20230411課堂筆記

一次只先用一個色

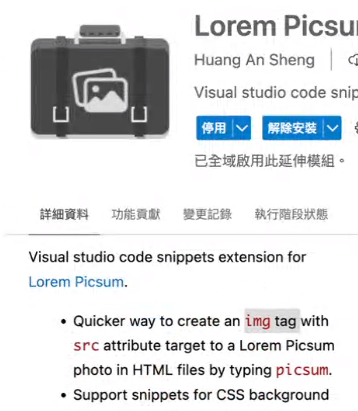
<https://nipponcolors.com/#umenezumi>

主要色 (面積大)

強調色 (面積小)

輔助色

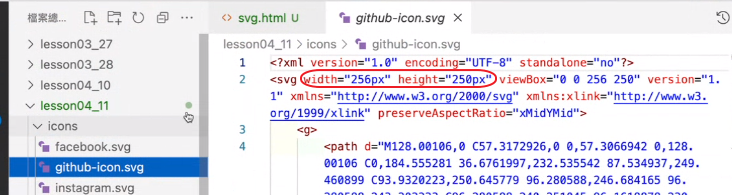
偽圖工具



Margin 與父容器的距離

SVG是向量圖

檔案上面會有原始檔大小



<svg></svg>

<picture> </picture>











viewport表示適合手機介面

<!--

srcset="<image-url> <descriptor>"

圖片尺寸使用 "w" 當單位，而不是 "px"，填入該圖檔的實際寬度

-->

<img srcset="elva-fairy-320w.jpg 320w, elva-fairy-480w.jpg 480w, elva-fairy-800w.jpg 800w" />

<!--

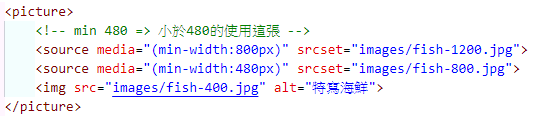
也可以使用螢幕解析度 device pixel ratio 當作依據

-->

<img src="sample.jpg" srcset="sample\_1x.jpg 1x, sample\_2x.jpg 2x" />

依不同螢幕寬顯示不同圖片

若用 min-width, 則要由大到小去設定, 先符合的會先使用

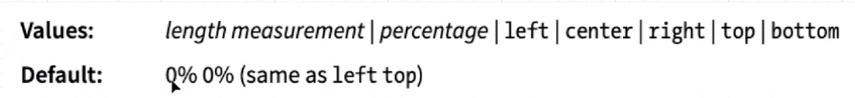


<https://www.figma.com/downloads/>

網頁做背景

2023-04-17 筆記

Background-position: 0% 0%

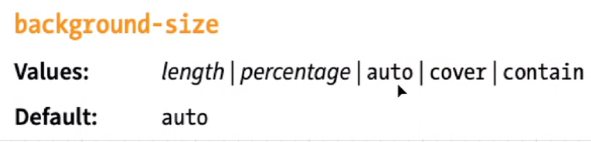




左右, 上下

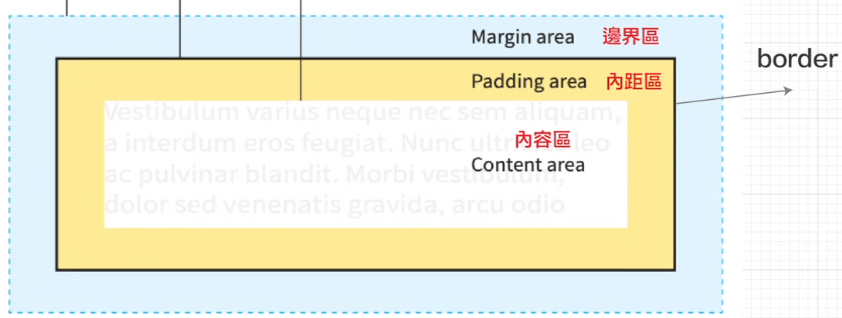
background-attachment 接合

background-size: 預設是 auto

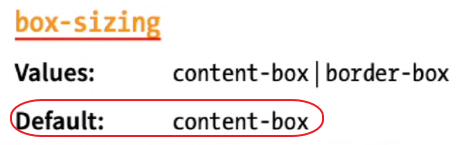


Background 寫在一行



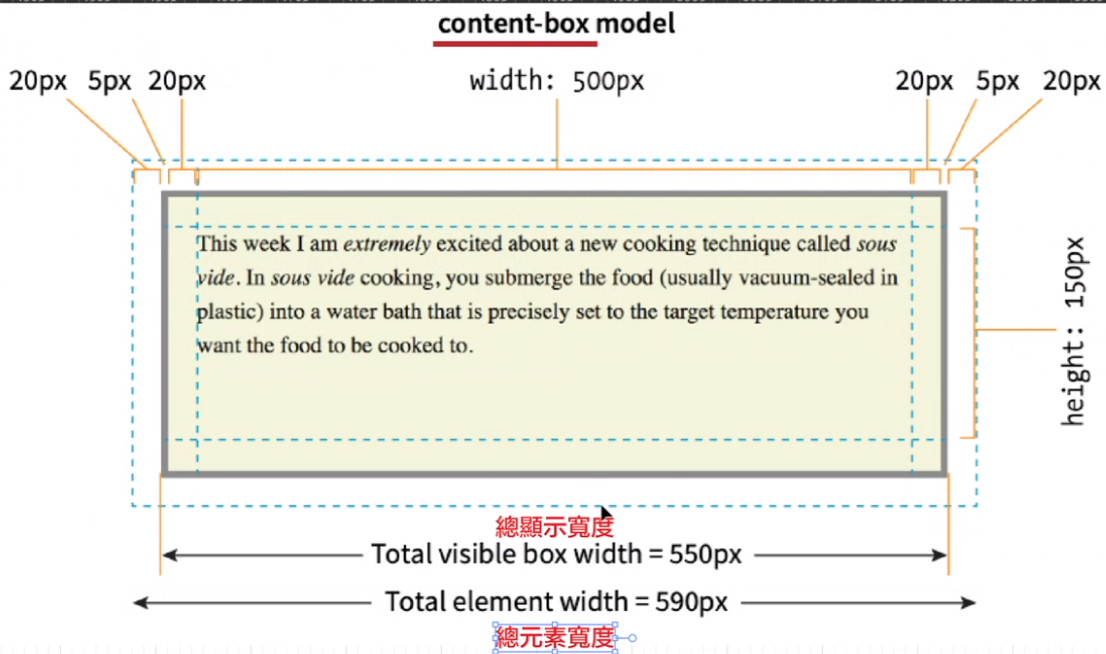


Box模型 width height box-sizing



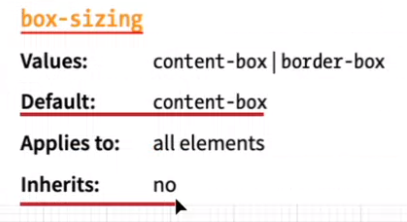
當設定box-size=content-box時, Width 設定的寬是指內容的寬度

當設定box-size=border-box時, Width 設定的寬是指內容區+padding+Border的寬度

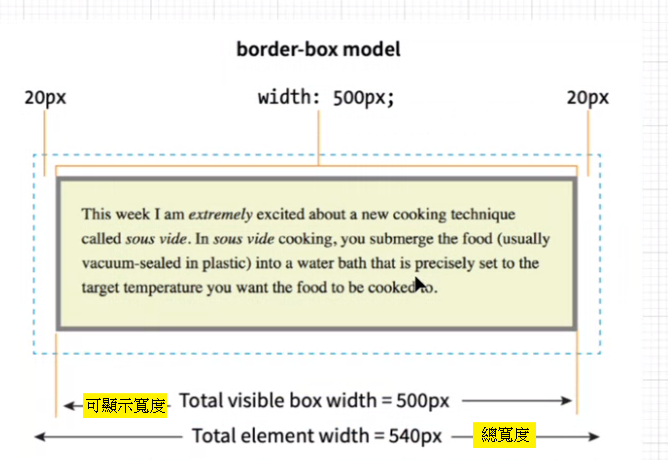


總顯示寬度: 內容寬度 content-box (預設值) width:

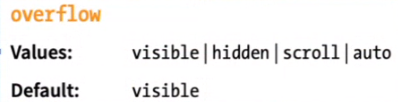
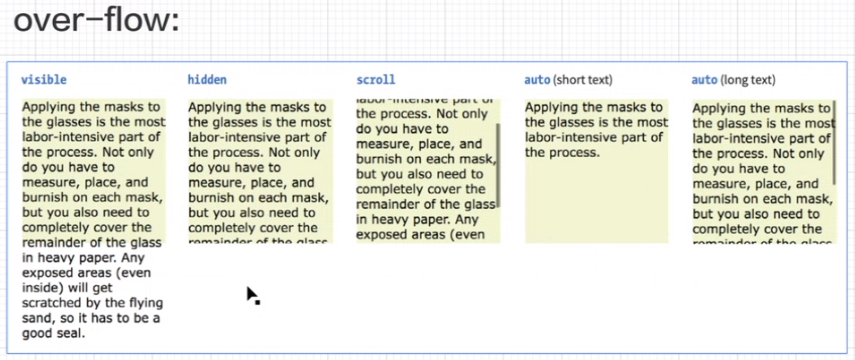




無法繼承

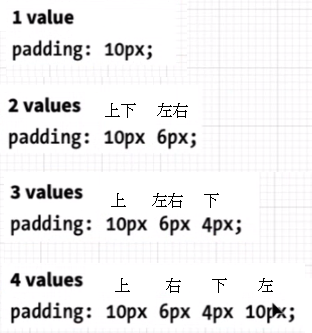


height 與 over-flow

Padding 單行簡寫

padding: 10px 11px 12px 13px;



\* 所有元素

inherit 父類別

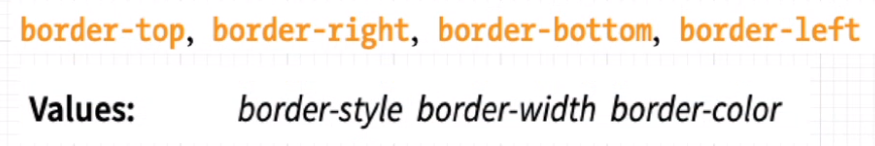
\* {

box-sizing: inherit;

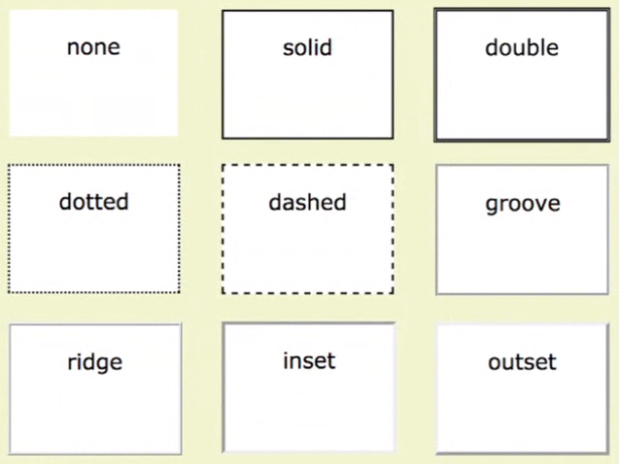
}

<main></main>

<aside></aside>



border-style



圓角 border radius



Margin 上, 下 會合併, 取最值

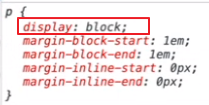
Margin: 0 20%; -> 相對於父容器的 20% 做居中的效果

**list-style-type 很常用**

首先先來了解前置符號的 list-style-type 屬性，這個屬性就是最主要的一個設定了，我們需要哪一種符號，都是利用這個屬性來控制的，它可以接受的值非常的多，至少高達 20 個之多，讓我們快速了解一下幾個重點項目就好

* none ( 沒有符號 )
* cjk-ideographic ( 一、二、三 )
* decimal ( 1、2、3 )
* decimal-leading-zero ( 01、02、03 )
* lower-alpha ( a、b、c )
* upper-alpha ( A、B、C )
* lower-roman ( i、ii、iii )
* upper-roman ( I、II、III )
* initial





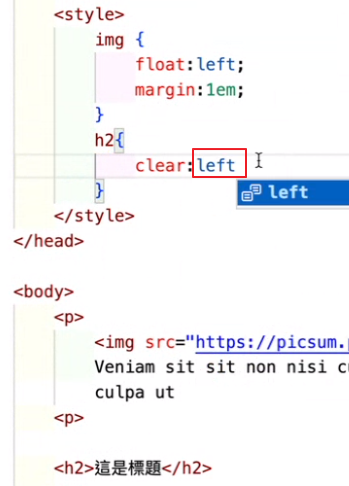
Inline 的不能有寬高

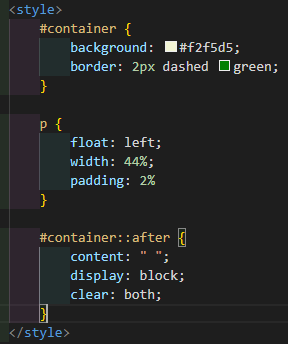
Inline 類型的像 <span> 用成float 則, 變成block,

當變成 block時, 就可以設成 寬高

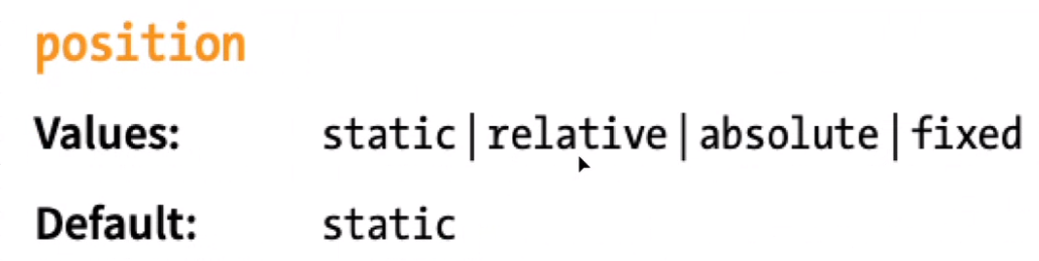
Float 要設定寬, 只有IMG本身已有寬, 所以可以不用設定

取消 float 用 clear



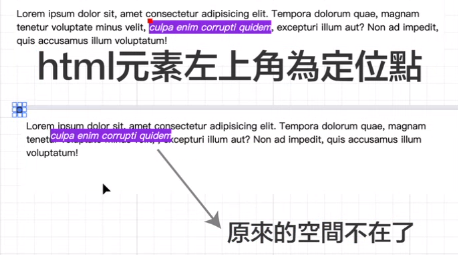


::after :after 一個冒號是偽類別 二個冒號是偽元素



Relative : 相對於原來的位置

absolute, 會找之前 有position:為定位點, 若都沒有, 則以 html 為定位點



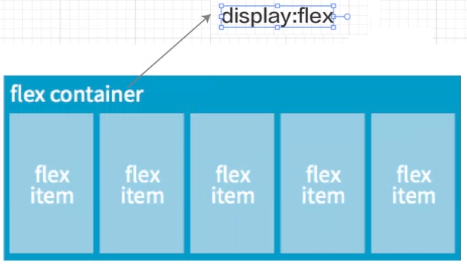
z-index 數字愈大, 在愈下面

排版規則

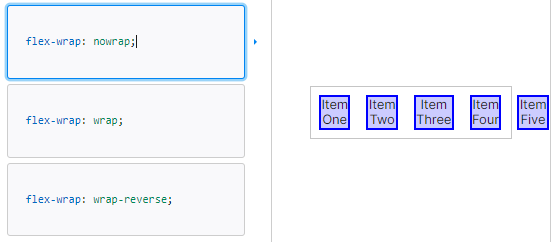
預設為 normal flow 一格一格由上而下顯示

單一色調配色

Ctrl + 0 瀏覽器 恢復 100%



**flex-wrap** 属性指定 flex 元素单行显示还是多行显示。如果允许换行，这个属性允许你控制行的堆叠方向。



**flex-direction** 属性指定了内部元素是如何在 flex 容器中布局的，定义了主轴的方向 (正方向或反方向)。

<h4>This is a Column-Reverse</h4>

<div id="content">

<div id="box" style="background-color:red;">A</div>

<div id="box" style="background-color:lightblue;">B</div>

<div id="box" style="background-color:yellow;">C</div>

</div>

<h4>This is a Row-Reverse</h4>

<div id="content1">

<div id="box1" style="background-color:red;">A</div>

<div id="box1" style="background-color:lightblue;">B</div>

<div id="box1" style="background-color:yellow;">C</div>

</div>

Css

#content {

width: 200px;

height: 200px;

border: 1px solid #c3c3c3;

display: -webkit-flex;

-webkit-flex-direction: column-reverse;

display: flex;

flex-direction: column-reverse;

}

#box {

width: 50px;

height: 50px;

}

#content1 {

width: 200px;

height: 200px;

border: 1px solid #c3c3c3;

display: -webkit-flex;

-webkit-flex-direction: row-reverse;

display: flex;

flex-direction: row-reverse;

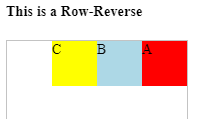
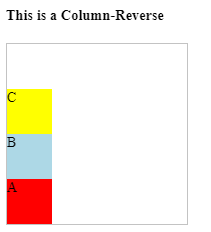
}

#box1 {

width: 50px;

height: 50px;

}



**flex-flow** 属性是 flex-direction 和 flex-wrap 的简写。

/\* Positional alignment \*/

justify-content: center; /\* 居中排列 \*/

justify-content: start; /\* Pack items from the start \*/

justify-content: end; /\* Pack items from the end \*/

justify-content: flex-start; /\* 从行首起始位置开始排列 \*/

justify-content: flex-end; /\* 从行尾位置开始排列 \*/

justify-content: left; /\* Pack items from the left \*/

justify-content: right; /\* Pack items from the right \*/

/\* Baseline alignment \*/

justify-content: baseline;

justify-content: first baseline;

justify-content: last baseline;

/\* Distributed alignment \*/

justify-content: space-between; /\* 均匀排列每个元素

首个元素放置于起点，末尾元素放置于终点 \*/

justify-content: space-around; /\* 均匀排列每个元素

每个元素周围分配相同的空间 \*/

justify-content: space-evenly; /\* 均匀排列每个元素

每个元素之间的间隔相等 \*/

justify-content: stretch; /\* 均匀排列每个元素

'auto'-sized 的元素会被拉伸以适应容器的大小 \*/

/\* Overflow alignment \*/

justify-content: safe center;

justify-content: unsafe center;

/\* Global values \*/

justify-content: inherit;

justify-content: initial;

justify-content: unset;