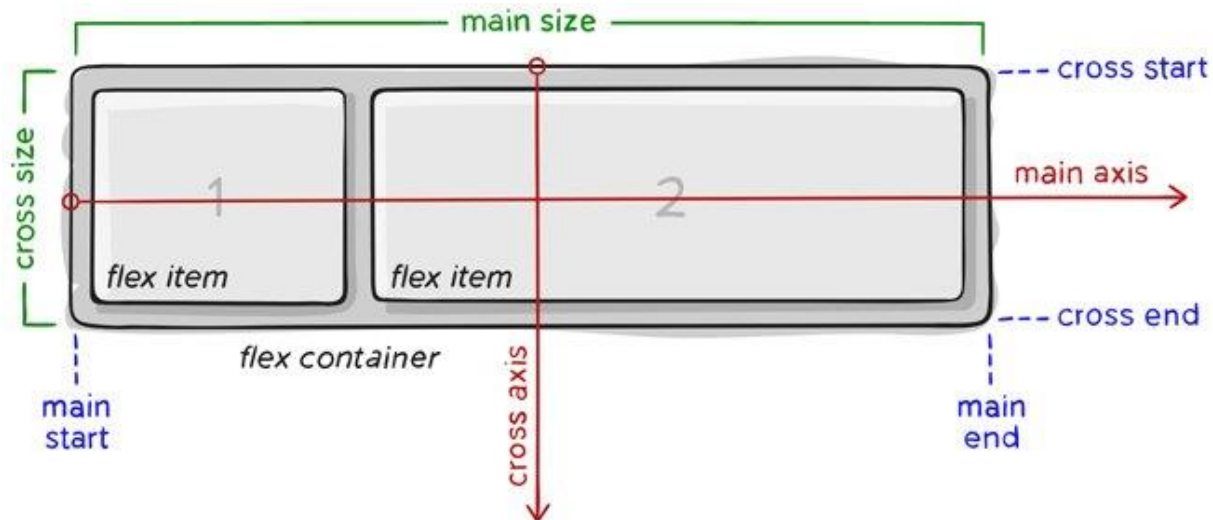
❖ B6: Tiến hành clear float.

- C1: sử dụng thêm 1 thẻ div trong container đặt class = “clear”
 - Trong CSS viết thêm .clear{clear: both}
- C2: thêm thuộc tính overflow:auto cho container chứa là được.



8. Sắp xếp vị trí các khối Flexbox

❖ **Flexbox Layout (hay còn gọi là Flexible Box)** là một kiểu bố cục trang có khả năng tự cân đối kích thước, thay đổi chiều rộng/chiều cao và thứ tự phần tử bên trong để phù hợp với tất cả các loại thiết bị hiển thị và kích thước màn hình





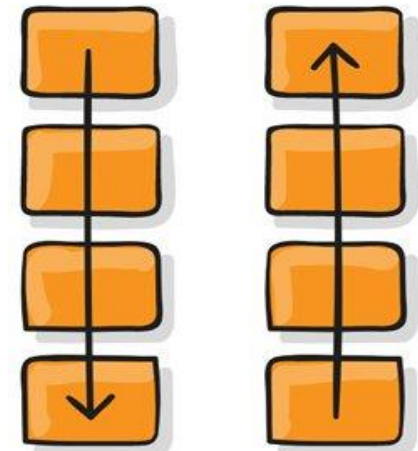
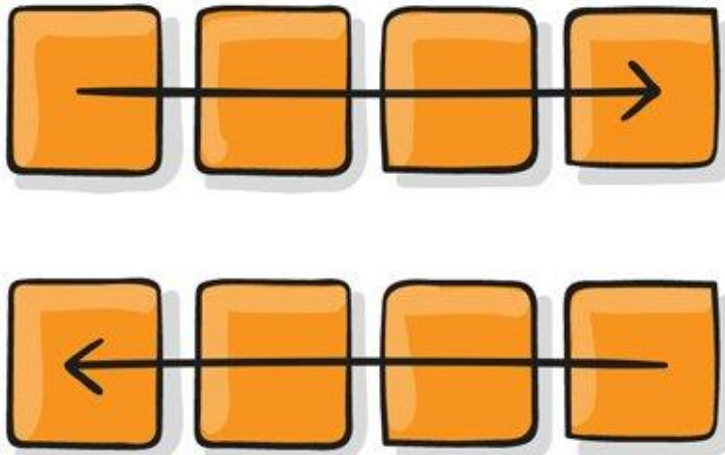
8. Sắp xếp vị trí các khối Flexbox -container (tiếp)

❖ **Sử dụng flexbox:** Khai báo thuộc tính **display** trong thẻ chứa.

```
.container {  
  display: flex; /* hoặc inline-flex */  
}
```

❖ Thuộc tính **flex-direction**:

```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```



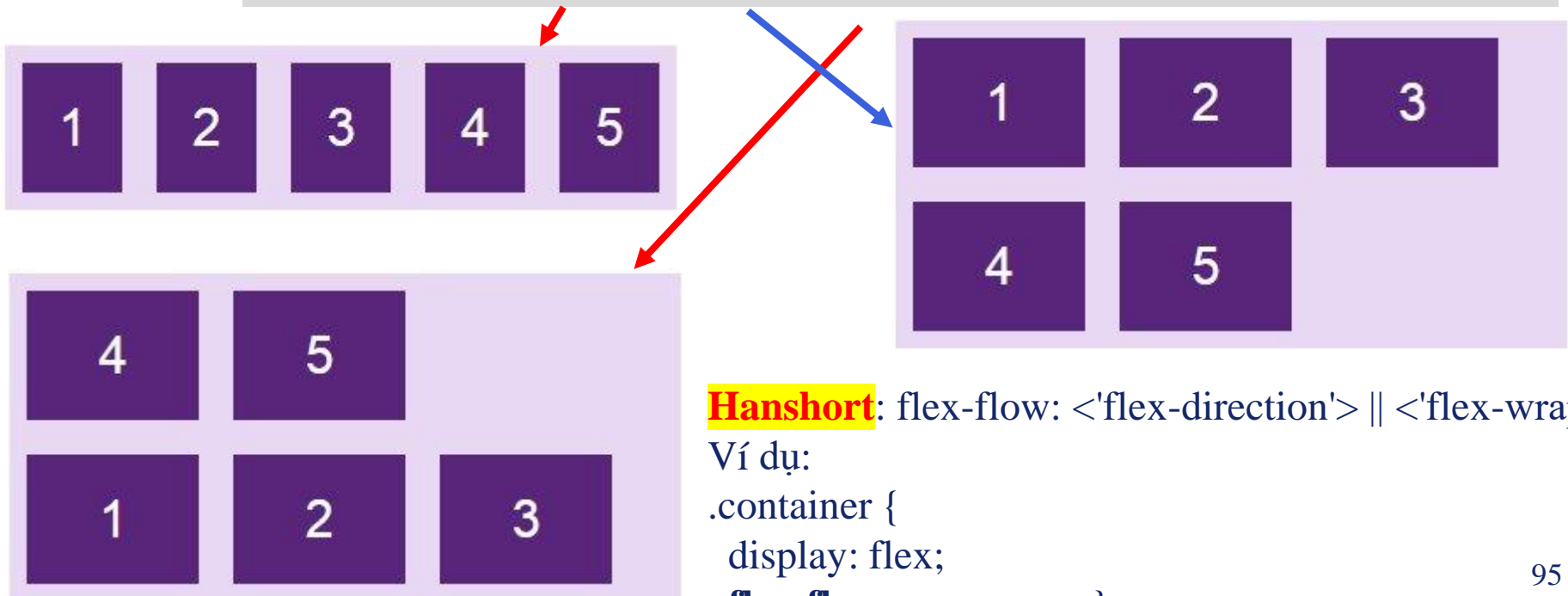


8. Sắp xếp vị trí các khối Flexbox -container (tiếp)

❖ **Thuộc tính flex-wrap:** mặc định item tự thay đổi kích thước để luôn hiển thị trên 1 dòng => giãn nhỏ sẽ xấu

- Flex-wrap: item tự động xuống dòng khi kích thước container thay đổi.

```
.container{  
  flex-wrap: nowrap | wrap | wrap-reverse;}
```



Hanshort: flex-flow: <'flex-direction'> || <'flex-wrap'>

Ví dụ:

```
.container {  
  display: flex;  
  flex-flow: row wrap;}
```



8. Sắp xếp vị trí các khối Flexbox -container (tiếp)

❖ **Thuộc tính justify-content:** mặc định các item bắt đầu từ trái sang phải.

```
.container{  
  justify-content: flex-start | flex-end | center | space-between |  
                 space-around | space-evenly;  
}
```

flex-start



center



flex-end



space-between



space-around



space-evenly

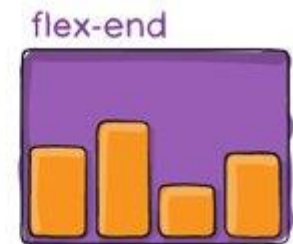
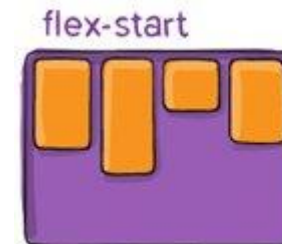
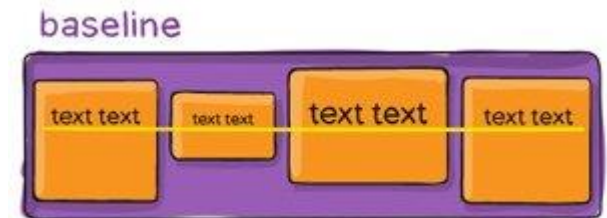
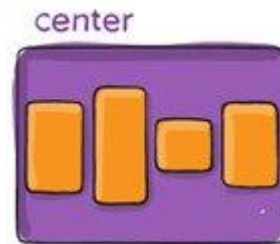




8. Sắp xếp vị trí các khối Flexbox -container(tiếp)

- ❖ Thuộc tính **align-items**: Điều chỉnh vị trí bắt đầu và căn chỉnh các item bên trong container dọc theo trục cross axis, chiều ngang hoặc chiều dọc tùy thuộc vào flex-direction.

```
.container{  
  align-items: stretch | flex-start | flex-end | center | baseline;  
}
```





8. Sắp xếp vị trí các khối Flexbox -container(tiếp)

❖ Thuộc tính **align-content**: căn hình khoảng cách các item bên trong container dọc theo cross axis, chiều ngang tùy vào flex-direction:

- **flex-start**: item sẽ bắt đầu từ lề chứa cross-start của container.
- **flex-end**: item sẽ bắt đầu từ lề chứa cross-end của container.
- **center**: item sẽ nằm giữa container căn theo cross-axis.
- **space-between**: các item sẽ có khoảng cách giữa các phần tử bằng nhau do container tự động căn khoảng cách, item đầu tiên sát lề chứa cross-start, item cuối cùng sát lề chứa cross-end.
- **space-around**: tương tự space-between, nhưng khác ở chỗ là mỗi item có khoảng cách 2 bên cạnh và những khoảng cách này bằng nhau.
- **stretch**: giá trị mặc định, các phần tử sẽ được kéo dài, căn chỉnh sao cho lấp đầy container chứa nó (vẫn ưu tiên giá trị height/width nếu có).

```
.container{  
  align-content: flex-start | flex-end | center | space-between | space-around | stretch;  
}
```



8. Sắp xếp vị trí các khối Flexbox - Item (tiếp)

❖ Các thuộc tính của flex-Item

- Order: sắp xếp lại vị trí của các item

```
.item {  
  order: <integer>; /* mặc định là 0 */  
}
```



- Flex-grow: cho phép các phần tử giãn theo độ rộng của container

```
.item {  
  flex-grow: <number>; /* mặc định là 0 */  
}
```

flex-grow: 1; width: 100px



width: 100px



#all-item: flex-grow: 1 || #item3: flex-grow: 3





8. Sắp xếp vị trí các khối Flexbox - Item (tiếp)

❖ Các thuộc tính của flex-Item

- Flex-shrink: Ngược lại với grow là nó co lại

```
.item {  
  flex-shrink: <number>; /* mặc định là 1 */  
}
```



Cửa sổ bình thường co lại ít



Co cửa sổ lại thì item cũng thu nhỏ trông thấy



8. Sắp xếp vị trí các khối Flexbox - Item (tiếp)

❖ Các thuộc tính của flex-Item

- Flex-basis: Xác định độ dài ban đầu của item

```
.item {  
  flex-basis: <length> | auto; /* mặc định là auto */ }
```

Nếu ban đầu độ dài chung của class item là 100px, ta tùy chỉnh item3 với flex-basis:250px



Thuộc tính **flex** dùng để gộp chung 3 thuộc tính: flex-grow flex-shrink flex-basis

Giá trị mặc định:

flex: 0 1 auto;



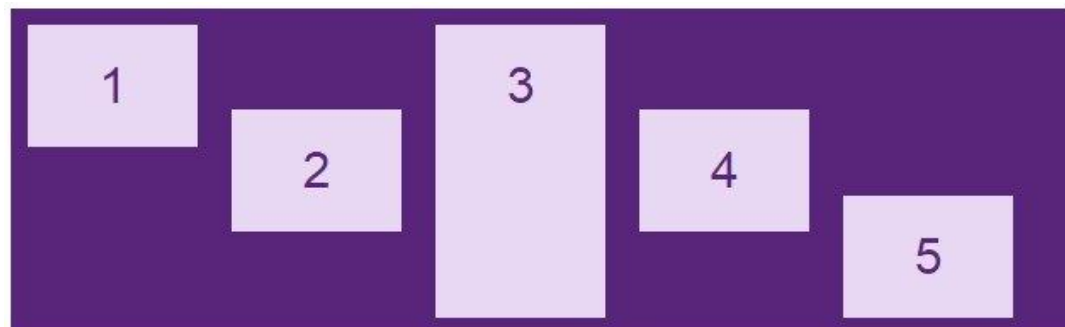
8. Sắp xếp vị trí các khối Flexbox - Item (tiếp)

❖ Các thuộc tính của flex-Item

- Align-self: có tác dụng tương tự align-items của container nhưng chỉ tác động riêng tới item đó.

```
.item {  
  align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```

```
<div class="flex-container">  
  <div style="align-self: flex-start">1</div>  
  <div>2</div>  
  <div style="align-self: stretch">3</div>  
  <div>4</div>  
  <div style="align-self: flex-end">5</div>  
</div>
```





8. Sắp xếp vị trí các khối Grid Layout

❖ Tự tìm hiểu





4. Bài tập ứng dụng

- ❖ Tạo layout cho trang web
- ❖ Tạo thanh menu ngang và menu dọc
- ❖ Sử dụng menu trên tạo sub-menu