CSS階層式樣式表Cascading Style Sheets

樣式定義如何顯示HTML元素，通常存儲在樣式表中

外部樣式表可以極大提高工作效率，通常存儲在CSS文件中

多個樣式定義可層疊為一

層疊次序

當同一個HTML 元素被不止一個樣式定義時，會根據下面的規則層疊於一個新的虛擬樣式表中，由上至下優先權遞減

內聯樣式:當個別元素需要特殊的樣式，在HTML元素內部

內部/外部樣式表:當樣式應用於單個/很多文檔，在<head>標籤內部

瀏覽器默認設置

CSS單位

絕對單位

cm centimeters

mm millimeters

in inches (1in = 96px = 2.54cm)

px \* pixels (1px = 1/96th of 1in)

pt points (1pt = 1/72 of 1in)

pc picas (1pc = 12 pt)

相對單位

em Relative to the font-size of the element (2em means 2 times the size of the current font)

ex Relative to the x-height of the current font (rarely used)

ch Relative to the width of the "0" (zero)

rem Relative to font-size of the root element

vw Relative to 1% of the width of the viewport\*

vh Relative to 1% of the height of the viewport\*

vmin Relative to 1% of viewport's\* smaller dimension

vmax Relative to 1% of viewport's\* larger dimension

% Relative to the parent element

CSS語法

CSS註釋

放置在<style>內

<style>

/\* This is a single-line comment \*/

</style>

CSS規則集由兩個主要的部分構成：選擇器selector (需要改變樣式的HTML元素)，以及一條或多條聲明declaration，使用花括號來包圍聲明

selector {declaration1; declaration2; ... declarationN }

每條聲明由一個屬性和一個值組成，屬性property是希望設置的樣式屬性style attribute，屬性和值被冒號分開，請勿在屬性和值之間添加空格

selector {property:value}

如果值為若干單詞，則要給值加引號

p {font-family: "sans serif";}

多重聲明

如果要定義不止一個聲明，則需要用分號將每個聲明分開，最後一個聲明是不需要加分號的，然而，這麼做的好處是，當你從現有的規則中增減聲明時，會盡可能地減少出錯的可能性

p {text-align:center ;color:red ;}

你應該在每行只描述一個屬性，以增強樣式定義的可讀性

p {

text-align: center;

color: black;

font-family: arial;

}

CSS區分大小寫，如果涉及到與HTML文檔一起工作的話，class和id名稱請注意大小寫

選擇器

元素選擇器

p {

font-family: arial;

}

可以對選擇器進行分組，被分組的選擇器就可以分享相同的聲明，用逗號分開多個選擇器

h1,h2,h3,h4,h5,h6 {

color: green;

}

通用選擇器

選擇頁面上的所有HTML元素

\* {

text-align: center;

color: blue;

}

id選擇器

可以為標有特定id 的HTML 元素指定特定的樣式，以#定義，id不能以數字開頭

CSS:

#red{color:red;}

#green{color:green;}

HTML:

<p id="red">這個段落是紅色</p>

<p id="green">這個段落是綠色</p>

id屬性只能在每個HTML文檔中出現一次

id 選擇器常常用於建立派生選擇器

只會用於出現在id是sidebar的p元素內

#sidebar p {

font-style: italic;

text-align: right;

margin-top: 0.5em;

}

類選擇器

以一個.顯示，class不能以數字開頭

CSS:

.center {text-align: center}

HTML:

<h1 class="center">

This heading will be center-aligned

</h1>

<p class="center">

This paragraph will also be center-aligned.

</p>

class也可被用作後代選擇器

.fancy td {

color: #f60;

background: #666;

}

元素也可以基於它們的類而被選擇

限制於類為fancy 的td元素

CSS:

td.fancy {

color: #f60;

background: #666;

}

HTML:

<td class="fancy">

屬性選擇器

為擁有指定屬性的HTML元素設置樣式，而不僅限於class和id屬性，在為不帶有class或id的表單設置樣式時特別有用

[attribute]選擇器

具有指定屬性的元素

a[target] {

background-color: yellow;

}

[attribute="value"]選擇器

具有指定屬性和指定屬性值的元素

a[target="\_blank"] {

background-color: yellow;

}

[attribute~="value"]選擇器

具有指定屬性含有指定屬性值的元素，該指定屬性值必須是一個完整的單詞

[title~="flower"] {

border: 5px solid yellow;

}

匹配所有title包含flower的元素，summer flower、flower new皆可

[attribute|="value"]選擇器

具有指定屬性以指定屬性值開始的元素，該指定屬性值必須是一個完整的單詞

[class|="top"] {

background: yellow;

}

匹配所有class以top開頭的元素，top、top-text皆可

[attribute^="value"]選擇器

具有指定屬性以指定屬性值開始的元素，該指定屬性值不必是一個完整的單詞

[class^="top"] {

background: yellow;

}

匹配所有class以top開頭的元素，top、top text皆可

[attribute$="value"]選擇器

具有指定屬性以指定屬性值結束的元素，該指定屬性值不必是一個完整的單詞

[class$="test"] {

background: yellow;

}

匹配所有class以test開頭的元素，my-test、top text皆可

[attribute\*="value"]選擇器

具有指定屬性含有指定屬性值的元素，該指定屬性值不必是一個完整的單詞

[class\*="te"] {

background: yellow;

}

後代選擇器

指定元素(div)後代的所有元素(div後代)

div p {

background-color: yellow;

}

子元素選擇器

指定元素(div)的子元素(p)

div > p {

background-color: yellow;

}

相鄰兄弟選擇器

緊接在指定元素(div)後的元素(p)，且二者為有相同父元素的同級元素

div + p {

background-color: yellow;

}

同級選擇器

所有指定元素(div)的同級元素(p)

div ~ p {

background-color: yellow;

}

CSS顏色

RGB值

red, green, blue定義0到255之間的顏色強度，使用相等的值來定義灰色陰影

rgb(red, green, blue)

RGBA值

alpha定義0.0和1.0之間的不透明度

rgba(red, green, blue, alpha)

十六進制值

#ff0000為紅色、#00ff00為綠色、#0000ff為藍色

HSL值

hsl(hue, saturation, lightness)

色相hue:色輪上從0到360的度，0是紅色，120是綠色，240是藍色

飽和度saturation:百分比值，0％是灰色陰影，100％是全色。

亮度lightness:百分比值，0％是黑色，100％是白色

通過將色相和飽和度設置為0來定義灰色陰影，並將亮度從0％調整為100％，以獲得更深/更淺的陰影

HSLA值

alpha定義0.0和1.0之間的不透明度

hsla(hue, saturation, lightness, alpha)

CSS背景

background-color屬性指定元素的背景色

opacity屬性指定元素的不透明度/透明度，其所有子元素都繼承相同的透明度，可能會使完全透明的元素內的文本難以閱讀，建議使用RGBA值，常與:hover一起使用以更改鼠標懸停時的不透明度

div {

background-color: green;

opacity: 0.3;

}

background-image屬性指定要用作元素背景的圖像，默認在水平和垂直方向上重複圖像以覆蓋整個元素

background-repeat屬性指定僅在水平repeat-x/垂直repeat-y方向上重複或不重複no-repeat

background-position屬性用於指定背景圖像的位置

background-attachment屬性指定背景圖像應該滾動scroll還是固定fixed

body {

background-image: url("img\_tree.png");

background-repeat: no-repeat;

background-position: right top;

background-attachment: fixed;

}

速記屬性

為了縮短代碼，在一個屬性中指定所有背景屬性

屬性值的順序為：

background-color

background-image

background-repeat

background-attachment

background-position

body {

background: #ffffff url("img\_tree.png") no-repeat right top;

}

background-size屬性指定背景圖像的大小

background-origin屬性指定背景圖像的位置

屬性值:

border-box背景圖片從外邊框/邊框的左上角開始

padding-box背景圖像從內邊框/填充的左上角開始

content-box背景圖片從內容的左上角開始

background-clip屬性指定背景的繪製區域

CSS線性漸變(向下/向上/向左/向右/對角線)

至少定義兩個色標，色標是您要呈現平滑過渡的顏色，可以設置起點和方向或角度以及漸變效果

方向默認從上到下

background-image: linear-gradient(direction, color-stop1, color-stop2, ...);

background-image: linear-gradient(angle, color-stop1, color-stop2);

重複線性漸變

background-image: repeating-linear-gradient(direction, color-stop1, color-stop2, ...);

background-image: repeating-linear-gradient(angle, color-stop1, color-stop2);

CSS徑向漸變(由其中心定義)

至少定義兩個色標

形狀默認橢圓形，尺寸默認最遠角，位置默認中心，色標默認均勻間隔

background-image: radial-gradient(shape size at position, start-color, ..., last-color);

重複徑向漸變

background-image: repeating-radial-gradient(shape size at position, start-color, ..., last-color);

CSS文字

改變文本的顏色、字符間距，對齊文本，裝飾文本，對文本進行縮進等

color設置文本顏色

direction設置文本方向

unicode-bidiu設置文本是否被重寫

line-height設置行高

letter-spacing 設置字符間距

word-spacing設置字間距

vertical-align設置行内元素inline或表格單元格table-cell元素的垂直對齊方式

text-align對齊元素中的文本

text-decoration向文本添加修飾

text-indent縮進元素中文本的首行

text-transform控制元素中的字母

屬性值:

none默認，定義帶有小寫字母和大寫字母的標準的文本

capitalize 文本中的每個單詞以大寫字母開頭

uppercase定義僅有大寫字母

lowercase定義僅有小寫字母

inherit規定從父元素繼承text-transform的屬性值

CSS文字效果

CSS文字溢出

text-overflow屬性指定應如何向用戶發信號通知未顯示的溢出內容

CSS文字包裝

word-wrap屬性允許將長單詞打斷並包裝到下一行

CSS分詞

word-break屬性指定換行規則

CSS編寫模式

writing-mode屬性指定文本行是水平放置還是垂直放置

CSS陰影

text-shadow屬性為文本添加陰影

box-shadow屬性將陰影應用於元素

CSS字體

通用字體系列:

襯線Serif字體:字母的邊緣都有一個小的筆觸，營造出一種形式感和優雅感

無襯線Sans-serif字體:字體線條簡潔，沒有小筆劃，營造出現代而簡約的外觀

等寬Monospace字體:字母都具有相同的固定寬度。它們產生機械外觀

草書Cursive字體:模仿了人類的筆跡

幻想Fantasy字體:是裝飾性/俏皮字體

font-family屬性來指定文本的字體，應包含多個字體名稱作為後備，以確保瀏覽器/操作系統之間的最大兼容性，從所需的字體開始，以通用系列結束

p {

font-family: Tahoma, Verdana, sans-serif;

}

Web安全字體:

在所有瀏覽器和設備上普遍安裝的字體

Arial (sans-serif)

Verdana (sans-serif)

Helvetica (sans-serif)

Tahoma (sans-serif)

Trebuchet MS (sans-serif)

Times New Roman (serif)

Georgia (serif)

Garamond (serif)

Courier New (monospace)

Brush Script MT (cursive)

font-style屬性主要用於指定斜體文本

font-weight屬性指定字體的粗細

font-variant屬性指定是否以小寫字體顯示文本

font-size屬性設置文本的大小

普通文本，如段落的默認大小為16px/1em

1em等於當前字體大小

1vw =視口寬度的1%

速記屬性font

屬性值的順序為：

font-style

font-variant

font-weight

font-size/line-height

font-family

CSS Web字體

允許Web設計人員使用用戶計算機上未安裝的字體，只需將字體文件包含在Web服務器上，它將在需要時自動下載給用戶

在@font-face中 首先定義字體的名稱，例如myFirstFont，然後指向字體文件，字體URL始終使用小寫字母

<style>

@font-face {

font-family: myFirstFont;

src: url(sansation\_light.woff);

}

\* {

font-family: myFirstFont;

}

</style>

CSS鏈接

根據鏈接所處的狀態來設置其樣式

a:link -正常，未訪問的鏈接

a:visited -用戶訪問過的鏈接

a:hover -用戶將鼠標懸停在鏈接上時的鏈接

a:active -點擊後的鏈接

a:hover必須在定義a:link和a:visited之後才能生效

a:active必須緊隨 a:hover才能生效

CSS列表

list-style-type屬性指定列表項標記的類型

list-style-image屬性將圖像指定為列表項標記

list-style-position屬性指定列表項標記的位置

速記屬性list-style

屬性值的順序為：

list-style-type

list-style-position

list-style-image

CSS表格

border-collapse屬性設置是否將表格邊框折疊為單個邊框

width/height屬性定義表格的寬/高度

CSS佈局

塊級元素總是從新行開始，並佔據可用的全部寬度

內聯元素不會從新行開始，僅佔用所需的寬度

浮動float

float屬性指定元素應如何浮動

clear屬性指定哪些元素可以在已清除元素旁邊以及在哪一側浮動

定位position

position屬性指定用於元素的定位方法的類型

static靜態定位

默認

relative相對定位

相對於元素在文檔中的初始位置，元素仍然佔據原來的空間，移動元素會導致它覆蓋其它框

fixed固定定位

相對於視口放置，這意味著即使滾動頁面，它也始終位於同一位置

absolute絕對定位

相對於最近的祖先位置，沒有祖先，它將使用文檔正文，並隨頁面滾動一起移動，絕對定位的框與文檔流無關，所以它們可以覆蓋頁面上的其它元素，可以通過設置z-index屬性來控制這些框的堆放次序

sticky粘性定位

根據滾動位置在relative和fixed之間切換，將相對定位，直到在視口中遇到給定的偏移位置為止-然後將其“粘貼”在適當的位置

top屬性定義了一個定位元素的上外邊距邊界與其包含塊上邊界之間的偏移

right 屬性定義了定位元素右外邊距邊界與其包含塊右邊界之間的偏移

bottom屬性定義了定位元素下外邊距邊界與其包含塊下邊界之間的偏移

left屬性定義了定位元素左外邊距邊界與其包含塊左邊界之間的偏移

top right bottom left必須是定位元素才可以使用

overflow屬性設置當元素的內容溢出其區域時發生的事情

clip屬性設置元素的形狀。元素被剪入這個形狀之中，然後顯示出來

vertical-align屬性設置元素的垂直對齊方式

z-index屬性指定一個元素的堆疊順序

顯示display

display屬性指定是否/如何顯示元素，僅會更改元素的顯示方式，而不會更改元素的種類

內聯display: inline

塊display: block

內聯塊display: inline-block

<img>是內聯元素，限制高/寬度前，需用display顯示為塊級元素

CSS邊框

border-style屬性指定要顯示的邊框類型，可以具有一到四個值，用於頂部/右側/底部/左側邊框，沒有設置border-style屬性，其他CSS邊框屬性都不會有任何效果

如果缺少左側值，則使用右側值

如果缺少底部值，則使用頂部值

如果缺少右側值，則使用頂部值

border-width屬性指定四個邊框的寬度

border-color屬性用於設置四個邊框的顏色

p.one {

border-style: solid;

border-width: medium;

border-color: red;

}

速記屬性border

屬性值的順序為：

border-width

border-style

border-color

p {

border: 5px solid red;

}

border-radius屬性用於向元素添加圓角邊框

border-image屬性指定要使用的圖做為邊框

CSS外邊距/邊距

margin速記屬性

margin-top

margin-right

margin-bottom

margin-left

可以具有以下值，允許使用負值：

auto-瀏覽器計算邊距

長度-以px，pt，cm等為單位指定邊距

% -指定包含元素寬度百分比的邊距

inherit-指定應從父元素繼承邊距

margin屬性設置為auto將使元素在其容器中水平居中，元素佔據指定的寬度，剩餘空間將在左右邊界之間平均分配

CSS內邊距/填充

padding屬性可以具有以下值，不允許使用負值：

長度-以px，pt，cm等指定填充

% -指定包含元素寬度的百分比的填充

inherit -指定填充應從父元素繼承

CSS高/寬度

height/width屬性可以具有以下值：

auto-默認，瀏覽器計算高度和寬度

length -以px，cm等定義高度/寬度

% -定義包含塊的百分比的高度/寬度

initial -將高度/寬度設置為其默認值

inherit -高度/寬度將從其父值繼承

max-width屬性用於設置元素的最大寬度，覆蓋width屬性

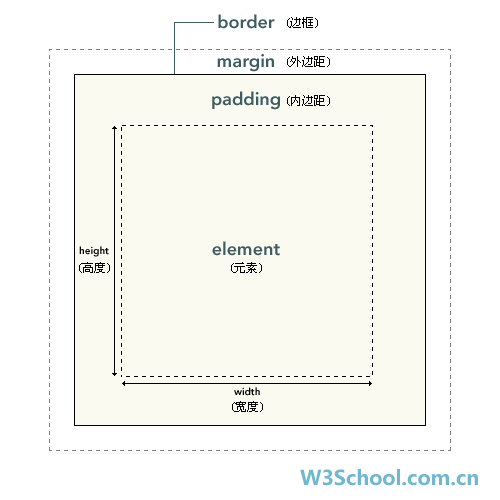
max-height

min-width

min-height

CSS框模型Box Model

元素框處理元素內容element、內邊距/填充padding、邊框border和外邊距/邊距margin的方式



背景應用於由內容和內邊距、邊框組成的區域，外邊距默認是透明

內邊距、邊框和外邊距可以應用於一個元素的所有邊，也可以應用於單獨的邊

外邊距可以是負值，而且在很多情況下都要使用負值的外邊距

當兩個垂直外邊距相遇時，它們將形成一個外邊距，合併後的外邊距的高度等於兩個發生合併的外邊距的高度中的較大者，如果沒有外邊距合併，所有段落之間的外邊距都將是相鄰上下外邊距和，即段落之間的空間是頁面頂部的兩倍，只有普通文檔流中塊框的垂直外邊距才會發生外邊距合併，行內框、浮動框或絕對定位之間的外邊距不會合併

寬度+內邊距+邊框=元素的總寬度

高度+內邊距+邊框=元素的總高度

box-sizing屬性限制元素的總寬度/高度中

用padding使文本距離邊框有一定距離，並用box-sizing限制元素的總寬度/高度，以防因為padding過大

box-sizing: border-box使得元素的總寬度/高度依循width/height屬性限制為300/100px

.div2 {

width: 300px;

height: 100px;

padding: 50px;

border: 1px solid red;

box-sizing: border-box;

}

CSS輪廓線

輪廓線與外邊框不同，是在元素外邊框之外繪製，並且可能與其他內容重疊，也不是元素尺寸的一部分，元素的總寬度和高度不受輪廓線寬度的影響

outline-style屬性指定輪廓的樣式，沒有設置outline-style屬性，其他CSS輪廓線屬性都不會有任何效果

outline-width屬性指定輪廓的寬度

outline-color屬性用於設置輪廓的顏色

速記屬性outline

屬性值的順序為：

outline-width

outline-style

outline-color

outline-offset屬性在輪廓和元素的邊緣/外邊框之間添加空間

CSS溢出

overflow屬性指定在元素的內容太大而無法放入指定區域時是剪輯內容還是添加滾動條

overflow-x/overflow-y屬性指定是否只是水平/垂直地改變的內容溢出

屬性值：

visible默認，溢出沒有被修剪，內容在元素框外渲染

hidden溢出被裁剪，其餘內容將不可見

scroll剪輯了溢出，並添加了滾動條以查看其餘內容

auto剪輯了溢出，僅在必要時添加滾動條以查看其餘內容

CSS偽類

用於定義元素的特殊狀態

selector:pseudo-class {

property: value;

}

:first-child匹配指定元素的第一個子元素

:lang允許定義不同語言的特殊規則

CSS偽元素

用於為元素的指定部分設置樣式

selector::pseudo-element {

property: value;

}

::first-line用於將特殊的樣式添加到文本的第一行，只能應用在塊級元素

::first-letter用於將特殊的樣式添加到文本的第一個字母，只能應用在塊級元素

::before用於插入某些內容到一個元素內容之前

::after用於插入某些內容到一個元素內容之後

::selection改變用戶選擇部分的樣式

CSS導航欄

導航欄基本上是鏈接列表

<ul>

<li><a href="default.asp">Home</a></li>

<li><a href="news.asp">News</a></li>

<li><a href="contact.asp">Contact</a></li>

<li><a href="about.asp">About</a></li>

</ul>

從列表中刪除項目符號以及邊距和填充

ul {

list-style-type: none;

margin: 0;

padding: 0;

}

CSS計數器

counter-reset創建或重置計數器

counter-increment增加計數器值

content插入生成的內容

counter()將計數器的值添加到元素

body {

counter-reset: section;

}

h2::before {

counter-increment: section;

content: "Section " counter(section) ": ";

}

CSS 2D轉換

transform屬性

屬性值:

translate()從當前位置移動元素

rotate()順時針或逆時針旋轉元素

scaleX()增加或減少元素的高度

scaleY()增加或減少元素的寬度

scale()增加或減少元素的寬度和高度

skewX()使元素沿X軸傾斜

skewY()使元素沿Y軸傾斜

skew()使元素沿X和Y軸傾斜

matrix()將所有2D變換方法組合

CSS 3D變形

transform屬性

屬性值:

rotateX()繞X軸旋轉元素

rotateY()繞Y軸旋轉元素

rotateZ()繞Z軸旋轉元素

CSS過渡

在給定的時間內平穩地更改屬性值

創建過渡效果，必須指定兩件事：

要添加效果的CSS屬性

效果的持續時間

div {

width: 100px;

height: 100px;

background: red;

transition: width 2s;

}

div:hover {

width: 300px;

}

transition-property屬性指定過渡效果所針對的CSS屬性的名稱

transition-duration屬性指定完成過渡效果需要多少時間

transition-timing-function屬性指定過渡效果的速度曲線

transition-delay屬性指定過渡效果的延遲

transition速記屬性

屬性值的順序為：

transition-property

transition-duration

transition-timing-function

transition-delay

CSS動畫

使元素逐漸從一種樣式變為另一種樣式，必須先指定一些關鍵幀(元素在特定時間將具有的樣式)

@keyframes內指定CSS樣式時，動畫會在特定時間逐漸從當前樣式更改為新樣式

from和 to代表0%和100%

@keyframes example {

from {background-color: red;}

to {background-color: yellow;}

}

或

@keyframes example {

0% {background-color: red;}

25% {background-color: yellow;}

50% {background-color: blue;}

100% {background-color: green;}

}

animation-name屬性為@keyframes動畫指定名稱

animation-duration屬性定義動畫花費多長時間，動畫結束後回歸原本樣式animation-timing-function屬性指定動畫的速度曲線

屬性值:

linear

ease

ease-in

ease-out

ease-in-out

cubic-bezier(n, n, n, n)

線性貝茲曲線

由P0至P1的連續點B(t)，描述一條線性貝茲曲線

二次貝茲曲線

由P0至P1的連續點Q0，描述一條線性貝茲曲線

由P1至P2的連續點Q1，描述一條線性貝茲曲線

由Q0至Q1的連續點B(t)，描述一條二次貝茲曲線

三次貝茲曲線cubic-bezier

由線性貝茲曲線描述的中介點Q0、Q1、Q2，和由二次曲線描述的點R0、R1所建構

animation-delay屬性指定動畫開始的延遲，可以為負數

animation-iteration-count屬性指定動畫應運行的次數

animation-direction屬性指定動畫是向前播放、向後播放、先向前後向後交替播放、先向後後向前交替播放

div {

width: 100px;

height: 100px;

background-color: red;

animation-name: example;

animation-duration: 5s;

animation-timing-function: linear;

animation-delay: 2s;

animation-iteration-count: infinite;

animation-direction: alternate;

}

animation速記屬性

屬性值的順序為：

animation-name

animation-duration

animation-timing-function

animation-delay

animation-iteration-count

animation-direction

animation-fill-mode屬性在不播放動畫時(在開始之前，結束之後或兩者)指定目標元素的樣式

animation-play-state屬性指定動畫是運行還是暫停，在JavaScript中使用此屬性可以在循環的中間暫停動畫

CSS對象適合屬性

object-fit屬性用於指定如何調整<img>或<video>的大小以適合其容器

CSS多列佈局

column-count屬性指定元素應劃分為的列數

column-width屬性為列指定瀏覽器的最小寬度建議，一旦瀏覽器無法以指定的寬度容納，將減少列數以維持寬度

column-gap屬性指定列之間的間隙

column-rule-width屬性指定列之間的規則寬度

column-rule-style屬性指定列之間的規則樣式

column-rule-color屬性指定列之間的規則顏色

<style>

.newspaper {

column-count: 3;

column-gap: 40px;

column-rule-style: solid;

column-rule-width: 1px;

column-rule-color: lightblue;

}

</style>

column-rule速記屬性

屬性值的順序為：

column-rule-width

column-rule-style

column-rule-color

<style>

.newspaper {

column-count: 3;

column-gap: 40px;

column-rule: 1px solid lightblue;

}

</style>

column-span屬性指定元素應跨越多少列

<style>

h2 {

column-span: all;

}

</style>

CSS用戶界面

resize屬性指定元素是否以及如何由用戶調整大小

CSS變量

可以訪問DOM，意味著可以創建局部或全局變量，使用JavaScript以及媒體查詢更改變量，可以將涉及設計的顏色放在變量中，而不必一遍又一遍地複制和粘貼

變量名稱必須以兩個破折號--開頭，並且區分大小寫，--color和--Color是兩個變量

全局變量可以在整個文檔中使用，在:root選擇器中聲明

局部變量只能在聲明它的選擇器內部使用，元素中同名稱的局部變量將覆蓋全局變量

/\*聲明變量\*/

<style>

:root {

--blue: #1e90ff;

--white: #ffffff;

}

/\*使用變量，var()函數用於插入CSS變量\*/

body {

background-color: var(--blue);

}

h2 {

border-bottom: 2px solid var(--blue);

}

</style>

CSS媒體查詢Media Queries MQ

用於檢查許多事情，例如:視口的寬度和高度、設備的寬度和高度、設備橫向還是縱向模式、解析度，向不同的媒介提供定制的樣式表

@media規則

在相同的樣式表中，使用不同的樣式規則來針對不同的媒介

媒體類型mediatype

all所有

print 影印

screen螢幕

speech螢幕閱讀器，閱讀頁面並讀出來

CSS:

@media not|only mediatype and (expressions) {

CSS-Code;

}

用not或only，必須有介質類型mediatype

HTML:

<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">

CSS 彈性布局Flexbox

彈性容器Flex container (父元素)

在display屬性用上flex或inline-flex值

display: flex

彈性容器的外部顯示類型是塊級元素，內部顯示類型是flex，彈性容器的子元素參與彈性布局

display: inline-flex

彈性容器外部顯示類型是行內元素，內部顯示類型是flex，彈性容器的子元素參與彈性布局

彈性項目Flex item

彈性容器的子元素都會變成彈性項目，直接包含在彈性容器的文字，會被包裝成匿名的彈性項目

主軸main axis

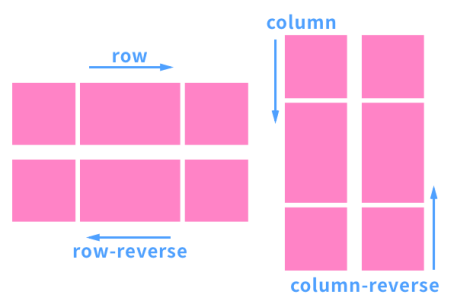
跟隨著彈性項目順序的軸

切軸cross axis

垂直於主軸的軸

彈性容器屬性:

flex-direction屬性定義容器要在哪個方向上堆疊彈性項目，啟用主軸



flex-wrap屬性指定彈性項目超出範圍是否換行/列

nowrap默認設置，不換行/列

wrap換行/列

wrap-reverse換行/列，但方向相反

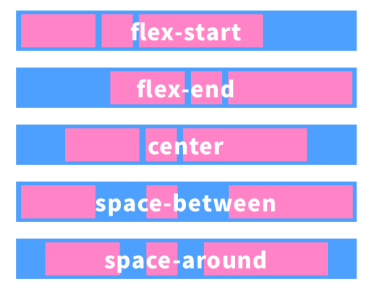
flex-flow速記屬性

屬性值的順序為：

flex-direction

flex-wrap

justify-content屬性定義彈性項目的主軸上對齊方向



flex-start將項目與flex容器的開頭對齊，默認對齊方式

flex-end將項目與flex容器的末端對齊

center將所有flex項目與flex容器內的中心對齊

space-between將項目與主軸中心對齊，在項目之間放置額外的空間，第一個和最後一個項目被推到flex容器的最邊緣

項目 空間 項目 空間 項目

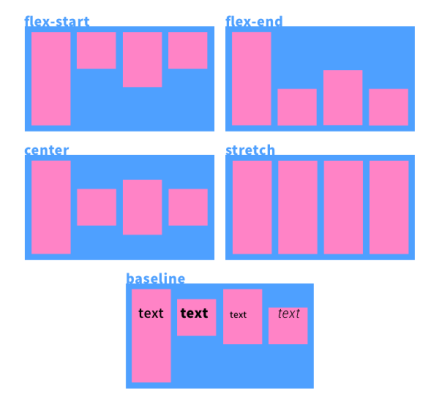
space-around類似於space-between，但第一個和最後一個項目未鎖定到容器的邊緣，該空間分佈在所有項目周圍

半個空間 項目 空間 項目 半個空間

space-evenly在flex項目之間平均分配空間，在flex容器的任一端都有一個完整的空間

空間 項目 空間 項目 空間

align-items屬性定義彈性項目的切軸上對齊方向



flex-start將項目與flex容器的開頭對齊

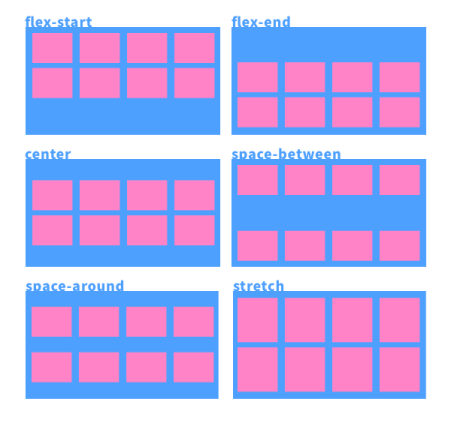
flex-end將項目與flex容器的末端對齊

center將項目對準中心

stretch拉伸項目以填充彈性容器，默認值

baseline將項目與基線對齊，基線是一個文本概念，可將其視為字母所在的行

align-content屬性定義彈性項目的切軸上對齊方向，align-items屬性的多行版本



align-content會操控整個內容(content)

align-content:center，所有彈性內容會以對齊"彈性容器的中間"為準

align-items 屬性會操控彈性行中的每個彈性項目(items)

align-items:center，所有的彈性項目都會對齊"它所在的那個彈性行的中線"

彈性項目屬性:

order屬性定義彈性項目的排列順序

flex-grow屬性定義當空間分配還有剩餘時，彈性項目的伸展性

flex-shrink屬性定義當空間分配還不足時，彈性項目的收縮性

flex-basis屬性定義彈性項目的基準值

flex速記屬性

屬性值的順序為：

flex-grow

flex-shrink

flex-basis

align-self屬性定義彈性項目的切軸上對齊方向，可以個別設定單一彈性項目的值，將覆蓋align-items屬性設置的對齊方式

網格佈局Grid Layout

一個基於網格的佈局系統，該系統具有行和列，可以更輕鬆地設計網頁，而無需使用浮動和定位，由一個父元素和一個或多個子元素組成

網格容器Grid container (父元素)

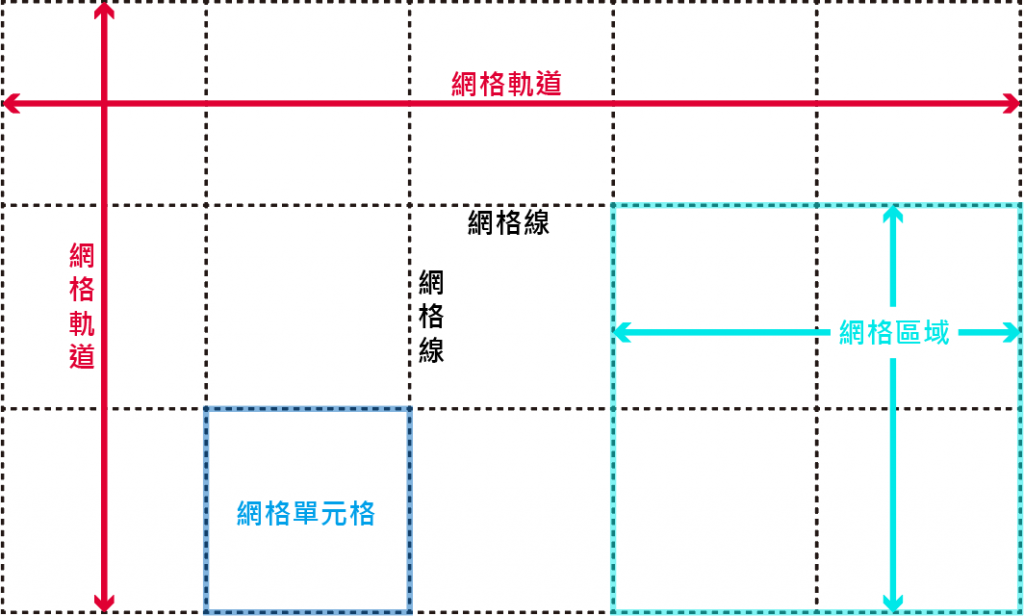
在display屬性用上grid或inline-grid值

display: grid;

display: inline-grid;

網格項目Grid item

網格容器的子元素都會變成網格項目



網格區域area

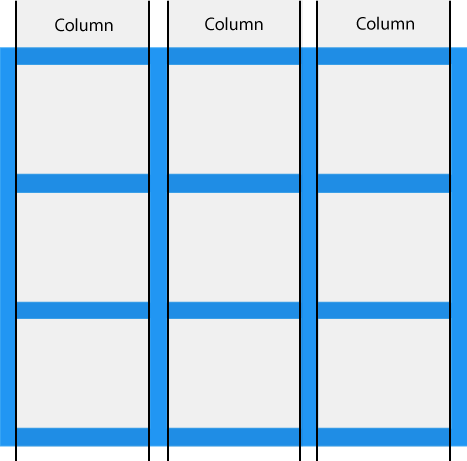
由四條自定義的網格線或邊緣組成的某個封閉區域

單元格cell

行和列的交叉區域，最小的網格區域單位

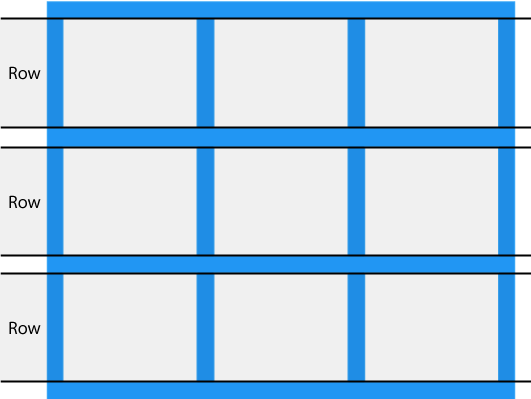
列column

垂直的網格項目



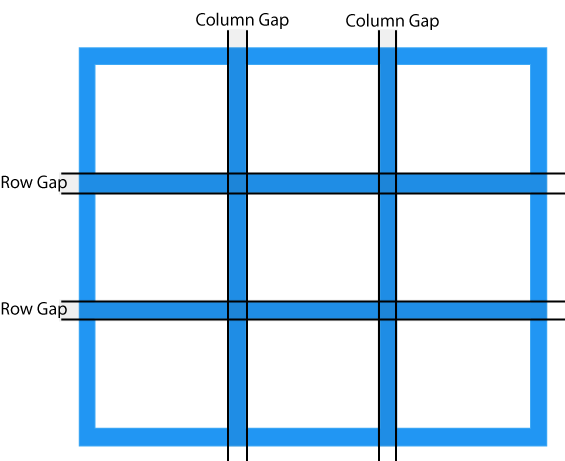
行row

水平的網格項目



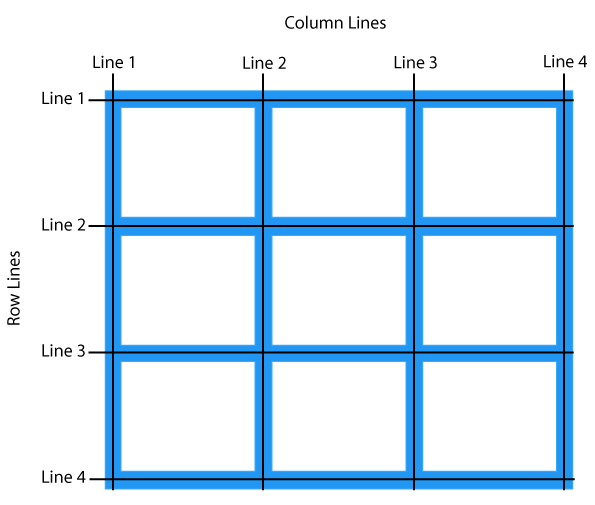
間隙gap

每列/行之間的間隔



網格線grid line

劃分網格的線，分為行之間的行線、列之間的列線



網格容器屬性:

grid-template-columns屬性定義網格佈局中的列數及每列的寬度

grid-template-rows屬性定義網格佈局中的行數及每行的高度

repeat函數

接受兩個參數，第一個參數是重複列或行的次數，第二個參數是要重複的值

grid-template-columns: repeat(2, 100px 20px 80px);

重複2次，定義了6列，第一列和第四列的寬度為100px，第二列和第五列為20px，第三列和第六列為80px

minmax函數

接受兩個參數，分別為最小值和最大值，產生一個長度範圍

minmax(100px, 1fr)

列寬不小於100px，不大於1fr

grid-template-columns: repeat(3, minmax(100px, 1fr));

重複3次，定義了3列，列寬不小於100px，不大於1fr

auto-fill

單元格的大小是固定的，但有時容器的大小不確定，如果希望每一行或列，容納盡可能多的單元格，這時可以使用auto-fill關鍵字表示自動填充

grid-template-columns: repeat(auto-fill, 100px);

每列寬度100px，然後自動填充，直到容器不能放置更多的列

fr

為了方便表示比例關係，網格佈局提供了fr(片段fraction)，如果兩列的寬度分別為1fr和2fr，就表示後者是前者的兩倍

grid-template-areas屬性定義網格區域

grid-auto-flow 屬性定義網格項目的排列順序

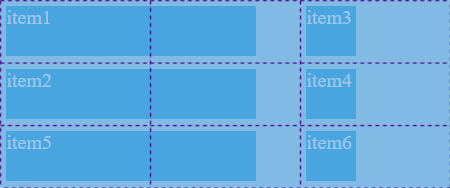
row預設，網格項目會優先佔據整"行"，再以列的方向一行一行堆疊下去



column網格項目會優先佔據整"列"，再以行的方向一列一列堆疊下去



row dense



column dense

只要加上 dense，網格項目會優先考慮把網格容器填滿，不會因為項目序號較大而留下空格，除非放不進去

一些項目的指定位置，在現有網格的外部，比如網格只有3列，但是某一個項目指定在第5行，瀏覽器會自動生成多餘的網格，以便放置項目

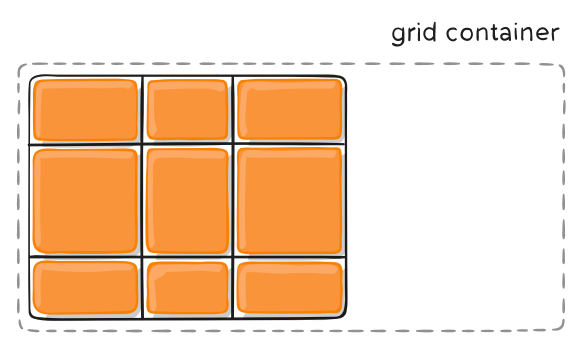
grid-auto-columns屬性設置瀏覽器自動創建的多餘網格的每列的寬度

grid-auto-rows屬性設置瀏覽器自動創建的多餘網格的每行的高度

對齊屬性:

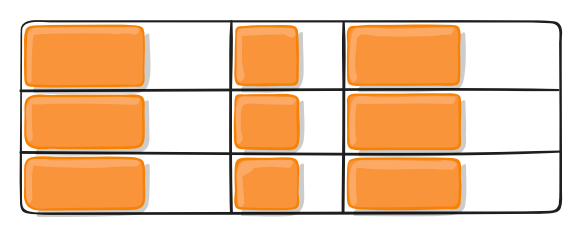
justify-content屬性定義網格區域在網格容器內水平對齊方向

align-content屬性定義網格區域在網格容器內垂直對齊方向



justify-items屬性定義單元格內容在單元格內水平對齊方向

align-items屬性定義單元格內容在單元格內垂直對齊方向



間隙屬性:

grid-row-gap屬性設置行之間的間隙大小

grid-column-gap屬性設置列之間的間隙大小

grid-gap速記屬性

屬性值的順序為：

grid-row-gap

grid-column-gap

網格項目屬性:

grid-row-start屬性定義網格項目從第幾行線開始

grid-row-end屬性定義網格項目從第幾行線結束

grid-row速記屬性

屬性值的順序為：

grid-row-start

grid-row-end

grid-column-start屬性定義網格項目從第幾列線開始

grid-column-end屬性定義網格項目從第幾列線結束

grid-column速記屬性

屬性值的順序為：

grid-column-start

grid-column-end

行線1、列線1開始，行線4、列線5結束，網格項目大小為3行、4列

grid-row: 1 / 4;或grid- row: 1 / span 3;

grid-column: 1 / 5;或 grid-column: 1 / span 4;

grid-area速記屬性

屬性值的順序為：

grid-row-start

grid-column-start

grid-row-end

grid-column-end

grid-area屬性還可以指定網格項目放在哪一個網格區域

grid-template-areas定義網格區域，一個網格區域由單個或多個單元格組成，一個單元格大小為1行、1列，一個點號.代表一個沒有名稱的單元格，這個語法本身可視作網格的視覺化結構

<style>

/\* class=item1，網格項目放在header網格區域\*/

.item1 { grid-area: header; }

.item2 { grid-area: menu; }

.item3 { grid-area: main; }

.item4 { grid-area: right; }

.item5 { grid-area: footer; }

.grid-container {

display: grid;

grid-template-areas:

/\* header網格區域位在第一行的一至六單元格\*/

'header header header header header header'

'menu . main main right right'

'menu footer footer footer footer footer';

}

</style>

<body>

<div class="grid-container">

<div class="item1">Header</div>

<div class="item2">Menu</div>

<div class="item3">Main</div>

<div class="item4">Right</div>

<div class="item5">Footer</div>

</div>

</body>

對齊屬性:

justify-self屬性定義單元格內容在單元格內水平對齊方向，可以個別設定單一單元格內容的值，將覆蓋justify-items屬性設置的對齊方式

align-self屬性定義單元格內容在單元格內水平對齊方向，可以個別設定單一單元格內容的值，將覆蓋align-items屬性設置的對齊方式

order屬性定義網格項目的排列順序

z-index屬性定義網格項目的重疊順序

響應式網頁設計Responsive Web Design RWD

使用CSS和HTML調整大小，隱藏，縮小，放大或移動內容以使其在任何媒介上看起來都很好

原則:

1.請勿使用較大的固定寬度元素，如果圖像的寬度大於視口的寬度，則可能導致視口水平滾動

2.不要讓內容依賴於特定的視口寬度來呈現效果很好，因為屏幕尺寸和CSS像素寬度在設備之間差異很大

3.使用CSS媒體查詢為小屏幕和大屏幕應用不同的樣式