**CSS Note ver 1.0**

1. Border:

* Hidden vs none đéo khác mẹ gì. !!!

1. Margin:

* inherit: nếu là div, chỉ ghi thừa kế margin-left: 100px;, nhưng thật ra thì vẫn thừa kế top, bottom, right của DIV.
* Collapse: chỉ cho bottom và top, và max(b, t), không cho left, right.
* AUTO;

1. Padding

* Element’s content area: vùng này là vùng của chữ, thẻ p, h,… width: tức là độ dài để mấy cái chữ hiện ra. Tức phần bên trong vùng đệm padding. 🡪 the box Model.
* box-sizing: the available Content Area space will decrease.

1. Outline:

Border -- outline -- margin; border ≈ outline.

1. Text:

* text-align: justify. giãn chữ như trong word.
* Thường trong link, sẽ có texr-decoration: none; 🡪 remove underlines from links.
* text-indent: 50px; *🡪* thụt đầu dòng.
* letter-spacing: 5px; *🡪* khoảng cách giữa các chữ cái.
* Word-spacing: 10px; 🡪 ……… giữa các chữ.
* line-height: không gian giữa các dòng.
* direction: rtl; right to left;
* text-shadow: 3px 2px red; 🡪 đây là tạo bóng chữ.

1. Font

* Thường, thì đặt font-size: 100%, 🡪 0%; ở tag body. Sau chỉnh lại theo em, của từng tag. Khi đó 1em = 16px, 8px, 4px, 2px,… 0px; || 1vw = 1% viewport width;
* font-style: normal, italics;
* font-weight: normal, bold;

1. Icons

<https://use.fontawesome.com/releases/v5.7.0/css/all.css>

* df

1. a tag

* a:hover MUST come after a:link and a:visited;
* a:active MUST come after a:hover;
* a:hover: không cần phải css lại 1 số vấn đề đã có trong a: link,… nhưng ví như border,… thay đổi thì phải thay đổi toàn bộ câu CSS.

1. List

* Số, , … nằm chồng lấn giữa vùng margin của <li> và padding của <ul>. Nếu margin == 0, thì số nằm hẳn trong padding. Nếu padding = 0, thì số nằm ở vùng tự do, hoặc margin. Padding = Margin == 0 mất mẹ cái số.
* Backup style:

List-style: circle inline url(“”);

1. Table

* Chỉ div mới có ; overflow-x: auto;

1. Display:

* Đã là inline element, sau khi đổi qua block, vẫn không cho chứa thêm trong bụng nó cái gì.
* display: none; ### visibility: hidden; hidden, vẫn chiếm S;
* display: table; thì chới nguyên 1 dòng như 1 cái table;

1. Position property

* Static: tĩnh; Relative: tương đối; Fixed: cố định; Absolute: tuyệt đối; Sticky: dính;
* Relative: so với bình thường là static;
* Fixed: so với viewport;
* Absolute: so với div cha;
* position: relative; left: 75px; Cái vùng thừa ra, đéo cho thằng nào nhảy vào cả.
* position: fixed; CỐ ĐỊNH. thường đủ 2 chiều x, y và width; để định vị. Thường là menu, … nên vùng trống còn đề làm nhiều cái khác nữa. huhu sợ vl; Tao ĐÈ lên tất cả loại position ở trên. Bố mày đè lên luôn.
* Absolute:
* ----- trừ static nha mày; không hiểu vì sao???
* haha!!! vẫn được, nhưng nhảy mẹ ra khỏi cái border của div cha static, sau định vị theo viewport chỗ div cha. Vậy nên chỉ relative, …
* vùng trống vẫn làm dược nha.

1. Overflow:

* visible: mặc định, tràn ra ngoài vùng content area;
* hidden:tràn bị cắt và ẩn cmnl, hihi.
* scroll: tự srcoll, và thanh scrollbar x, y tự có luôn.
* auto: khi chiều: x, y cần scroll mới có scrollbar.
* overflow-x: hidden; là ẩn thanh x, chứ không giống như hidden của overflow.
* overflow-y: scroll;

1. Float

* Nhớ kĩ vùng area content và vùng không gian. Đã float, thì mất hết vùng không gian. DIV cha cũng chả biết vùng không gian của nó để mà bao bọc. Bình thường ảnh và văn bản, thì div cha vẫn nhận vùng không gian và auto bao bọc. ahhuhu.
* Overflow: auto; vẫn xóa float được., thường dùng cho div CHA overflow xuống div con to bự siêu khổng lồ vê lốc.

*.clearfix::after {*

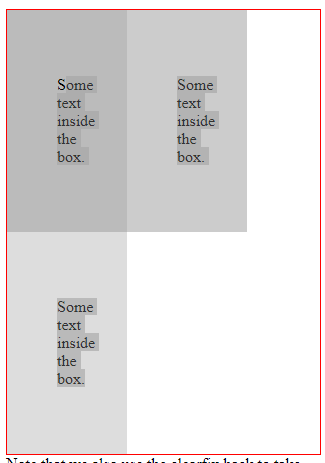
*content: "";*

*clear: both;*

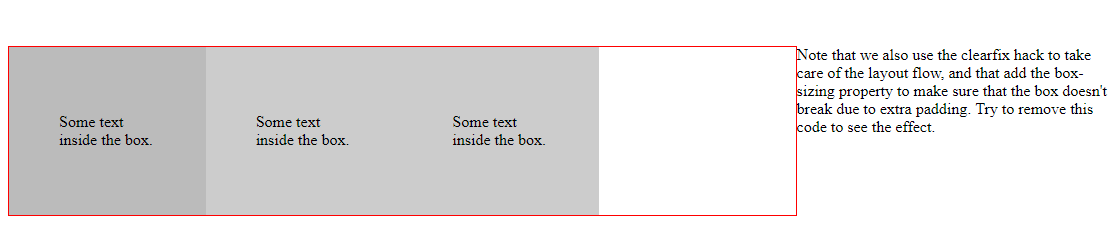
*display: table;*

*}*

* Csd
* Cha và Con cùng nổi thì sẽ ôm nhau, mặc kệ đứa con chìm kia. 🡪 cùng ôm thì cùng trôi.
* Vì <div> là GOM phần tử nên, ưu tiên phần tử float, so với văn bản của chính nó. Tức cái div con có cái float, dù ở dòng dưới vẫn float qua trái trước, chiếm mẹ nó đất ở đó luôn, iu vl.



* Vì cái div3 không đủ không gian nên thấy dư cái trắng.



* Width: 25%; LÀ THEO CÁI div cha, không phải theo cái viewport mô.

1. span:

* gom nhóm thẻ inline, trên 1 dòng, để tiện CSS. Như div thôi mà.

1. Inline-block:

* Với INLINE, width, height chả có tác dụng gì cả.
* Tiếp với INLINE: quan tâm nội dung thôi, còn lại, tất cả bị tràn hết như padding (tràn qua BLOCK trên hoặc dưới, vì nó bị xem như 1 đoạn chữ trong câu), nói chung nó không hòa hợp với BLOCK
* INLINE không chứa BLOCK được (lỗi tè le hột me). <http://localhost:8080/HTMLw3schools/Other01/new%202.html>
* Inline-block && block: tôn trọng mọi yếu tố.
* Vì trong NAV, thẻ li inline-block, nên tự có width cũng được, nên 3 cái a tự gần lại nhau;

1. Align

* Bình thường, với width, không chỉ định thì sẽ không tự động có margin, tất cả hướng margin = 0;
* Block: mặc định là width: auto, không có margin.
* 1 khi đã đặt width, thì bên right sẽ có margin thừa, nếu không đặt thêm margin, thì căn lè sát trái nhé.
* Nên muốn 1 block center, thì cứ *margin: auto;*

1. Ai cưn ba

* Trong thẻ a, nếu cho font-size và font-color: thì, nếu nó chứa <i>, i xem như là chữ, và chịu tác động CSS.

1. Transform: translate(50%, 0);

* X, Y ở đây là so với chính width, height của p, div.
* Nếu p có absolute, thì mặc định width, height, sẽ được gán y như phần wight, height auto do absolute tạo ra. Lúc này, tranform, vẫn theo width, heigth gốc này.
* Thường sẽ margin: auto; !!!

1. Transition:

* transition: [property] [duration] [timing-function] [delay]
* transition: all 10s ease 1s;
* áp dụng cho tất cả (height, width,…) trong vòng 10s, với loại ease là chậm sau nhanh (ease-in, ease-out,…), sau khi chờ 1s.

1. Other

* **a** : inline;

1. Question:

* Tại sao padding thẻ inline, lại tràn lên cả thẻ div trên, ??? phải chăng là do inline, nên chỉ quan tâm nội dung.