Visual Studio Code

## 插件

Open in Browser 右击选择在浏览器中打开HTML文件

JS-CSS-HTML Formatter 每次保存，都会自动格式化js css和HTML代码

Auto Rename Tag 自动重命名配对的HTML/XML标签

CSS Peek 追踪样式

## Emmet語法

Emmet語法的前身是Zen coding，它使用縮寫，來提高html/css的編寫速度。Vscode內部已集成該語法。

### （1）Emmet快速生成HTML結構語法

零 生成HTML框架：輸入!，按tab鍵

① 生成標籤：直接輸入標籤名，按tab鍵

② 生成多個相同標籤：輸入div\*3，按tab鍵

③ 父子級關係：輸入ul>li，按tab鍵

④ 兄弟關係：輸入div+p，按tab鍵

⑤ 生成帶有類名或者id名字的div：輸入.demo或者#demo，按tab鍵

⑥ 如果類名或者id名帶有序號：輸入.demo$\*5，按tab鍵，可以得到：

<div class="demo1"></div>

<div class="demo2"></div>

<div class="demo3"></div>

<div class="demo4"></div>

<div class="demo5"></div>

⑦標籤裡默認顯示文字：

輸入div{天朗氣清}\*5，按tab鍵

<div>天朗气清</div>

<div>天朗气清</div>

<div>天朗气清</div>

<div>天朗气清</div>

<div>天朗气清</div>

輸入div{$}\*5，按tab鍵

<div>1</div>

<div>2</div>

<div>3</div>

<div>4</div>

<div>5</div>

### （2）Emmet快速生成CSS結構語法

①寫單詞開頭字母。比如text-align：輸入ta，然後按tab鍵或者回車。

②寫寬：輸入w100，然後按tab鍵或者回車。寫高同理。

③寫縮進：輸入ti2e，然後按tab鍵或者回車。

### （3）格式化代碼

Visual Studio Code中，在代碼文件中單擊右鍵，選擇Format Document，可以自動格式化代碼。

在保存代碼文件時，自動格式化代碼的設置：

打開File→Preferences→Settings

搜索Emmet.include

必須選擇User下的項目

點擊 Edit in settings.json



添加：

“editor.formatOnType”: true,

“editor.formatOnSave”: true

HTML基礎知識

## HTML概念

HTML指的是超文本標記語言（Hyper Text Markup Language），它是用來描述網頁的一種語言。

HTML是標記語言，而非編程語言。

超文本，有2層含義：

1. 它可以加入圖片、聲音、動畫、多媒體等內容（超越了文本限制）。

2. 它還可以從一個文件跳轉到另一個文件，與世界各地主機的文件連接（超級鏈接文本）。

## 常用瀏覽器以及內核

常用瀏覽器：谷歌，IE，Edge，火狐，Safari，Opera

瀏覽器內核（渲染引擎）：負責讀取網頁內容，整理訊息，計算網頁的顯示方式並顯示頁面。

|  |  |  |
| --- | --- | --- |
| 瀏覽器 | 內核 | 備註 |
| IE | Trident | IE，獵豹安全，360急速瀏覽器，百度瀏覽器 |
| Firefox | Gecko | 火狐瀏覽器內核 |
| Safari | Webkit | 蘋果瀏覽器內核 |
| Chrome，Opera | Blink | Chrome/ Opera瀏覽器內核，Blink其實是WebKit的分支 |

國產瀏覽器一般採用Webkit/ Blink內核，如360，UC，QQ，搜狗等。

## Web標準

Web標準是由W3C組織和其它標準化組織制定的一系列標準的集合。

W3C（萬維網聯盟）是國際著名的標準化組織。

Web標準的必要性：

瀏覽器不同，同樣的代碼，在不同的瀏覽器上有不同的顯示。遵循Web標準，可以讓開發代碼更規範，這樣在不同的瀏覽器裡，用戶看到的效果基本一致。

Web標準構成：

① 結構 structure 對網頁元素進行整理和分類 HTML

② 表現 presentation 設置網頁元素的板式、顏色、大小等外觀樣式 CSS

③ 行為 behaviour 網頁模型的定義及交互的編寫 JavaScript

## VSCode工具生成骨架標籤新增代碼

1

<!DOCTYPE> 文檔類型聲明，告訴瀏覽器使用哪種HTML版本來顯示網頁。

<!DOCTYPE html> 這段代碼意思：當前頁面採用的是HTML5版本來顯示網頁。

<!DOCTYPE>位置：位於文檔中最前面的位置，處於<html>標籤之前。

<!DOCTYPE>不是HTML標籤，是文檔類型聲明標籤。

2

<html lang=”en”>

定義文檔語言，英文是en，中文是”zh-CN”，法語只”fr”。其實不論定義那種語言，文檔中都能寫中英雙語。

這個屬性對瀏覽器和搜索引擎有作用。比如Chrome讀到這裡，會提示，是否需要翻譯。

3

<meta charset=”UTF-8”>必須寫，否則亂碼。

字符集（Character set）是多個字符的集合，以便計算機能夠識別和存儲各種文字。

在<head>標籤內，通過<meta>標籤的charset屬性來規定HTML文檔應該使用哪種字符編碼。

charset常用的值有：GB2312、BIG5、GBK和UTF-8，其中UTF-8被稱為萬國碼，基本包含了全世界所有國家需要用到的字符。

## 加粗，傾斜，刪除，下劃線

|  |  |  |
| --- | --- | --- |
| 語義 | 標籤 | 推薦使用下列標籤，因為語義更強烈 |
| 加粗 | <strong></strong>或者<b></b> | strong |
| 傾斜 | <em></em>或者<i></i> | em |
| 刪除線 | <del></del>或者<s></s> | del |
| 下劃線 | <ins></ins>或者<u></u> | ins |

## img標籤

屬性：title 屬性值：文本 說明：提示文本，鼠標放到圖像上，顯示的文字。

## 合併單元格

跨行：最上側單元格為目標單元格，寫合併代碼

跨列：最左側單元格為目標單元格，寫合併代碼

合併單元格3步：

1. 確定跨行還是跨列合併

2. 找到目標單元格，寫上合併方式=合併的單元格數量。比如：<td colspan= “2”></td>

3. 刪除多餘的單元格

## 自定義列表

商務網站首頁底部的佈局，一般用自定義列表

基本語法如下：

<dl>

<dt>名詞1</dt>

<dd>名詞1解釋1</dd>

<dd>名詞1解釋2</dd>

<dd>名詞1解釋3</dd>

</dl>

## 表單標籤



表單域=form標籤

表單控件=表單元素

## <label>標籤

<label>標籤為input元素定義標註（標籤）

<label>標籤用於綁定一個表單元素，當點擊<label>標籤內的文本時，瀏覽器會自動將焦點（光標）轉到或者選擇對應的表單元素上，用來增加用戶體驗。

語法：

<label for=”gender”>女</label>

<input type=”radio” name=”gender” id=”gender” />

其中lable的for屬性和input的id屬性值要一樣

## HTML元素顯示模式

HTML元素顯示模式：塊元素，行內元素，行內塊元素。

### （1）塊元素

<h1>到<h6>，<p>，<div>，<ul>，<ol>，<li>等

特點：

① 自己佔一行。

② 可改變高、寬、外邊距、內邊距。

③ 默認寬度是容器（父級寬度）的100%。

④ 是一個容器/盒子，裡面可以放塊元素、行內元素或者行內塊元素。

注意：

文字類的元素（<h1>到<h6>，<p>）內不能使用塊元素。

### （2）行內元素

行內元素=內聯元素

<a>，<span>，<strong>，<b>，<em>，<i>，<del>，<s>，<ins>，<u>

特點：

① 相鄰行內元素在一行上，一行可以顯示多個。

② 無法直接設置寬、高。

③ 默認寬度是內容的寬度。

④ 行內元素只能容納文本或其它行內元素。

注意：

① 鏈接裡面不能再放鏈接。

② 鏈接<a>裡面可以放塊級元素，但是最好把<a>轉換成塊元素。

#### em製作小豎線

<style>

      em {

        font-style: normal;

        color: #ebe4e0;

        margin: 0 6px 0 15px; /\* 竖线和左右盒子的距离 \*/

      }

    </style>

  </head>

  <body>

    <em>|</em>

  </body>

### （3）行內塊元素

<img />，<input />，<td>同時具有塊元素和行內元素的特點。

特點：

① 一行可以顯示多個行內塊元素，但元素之間有空隙。

② 默認寬度是本身內容的寬度。

③ 可以設置高、寬、行高、外邊距、內邊距。

### （4）元素顯示模式轉換

比如要增加<a>的觸發範圍。把<a>從行內元素變成塊元素。

變成塊元素方式：

display: block;

變成行內元素方式：

display: inline;

變成行內塊元素方式：

display: inline-block;

HTML5

HTML5增加了新標籤，新表單，新表單屬性。

這些新特性都有兼容性問題，基本是IT9+以上的版本的瀏覽器才支持，如果不考慮兼容性問題，可以大量使用這些新特性。

## HTML5新增語義化標籤

<header>：頭部標籤

<nav>：導航標籤

<article>：內容標籤

<section>：定義文檔某個區域（大號的div）

<aside>：側邊欄標籤

<footer>：尾部標籤

語義化標籤針對搜索引擎

這些新標籤可以在頁面中多次使用

在IE9中，需要把這些元素轉換為塊級元素，因為在IE9裡面，它們默認狀態下，不是塊級元素。

移動端更常用這些標籤

## HTML5新增多媒體標籤

### （1）視頻：<video>

使用它們可以很方便地在網頁中嵌入音頻和視頻，而不再去使用flash和其它瀏覽器插件。

當前<video>元素支持三種視頻格式，盡量使用mp4格式。



語法

<video src=”文件地址” controls=”controls”></video>

為了兼容性更好，可以這樣寫（了解即可，現在瀏覽器兼容性都比較好了）：

<video width=”320” height=”240” controls>

<source src=”movie.mp4” type=”video/mp4”>

<source src=”movie.ogg” type=”video/ogg”>

您的瀏覽器不支持video標籤

</video>

設置video標籤的值時，最好寫雙標籤，比如autoplay=”autoplay” muted=”muted” loop=”loop”



### （2）音頻：<audio>

當前<audio>元素支持三種音頻格式

所有主流瀏覽器都支持MP3

語法：

<audio src=”文件地址” controls=”controls”></audio>

如果一定要考慮兼容性的問題：

< audio controls="controls" >

<source src="happy.mp3" type="audio/mpeg" >

<source src="happy.ogg" type="audio/ogg" >

您的浏览器暂不支持 <audio> 标签。

</ audio>



谷歌瀏覽器把音頻和視頻自動播放都禁止了，用JavaScript解決禁止音頻自動播放問題，用muted解決禁止視頻自動播放問題。

## HTML5新增input表單



重點記住：number，tel，search這三個



上面的表單樣式由下列代碼寫成

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

  </head>

  <body>

    <!-- 驗證的時候必須添加form表單域 -->

    <form action="">

      <ul>

        <li>郵箱：<input type="email" /></li>

        <li>網址：<input type="url" /></li>

        <li>日期：<input type="date" /></li>

        <li>時間：<input type="time" /></li>

        <li>數量：<input type="number" /></li>

        <li>手機號碼：<input type="tel" /></li>

        <li>搜索：<input type="search" /></li>

        <li>顏色：<input type="color" /></li>

        <!-- 點擊提交按鈕可以驗證表單 -->

        <li><input type="submit" value="提交" /></li>

      </ul>

    </form>

  </body>

</html>

## HTML5新增表單屬性

|  |  |  |
| --- | --- | --- |
| **屬性** | **值** | **說明** |
| required | required | 表單擁有該屬性表示其內容不能為空，必填 |
| placeholder | 提示文本 | 表單的提示信息，存在默認值將不顯示 |
| autofocus | autofocus | 自動聚焦屬性，頁面加載完成自動聚焦到指定表單 |
| autocomplete | off / on | 當用戶在字段開始鍵入時，瀏覽器基於之前鍵入過的值，應該顯示出在字段中填寫的選項。  默認已經打開，如autocomplete=“on”。成功提交過才會有autocomplete的內容。關閉是autocomplete=“off”，一般情況下，為了安全，會關閉。需要放在表單內，同時加上name屬性。 |
| multiple | multiple | 可以多選文件提交 |

可以通過以下設置方式修改placeholder裡面的字體顏色：

input::placeholder {

color: pink;

}

CSS

## CSS代碼風格

1. 樣式格式書寫

使用展開式的格式：

h3 {

color: pink;

font-size: 20px;

}

2. 樣式大小寫

寫樣式時，用小寫字母。

大寫字母也能執行，但是不規範。

3. 空格

① 屬性值前面和冒號的後面，寫1個空格。比如上面的pink和20前面都有1個空格。

② 選擇器（標籤）和大括號中間保留空格。比如上面h3和大括號之間有空格。

## CSS選擇器

CSS選擇器分為基礎選擇器和復合選擇器兩大類。

### 1. CSS基礎選擇器

基礎選擇器由單個選擇器組成

基礎選擇器包括：標籤選擇器，類選擇器，id選擇器，通配符選擇器

#### （1）標籤選擇器（=元素選擇器）

指用HTML標籤名稱作為選擇器，為頁面中某一類標籤指定統一的CSS樣式。

比如把所有的<div>字體變成紅色，把所有的<span>字體變成藍色。

例子：

h3 {

color: pink;

font-size: 20px;

}

#### （2）類選擇器

單獨選擇一個或者某幾個標籤，可以使用類選擇器。

語法

.類名 {

屬性1: 屬性值1;

…

}

類選擇器口訣：樣式點定義，結構類調用；一個或多個，開發最常用。

例子：

定義

.tea {

color: red;

}

調用

<div class=’tea’>變紅色</div>

不要用標籤名字作為類名。

一個標籤可以調用多個類。兩個類名之間用空格分開。

例子<div class=’red size’>變紅色</div>

#### （3）ID選擇器

語法

#id名{

屬性1: 屬性值1;

…

}

例子

定義

#tea {

color: red;

}

調用

<div id=”tea”>茶葉的顏色</div>

id選擇器只能被調用1次；類選擇器可以被多次調用。

類選擇器在修改樣式中用得最多，id選擇器一般用於頁面唯一性的元素上，經常和JavaScript搭配使用。

#### （4）通配符選擇器

通配符選擇器用“\*”定義，它表示選取頁面中的所有元素（標籤）。定義後系統會自動調用。

語法

\* {

屬性1: 屬性值1;

…

}

### 2. CSS復合選擇器

把基礎選擇器組合在一起，就形成了復合選擇器。

常用的復合選擇器包括：後代選擇器，子選擇器，並集選擇器，偽類選擇器等。

#### （1）後代選擇器（常用）

後代選擇器=包含選擇器

例子

    <style>

      ol li {

        font-weight: bold;

      }

    </style>

  </head>

  <body>

    <ol>

      <li>天朗气清</li>

      <li>天朗气清</li>

      <li>天朗气清</li>

    </ol>

    <ul>

      <li>惠风和畅</li>

      <li>惠风和畅</li>

      <li>惠风和畅</li>

    </ul>

元素1和元素2中間用空格隔開。

元素1 是父級，元素2是後代，最終選擇的是元素2。

元素2可以是子級，也可以是孫級，只要是元素1 的後代即可。

元素1和元素2可以是任意基礎選擇器。

#### （2）子選擇器

子元素選擇器（子選擇器）只能選擇作為某元素的最近一級子元素=子級元素

語法：

元素1>元素2（樣式聲明）

例子：

    <style>

      .nav > a {

        font-weight: bold;

      }

    </style>

  </head>

  <body>

    <div class="nav">

      <a href="">天朗气清</a>

      <p><a href="">惠风和畅</a></p>

    </div>

  </body>

#### （3）並集選擇器（常用）

並集選擇器可以選擇多組標籤，同時為它們定義相同的樣式。通常用於集體聲明。

各個選擇器通過英文逗號連接而成，任何形式的基礎選擇器都可以作為並集選擇器的一部分。

語法：

元素1, 元素2 {樣式聲明}

例如：

ul, div {

font-size: 20px;

}

語法規範：並集選擇器豎著寫。最後一個選擇器後面不用加逗號。

<style>

      div,

      p,

      .tea>li {

        font-size: 20px;

      }

    </style>

#### （4）偽類選擇器

偽類選擇器作用：①給鏈接添加特殊效果，②選擇第1個、第n個元素。

書寫特點：用冒號表示。例子 :hover :first-child。

##### （4-①）連接偽類選擇器（常用）

語法：

a:link 選擇所有未被訪問的連接

a:visited 選擇所有已被訪問的連接

a:hover 選擇鼠標指針位於其上的連接 （最常用）

a:active 選擇活動連接（鼠標按下未彈起的連接）

為了確保生效，按照LVHA的順序聲明：:link :visited :hover :active

例子：

a {

color: gray;

}

a:hover {

color: red;

}

##### （4-②）focus偽類選擇器

:focus 用於選取獲得焦點的表單元素，主要針對<input>

例子：

input:focus {

background-color:yellow;

}

## CSS的font

### （1）font-family

如果字體名字由多個單詞構成，要給字體名字加引號。單引號和雙引號都可以。

如果寫了多個字體，每個字體之間用逗號+空格隔開。瀏覽器會根據列出的字體順序逐個去檢查，優先使用已經安裝了的、排在前面的字體。如果所有字體都沒安裝，就使用瀏覽器默認的字體。Chrome默認的是微軟雅黑

例子：

p {

font-family: ‘Times New Roman’, Times, serif, Arial;

}

字體名字可以用中文，用引號包裹，但是用英文會更規範。

h2 {

font-family: ‘微軟雅黑’;

}

微軟雅黑的英文：

h2 {

font-family: ‘Microsoft yahei’;

}

在開發裡，把字體定義進body標籤

### （2）font-size

有單位px，一定要寫上。

Chrome默認字體大小為16px

不同瀏覽器默認顯示的字號大小不一致，盡量給一個明確大小值，不要默認。

設置了

body {

font-size: 16px;

}

標題的字號不會跟著變，需要另外設置標題的字號大小，比如

h2 {

font-size: 16px;

}

### （3）font-weight

font-weight: 700; （沒有單位）等於 font-weight: bold

標題文字默認是加粗的，如果想讓標題文字變細：

h2 {

font-weight: 400;

}

或者

h2 {

font-weight: normal;

}

在實際開發中，數字用得更多。

### （4）font-style

font-style的值有2個：normal和italic

平時很少給文字加斜體，一般是把斜體標籤（em，i）改為不傾斜字體。

例子：

em {

font-style: normal;

}

### （5）字體復合屬性

可以把上述的font屬性寫在一行。

語法：

body {

font: font-style font-weight font-size/ line-height font-family;

}

要按照上面的順序來寫，不能隨意顛倒順序。

各個屬性間用空格隔開。

不需要設置的屬性可以省略，但必須保留font-size和font-family屬性。否則font不起作用。

例子1：

div {

font: italic 700 16px ‘Microsoft yahei’;

}

例子2：

body {

font: 16px/28px ‘Microsoft yahei’

}

上述代碼意思：所有字體設為16px，行高28px，使用微軟雅黑字體。

## CSS修飾文本

文本屬性定義文本的外觀，比如文本的顏色、對齊、裝飾（刪除線、下劃線）、文本縮進、行間距等。

### （1）color

color的屬性值，可以用顏色的英文單詞（red，green等）、16進制值（#FF0000，#29D795等）、RGB代碼（ rgb(255,0,0)或者rgb(100%,0%,0%) ）。用得最多的是16進制值。

16進制的值，如果兩兩相同，可以只寫3個值

比如#666666可以寫為#666；#ff00ff可以寫為#f0f

### （2）text-align

設置元素內文本內容的水平對齊方式

例子：讓h1盒子裡的文字水平居中對齊（h1自己占1行的）

h1 {

text-align: center;

}

text-align 的值：left（默認左對齊），right，center

### （3）text-decoration

添加下劃線

div {

text-decoration: underline;

}

刪除下劃線

a {

text-decoration: none;

}

### （4）text-indent

em是個相對單位，相當於當前元素（font-size）1個文字的大小。

比如當前字體大小為16px，那麼1em=16px

如果當前元素沒有設置大小，則會按照父元素的1個文字大小。

如果中文文章段首要縮進2個漢字，就可以不用px而用em這個單位。

比如：

p {

text-indent: 2em;

}

### （5）line-height

完整的行間距由3部分構成：上間距+文本高度+下間距



比如給font-size: 16px的div設定line-height: 16px; 那麼上間距和下間距都是0。

如果給font-size: 16px的div設定line-height: 26px; 那麼上間距和下間距都是5。

在PS切圖測量行高時，從第一行的下沿測量到第二行的下沿，就是行高了。

### （6）單行文字垂直居中

設置方式：文字的行高=盒子的高度

原理







## CSS引入方式

### （1）行內樣式表（行內式）

在元素標籤內部的style屬性中設定CSS樣式。

用得非常少，適用於簡單修改樣式。

書寫方便，權重高。

例子：

<div style=”color: red; font-size: 12px;”>碧螺春</div>

style是標籤的屬性。

要用雙引號。

### （2）內部樣式表（嵌入式）

寫在HTML頁面內部，把所有的CSS代碼抽取出來，單獨放到<style>標籤中。

理論上CSS代碼可以放在HTML頁面的任何位置，但是一般放在<head>部分<title>的後面。

初級練習的時候使用。

### （3）外部樣式表（鏈接式）

把樣式寫在CSS文件中，然後在HTML文件中引入。

在開發中用得最多的。

在HTML頁面中引入CSS文件的語法：

<link rel=”stylesheet” href=”css文件路徑”>

放在<head>部分<title>的後面

## CSS背景

### （1）背景顏色

默認值：

background-color: transparent;

取值可以是顏色單詞，16進制值，RGB代碼。

### （2）背景圖片

背景圖片比插入圖片更容易控制位置。

頁面元素可以同時添加背景圖片和背景顏色。背景圖片在背景顏色上層。

運用場景：logo，裝飾性小圖片，超大背景圖片，精靈圖

background-image: none | url(url)

none為默認值

例子：

background-image: url(images/logo.png);

### （3）背景平鋪

background-repeat: repeat | no-repeat | repeat-x | repeat-y

上述值的意思：平鋪，不平鋪，沿x軸平鋪，沿y軸平鋪。

默認值是repeat

### （4）背景圖片位置（重要）

background-position: x y;

參數代表的意思是：x坐標和y坐標。可以使用方位名詞，也可以使用精確單位。

#### ①參數是方位名詞：

top，center，bottom，left，center，right

①-a）如果指定的2個值都是方位名詞，兩個值前後順序無關。

比如這2個是一樣的：

background-position: left top;

background-position: top left;

比如這2個也是一樣的（都是水平靠右，垂直居中）：

background-position: right center;

background-position: center right;

①-b）如果只指定了一個方位名詞，另一個值省略，則第二個值默認center。

#### ②參數是精確單位：px，%

第1個參數必定是x坐標，第2個參數必定是y坐標。

例子：

background-position: 50px 20px;

如果只寫1個參數，那個參數必定是x軸的。

比如：

background-position: 50px; 這是x軸上的50px，另外一個值默認是垂直方向的center了。

#### ③參數混合使用

第1個值一定是x坐標，第2個值一定是y坐標。

例子：

background-position: 20px center; x軸是20px，y軸是center。

background-position: center 20px; x軸是center ，y軸是20px。

### （5）固定背景圖片

background-attachment屬性設置背景圖像是否固定，或者隨著頁面的其餘部分滾動。

background-attachment應用場景：製作視差滾動的效果。

語法：

background-attachment: scroll | fixed

scroll是默認值

### （6）背景屬性復合寫法（常用）

沒有特定的書寫順序，一般習慣約定順序為：

background: 背景顏色 背景圖片地址 背景圖片是否平鋪 背景圖片是否滾動 背景圖片位置;

各個值用空格隔開。

### （7）背景顏色半透明

background: rgba(0, 0, 0, 0.3);

a是alpha的簡稱，取值範圍在0-1之間。

3個0是黑色。上面這個例子是黑色的半透明。

上面0.3裡面的0可以省略，寫成：background: rgba(0, 0, 0, .3);

這只是讓盒子的背景色半透明，盒子的內容不受影響。

這是CSS3新增屬性，是IE9+版本瀏覽器才支持的。但是現在實際開發，不太關注兼容性寫法了，可以放心使用。

## CSS三大特性

层叠性，继承性，优先级

### （1）层叠性

相同選擇器給設置相同的樣式，一個樣式就會覆蓋（層疊）另一個衝突的樣式。

①樣式衝突，遵循的原則是就近原則：哪個樣式離結構近，就執行哪個樣式。

比如這個例子裡，最終執行pink顏色

<style>

      div {

        color: blue;

      }

      div {

        color: pink;

      }

    </style>

  </head>

  <body>

    <div></div>

  </body>

② 樣式不衝突，就不會覆蓋

下面的例子中，只有pink會覆蓋blue，height不會被覆蓋

<style>

      div {

        color: blue;

        height: 30px;

      }

      div {

        color: pink;

      }

    </style>

  </head>

  <body>

    <div></div>

  </body>

### （2）繼承性

子標籤會繼承父標籤的某些樣式。

繼承的樣式一般是：text-，font-，line-，color，行高。

高度、內外邊距不會繼承。

body {

font: 12px/1.5 Microsoft YaHei;

}

行高可以寫單位，也可以不寫。上面的例子就沒有寫單位，直接寫了1.5。這指的是行高時當前文字大小（font-size）的1.5倍。

比如下面的例子，是要body裡面所有子元素的行高時font-size的1.5倍。div的font-size是14px，那麼line-height就是21px。

<style>

      body {

        font: 12px/1.5 Microsoft YaHei;

      }

      div {

        font-size: 14px;

      }

    </style>

  </head>

  <body>

    <div></div>

  </body>

這樣寫的優勢：讓子元素根據自己的文字大小來調整行高。

### （3）優先級

選擇器相同，執行層疊性。

選擇器不同，根據選擇器權重執行。

|  |  |
| --- | --- |
| 選擇器 | 選擇器權重（依次提高）（權重會疊加，但是不會進位） |
| 繼承 或者 \*（通配符選擇器） | 0，0，0，0 |
| 元素選擇器 | 0，0，0，1 |
| 類選擇器，偽類選擇器 | 0，0，1，0 |
| ID選擇器 | 0，1，0，0 |
| 行內樣式 style=“” | 1，0，0，0 |
| !important 重要的 | ∞無窮大 |

標籤a有默認樣式：顏色藍色，有下劃線。這個時候，如果其父級元素雖然設置了顏色，和text-decoration，但由於是繼承來的，權重小於a本身的元素選擇器權重，所以顯示的a是藍色有下劃線。

#### 優先級——權重疊加

div ul li → 0，0，0，3

.nav ul li → 0，0，1，2

a:hover → 0，0，1，1

.nav a → 0，0，1，1

## CSS盒子模型

### （1）網頁佈局過程

① 準備好相關的網頁元素，網頁元素基本都是盒子Box

② 利用CSS設置盒子樣式，然後擺放到相應位置

③ 往盒子裡裝內容

### （2）盒子模型（Box Model）組成

外邊距margin，邊框border，內邊距padding，內容content。

#### ①外邊距

a）margin的簡寫方式和padding完全一致

#### b）外邊距典型運用：讓塊級盒子水平居中

條件：

① 盒子指定了寬度（width）；

② 盒子左右外邊距設置為auto，常見寫法如下3種：

margin-left: auto; margin-right: auto;

margin: auto;

margin: 0 auto;

注意：標準流、相對定位的盒子才可以這樣寫。絕對定位和固定定位的盒子，這樣寫無效。

#### c）讓行內元素或者行內塊元素水平居中

讓行內元素或者行內塊元素水平居中：給父元素添加text-align: center;

#### d）相鄰塊元素垂直外邊距的合併

當兄弟關係的上下2個塊元素相遇時，如果上面的元素有下外邊距margin-bottom，下面的元素有上外邊距margin-top，他們之間的垂直間距不是margin-bottom和margin-top之和，而是兩個值之中的加大者，這種現象被稱為相鄰塊元素垂直外邊距的合併。



解決方案：盡量只給一個盒子添加margin值

#### e）嵌套塊元素垂直外邊距的塌陷

對於2個嵌套關係（父子關係）的塊元素，父元素有上外邊距，同時子元素也有上外邊距，此時上外邊距會變成較大的外邊距值。



解決方案：

* 為父元素定義上邊框。
* 為父元素定義上內邊距padding-top。
* 為父元素添加overflow: hidden。
* 還有其他方法，比如浮動、固定、絕對定位的盒子不會有塌陷問題。

#### f）行內元素的外邊距

為了兼容性更好，盡量只設置行內元素左右的內外邊距，不要設置上下的內外邊距。

行內元素轉為塊級元素和行內塊元素就可以四邊都設置內外邊距了。

#### ②邊框

border: border-width（粗細，單位px） | border-style | border-color

border-style的值：none（無邊框），solid（實線邊框），dotted（虛線邊框），dashed（虛線邊框）

邊框值的復合寫法，任意順序：

border: 1px solid red;

只設定某一條邊框：

border-top: 1px solid red; （這是上邊框，其餘同理）

邊框會影響盒子寬高。寫盒子寬高時，要減去border值。

#### ③內邊距

padding簡寫

padding: 5px; 上下左右都有5個像素內邊距；

padding: 5px 10px; 上下5px，左右10px；

padding: 5px 10px 20px; 上5px，左右10px，下20px；

padding: 5px 10px 20px 30px; 上右下左。

內邊距會影響盒子寬高。寫盒子寬高時，要減去padding值。

只要不指定盒子寬高，padding就不會撐開盒子。

清除內外邊距

\* {

padding: 0;

margin: 0;

}

## CSS浮動

網頁佈局的本質：用CSS來擺放盒子

傳統網頁佈局的三種方式：

* 普通流（標準流）（標籤按照默認的方式排列）
* 浮動
* 定位

### （1）浮動定義

float屬性用於創建浮動框，將其移動到一邊。直到左邊緣或者右邊緣觸及包含塊或另一個浮動框的邊緣。

語法：

選擇器 {float: 屬性值;}

屬性值：

* none 元素不浮動（默認值）
* left 元素向左浮動
* right 元素向右浮動

### （2）浮動特性

#### （2-1）浮動元素會脫離標準流（脫標）

① 脫標：脫離標準普通流的控制，移動（浮動）到指定位置

② 浮動的盒子不再保留原先的位置

例子：兩個div，第1個加了左浮動，第2個標準流。那麼視覺上，第1個div會壓住第2個div。

③ 浮動只會影響後面的標準流，不會影響前面的。

#### （2-2）浮動元素會一行內顯示並且元素頂部對齊

#### （2-3）浮動元素會具有行內塊元素的相似特性

任何元素都可以浮動。不管原先是什麼模式的元素，添加浮動之後具有行內塊元素相似的特性。

給行內元素設置了浮動，則不需要轉換為塊元素/行內塊元素，就可以直接給寬度和高度。

如果塊元素沒有設置寬度，默認寬度和父級元素一樣寬；添加浮動後，如果沒有設置寬度，其大小根據內容來決定。

### （3）浮動元素經常搭配標準流的父元素

先用標準流的父元素排列上下位置，之後內部子元素採取浮動排列左右位置。



### （4）常見網頁佈局





上面3個圖依次更複雜。

其中top和footer不需要設置寬度，默認和瀏覽器一樣寬。

## CSS清除浮動

### （1）為什麼要清除浮動

不方便給父盒子高度的場景：

① 產品很多

② 每篇新聞字數不一樣

這時候，讓子盒子撐開父盒子，不給父盒子高度。這時候，出現一個問題：當所有子盒子浮動，父盒子高度會變成0；父盒子所在標準流下面的盒子，會佔據本應屬於父盒子的空間。

### （2）清除浮動本質

父盒子根據浮動的子盒子自動檢測高度。父盒子有了高度，就不會影響下面的標準流了。

### （3） 清除浮動語法

選擇器{clear: 屬性值;}

|  |  |
| --- | --- |
| **屬性值** | **描述** |
| left | 不允許左側有浮動元素（清除左側浮動的影響） |
| right | 不允許右側有浮動元素（清除右側浮動的影響） |
| both | 同時清除左右兩側浮動的影響 |

實際工作中，最常用clear: both;

### （4） 清除浮動方法① 額外標籤法=隔墻法

在浮動元素末尾添加一個空標籤（必須是塊級元素），並寫上清除浮動的語法。

比如<div style=”clear: both”></div>

優點：通俗易懂，書寫方便。

缺點：添加許多無意義的標籤，影響結構優化。

雖然W3C推薦這麼寫，但講師不推薦。

### （5）清除浮動方法② 父級添加overflow屬性

給父盒子添加overflow屬性，將值設為hidden（常用）、auto或者scroll。

優點：代碼簡潔

缺點：無法顯示父盒子溢出的部分

### （6）清除浮動方法③ 父級添加after偽元素

.clearfix: after {

content: “”,

display: block;

height: 0;

clear: both;

visibility: hidden;

}

IE6、7專用：

.clearfix {

\*zoom: 1;

}

把上面的2組代碼（即class=”clearfix”）運用到父盒子上。

優點：沒有增加標籤，結構更簡單。

缺點：要照顧低版本瀏覽器。

代表網站：百度、淘寶、網易。

### （7）清除浮動方法④ 父級添加雙偽元素

.clearfix: before, .clearfix: after {

content: “”;

display: table;

}

.clearfix: after {

clear: both;

}

.clearfix {

\*zoom:1;

}

把上面的3組代碼（即class=”clearfix”）運用到父盒子上。

優點：代碼更簡潔。

缺點：照顧低版本瀏覽器。

代表網站：小米，騰訊等。

講師自己喜歡這種方式。

## 定位

### （1）定位組成

定位=定位模式+邊偏移

定位模式通過position設置，值有4個：

static 靜態定位

relative 相對定位

absolute 絕對定位

fixed 固定定位

邊偏移有top，bottom，left和right四個屬性。

|  |  |  |
| --- | --- | --- |
| 邊偏移屬性 | 示例 | 描述 |
| top | top: 80px; | 頂端偏移量，定義元素相對於其父元素上邊線的距離。 |
| bottom | bottom: 80px; | 底部偏移量，定義元素相對於其父元素下邊線的距離。 |
| left | left: 80px; | 左側偏移量，定義元素相對於其父元素左邊線的距離。 |
| right | right: 80px; | 右側偏移量，定義元素相對於其父元素右邊線的距離。 |

### （2）靜態定位

默認定位方式，無定位。

靜態定位按照標準流特性擺放位置，沒有邊偏移。

很少用到靜態定位。

### （3）相對定位（重要）

① 參照它自己原來的位置來移動。

② 相對定位前的位置以標準流的形式繼續保留。

### （4）絕對定位（重要）

① 元素移動位置時，相對於祖先元素。如果沒有祖先元素或者祖先元素沒有定位，則以瀏覽器為準定位（Document文檔）。

② 如果祖元素有定位（相對，絕對，固定定位），則以**最近一級**的**有定位**祖元素為參考點移動位置。

③ 絕對定位不再佔有原先的位置，脫離標準流了。

### （5）子絕父相

① 子級絕對定位，不會佔有位置，可以放到父盒子裡面的任何一個地方，不會影響其它的兄弟盒子。

② 給父盒子加定位，以限制子盒子，讓子盒子只在父盒子內移動。

不能給父盒子加絕對定位，因為加了絕對定位的父盒子會脫離標準流，進而影響整個頁面佈局。所以只能加相對定位。

### （6）固定定位（重要）

① 主要使用場景：瀏覽器頁面滾動時，元素的位置不變。

② 已瀏覽器**可視窗口**為參照點移動元素。根據瀏覽器界面放大縮小靈活變化。跟父元素沒關係。不隨滾動條滾動。

③ 固定定位不佔有原先的位置。

固定定位是脫標的，可以把固定定位看做是一種特殊的絕對定位。

### （7）固定定位小技巧：固定在版心右側位置

第①步：讓固定定位的盒子left: 50%，走到瀏覽器可視區（也是版心）的一半位置。

第②步：讓這個盒子margin-left：版心寬度的一半。這樣再往右走版心寬度的一半位置。就可以讓這個盒子貼著版心右邊對齊了。

比如下圖的示例



### （8）黏性定位（了解）

相對定位和固定定位的混合

語法： 選擇器{ position: sticky; top: 10px; }

黏性定位特點：

① 以瀏覽器可視窗口為參照點移動元素（固定定位特點）

② 黏性定位佔有原先的位置（相對定位特點）

③ 必須添加top、left、right、bottom其中一個才有效。

一般跟頁面滾動搭配使用。但因為兼容性差（IE不支持），所以不常用。

### （9）定位總結

|  |  |  |  |
| --- | --- | --- | --- |
| 定位模式 | 是否脫標 | 移動位置 | 是否常用 |
| static 靜態定位 | 否 | 不能使用邊偏移 | 很少 |
| relative 相對定位 | 否（佔有位置） | 相對於自身位置移動 | 常用 |
| absolute 絕對定位 | 是（不佔位置） | 帶有定位的父級 | 常用 |
| fixed 固定定位 | 是（不佔位置） | 瀏覽器可視區 | 常用 |
| sticky 黏性定位 | 否（佔有位置） | 瀏覽器可視區 | 當前階段少 |

① 一定記住 相對定位、固定定位、絕對定位兩個大的特點：a 是否佔位置（脫標了嗎）b 以誰為基準點移動位置。

② 重點學會子絕父相。

### （10）定位疊放次序 z-index

使用定位佈局時，可能會出現盒子重疊的情況。可以使用z-index來控制盒子的次序（z軸）

語法： 選擇器{z-index: 1;}

① 值可以是正整數，負整數，或者0。默認是auto，數值越大，盒子越靠上。

② 如果屬性值相同，則按照代碼順序，後來居上。

③ 數字後面不能加單位。

④ 只有定位的盒子才有z-index屬性。標準流和浮動沒有z-index屬性。

### （11）絕對定位的盒子居中方法

第①步： 往右走父容器寬度的50% left: 50%;

第②步： 往左走自己寬度的50% margin-left: -100px;

### （12）定位特點

絕對定位、固定定位和浮動類似。

① 行內元素添加絕對或者固定定位，可以直接設置高度和寬度。

② 塊級元素添加絕對或者固定定位，如果不給寬度或者高度，默認大小是內容的大小。

③ 脫標的盒子不會觸發外邊距塌陷。

浮動元素、絕對定位的元素、固定定位的元素都不會觸發外邊距合併的問題，因為它們都脫離了標準流。

④ 是否壓住文字和圖片

a 浮動元素只會壓住下面標準流的盒子，但不會壓住這些標準流盒子裡的文字和圖片。

浮動之所以不會壓住文字，是因為浮動產生的最初目的是為了做文字環繞效果的，文字會環繞浮動元素。

b 絕對定位和固定定位會壓住下面標準流盒子裡的所有內容。

## 元素的顯示和隱藏

### （1）display屬性

display屬性用於設置一個元素如何顯示

display: none; 隱藏元素。元素隱藏後，不再佔有原來的位置。

display: block; 除了轉換為塊級元素外，還有顯示元素的意思。

### （2）visibility可見性

visibility屬性用於指定一個元素應該可見還是隱藏。

visibility: visible; 元素可見

visibility: hidden; 元素隱藏。元素隱藏時，繼續佔有原來的位置。

### （3）overflow 溢出顯示隱藏

overflow: visible; 默認值。一個盒子大小固定，文字內容太多，超過盒子大小，默認狀態會溢出盒子全部顯示出來。

overflow: hidden; 把盒子裡溢出的部分隱藏起來。

overflow: scroll; 給盒子加滾動條，把所有的文字內容都顯示在盒子內。（不論文字是否超出盒子，都添加滾動條。）

overflow: auto; 在需要的時候（文字溢出盒子的時候），才給盒子添加滾動條。

### （4）應用案例：當鼠標經過父盒子，改變子盒子屬性

<div class=”tudou”>

<div class=”mask”></div>

<img src=”images/tudou.jpg” alt=”tudou”>

</div>

class=”mask”這個div本來是display: none;的。鼠標經過class=”tudou”這個div時，顯示class=”mask”這個div

.tudou:hover .mask {

display: block;

}

CSS進階

## 精靈圖

### （1）為什麼需要精靈圖

一個網頁中往往會應用很多小的背景圖作為修飾，當網頁中的圖像過多時，服務器就會頻繁地接收和發送請求圖片，造成服務器請求壓力過大，這將大大降低頁面的加載速度。

為了有效地減少服務器接收和發送請求的次數，提高頁面的加載速度，出現了CSS精靈技術（也稱CSS Sprites，CSS雪碧）。

核心原理：將網頁中的一些小背景圖像整合到一張大圖中，服務器只需要一次請求就可以了。

### （2）使用精靈圖

用於背景圖片。把多個小背景圖片整合到一張大圖片中。

移動背景圖片位置，用background-position

移動時沿著X和y軸移動，一般情況下是往上往左移動，其數值都是負數。

用PS來測量需要移動多少位置。在測量之前，先把圖片的圖層鎖定（在眼睛圖標後面，有個小鎖的圖標）

在PS中，選擇切片工具，選中相關圖片，在PS左下角，可以看到寬高和XY軸的數值。

## 字體圖標

### （1）為什麼需要字體圖標

字體圖標使用場景：顯示網頁中通用、常用的小圖標

精靈圖缺點：

① 圖片文件比較大

② 圖片本身放大和縮小會失真

③ 更換圖片複雜

字體圖標iconfont可以解決上述問題

字體圖標是一種高效方便的圖標使用方式，展示的是圖標，本質屬於字體。

字體圖標的優點：

① 輕量級：圖標字體比圖像小。一旦字體加載了，圖標會馬上渲染出來，減少服務器請求

② 靈活性：本質是文字，所以可以隨意改變顏色、產生陰影、透明效果、旋轉等。

③ 兼容性：幾乎支持所有瀏覽器。

注意：字體圖標不能代替精靈技術，只是對工作中圖標部分技術的提升和優化。

① 結構樣式簡單的小圖標，用字體圖標。

② 結構樣式複雜的圖片，用精靈圖。

### （2）如何使用字體圖標

① 下載字體圖標

推薦下載網站：

icomoon <https://icomoon.io/>

IcoMoon成立於2011年。推出了第一個自定義圖標字體生成器，允許用戶選擇所需要的圖標。種類繁多，全面。外國服務器，打開速度慢。

進入網站後，點擊右上角的IconMoon App紫色圖標，進入圖標挑選頁面。

在該頁面選擇需要的圖標。

如果沒找到需要的，可以在網頁最下面的Add Icons From Library…中找。

網頁最上方的工具欄裡，有編輯按鈕，可以編輯被選中的圖標。

選好所有圖標後，點擊頁面右下角Generate Font。

圖標生成好後，會進入新的頁面，在該頁面右下角Font Download。

阿里iconfont字庫

這是阿里媽媽M2UX的一個iconfont字體圖標字庫，包含了淘寶圖標和阿里媽媽圖標庫，可以使用AI製作圖標上傳生成。

② 引入字體圖標（引入到HTML頁面中）

下載後的字體是個壓縮包，將其解壓。

把fonts文件夾放到項目頁面的根目錄下。（fonts文件夾裡有4個字體文件，為了兼容不同瀏覽器）

在CSS樣式中全局聲明字體（把這些字體文件通過CSS引入頁面）

在被解壓的文件夾中，有個style.css文件，把裡面的代碼複製一下：

@font-face {

font-family: 'icomoon';

src: url('fonts/icomoon.eot?1gc3g9');

src: url('fonts/icomoon.eot?1gc3g9#iefix') format('embedded-opentype'),

url('fonts/icomoon.ttf?1gc3g9') format('truetype'),

url('fonts/icomoon.woff?1gc3g9') format('woff'),

url('fonts/icomoon.svg?1gc3g9#icomoon') format('svg');

font-weight: normal;

font-style: normal;

font-display: block;

}

把上面的代碼（叫做字體聲明）粘貼到HTML head 部分的<style></style>之間

在被解壓的文件夾裡，找到demo.html。在這裡可以找到所有下載下來的圖標的對應字符。複製一下小方塊，然後粘貼到HTML代碼相關的盒子中。



比如這個英鎊符號的小方塊放在了span裡面

然後在head 部分的<style></style>之間寫上：

span{

font-family: 'icomoon';

（還可以指定各種字體樣式，比如）

font-size: 10px;

color: pink;

}

③ 追加字體圖標

如果在工作中，原來的字體圖標不夠用了，需要添加新的字體圖標到原來的字體文件中。

打開icomoon.io主頁，點擊右上角的IcoMoon App紫色圖標。

在新頁面點擊左上角Import Icons紫色圖標。把壓縮包裡的selection.json上傳，然後選中自己想要的新圖標，重新下載壓縮包，並替換原來的文件即可。

## CSS寫三角形

寫個div，不給寬高，只給某一個邊框顏色，其它邊框顏色透明，同時設置所有邊框粗細和實線。通過有顏色邊框的方位、粗細、顏色來控制三角形的樣式。

語法如下：

div {

width: 0;

height: 0;

line-height: 0; （為了兼容低版本瀏覽器）

font-size: 0; （為了兼容低版本瀏覽器）

border: 50px solid transparent;

border-left-color: pink;

}

## 用戶界面樣式

### （1）鼠標樣式cursor

li {cursor: pointer;}

### （2）取消表單輪廓

去掉輸入框的默認藍色邊框

outline:0; 或者outline: none;

例子：

input, textarea {outline:none;}

### （3）防止拖拽文本域 resize

textarea{resize:none;}

### （4）防止文本域裡開頭出現空白區域

把textarea開頭結尾2個標識寫在一行：<textarea></textarea>。而不是分2行來寫。

## vertical-align 圖片或者以文本域為代表的表單（行內塊元素）和文字垂直對齊

用於設置一個元素的垂直對齊方式，只針對行內元素或者行內塊元素。



問題：圖片在不給寬高時，自己撐開父盒子，圖片和父盒子低端直接有個縫隙。因為盒子屬於行內塊元素，默認的vertical-align是基線對齊。

解決方法有2個：

① 給圖片添加vertical-align: middle | top | bottom等 （導師建議用這種）

② 把圖片轉換為塊級元素：img{display: block;} （塊級元素獨佔一行，可能會影響其它元素的佈局）

## 文字溢出省略號

#### ① 單行文本溢出顯示省略號

a 強制一行內顯示文本

white-space: nowrap; （默認normal，自動換行）

b 超出的部分隱藏

overflow: hidden;

c 文字用省略號代替超出的部分

text-overflow: ellipsis;

#### ② 多行文本溢出顯示省略號

多行文本溢出顯示省略號，有較大兼容性問題，適合於webKit瀏覽器或移動端（移動端大部分是Webkit內核）

使用下列代碼時，要先確定裝內容的盒子寬高。

overflow: hidden;

text-overflow: ellipsis;

/\* 彈性伸縮盒子模型顯示 \*/

display: -webkit-box;

/\* 限制在一個塊元素顯示的文本的行數 \*/

-webkit-line-clamp: 2;

/\* 設置或檢索伸縮和對象的子元素的排列方式 \*/

-webkit-box-orient: vertical;

更推薦讓後台人員來做這個效果，因為後台人員可以設置顯示多少個字，操作更簡單。

## 佈局技巧

### （1）margin負值

運用場景：在ul裡，給每個左浮動的li設置邊框1px，相鄰的邊框靠在一起，視覺上這裡的邊框就變成了2px。要避免這種情況，可以設置li{margin-left: -1px}，讓相鄰的邊框上下擺放，這樣就都是1px的邊框了。

上述場景延伸：鼠標經過某個li的時候，要顯示這個li的整個邊框。不做後續操作的時候，只能顯示上、左、下邊框，右側邊框被右邊的盒子壓住了。要解決這個問題，可以在hover裡面給當前的盒子裡：若無定位，加相對定位（保留位置）；若有定位，加z-index。

### （2）文字圍繞浮動元素巧妙運用

寫個父盒子，裡面直接放入文字，不用設置寬高，標準流。

然後放入img，給圖片添加左浮動。

文字就能自動環繞圖片。

比如做個這樣的效果圖：



代碼如下：

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <style>

      \* {

        margin: 0;

        padding: 0;

      }

      .box {

        width: 310px;

        height: 70px;

        background-color: pink;

        margin: 0 auto;

        padding: 5px;

      }

      .pic {

        float: left;

        width: 120px;

        height: 60px;

        margin-right: 5px;

      }

      .pic img {

        width: 100%;

      }

    </style>

  </head>

  <body>

    <div class="box">

      <!-- 拿盒子包住圖片是因為在後面圖片可能會被經常換掉。 -->

      <div class="pic"><img src="./th.jpg" alt="" /></div>

      <p>隐隐飞桥隔野烟，石矶西畔问渔船。 桃花尽日随流水，洞在清溪何处边。</p>

    </div>

  </body>

</html>

### （3）行內塊巧妙運用

需求：頁碼在頁面中間顯示



實現方法：把這些鏈接盒子轉換為行內塊元素，然後給父盒子指定text-align:center;

代碼：

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <style>

      \* {

        margin: 0;

        padding: 0;

      }

      .box {

        text-align: center;

      }

      .box a {

        display: inline-block;

        width: 36px;

        height: 36px;

        background-color: #f7f7f7;

        border: 1px solid #ccc;

        text-align: center;

        line-height: 36px;

        text-decoration: none;

        color: #333;

        font-size: 14px;

      }

      .box .prev,

      .box .next {

        width: 85px;

      }

      .box .current,

      .box .elp {

        background-color: #fff;

        border: none;

        /\* 或者border: 0; \*/

      }

      .box input {

        height: 36px;

        width: 45px;

        border: 1px solid #ccc;

        outline: none;

      }

      .box button {

        width: 60px;

        height: 36px;

        background-color: #f7f7f7;

        border: 1px solid #ccc;

      }

    </style>

  </head>

  <body>

    <div class="box">

      <a href="#" class="prev">&lt;&lt;上一頁</a>

      <a href="#" class="current">2</a>

      <a href="#">3</a>

      <a href="#">4</a>

      <a href="#">5</a>

      <a href="#">6</a>

      <a href="#" class="elp">...</a>

      <a href="#" class="next">&gt;&gt;下一頁</a>

      到第

      <input type="text" />

      頁

      <button>確定</button>

    </div>

  </body>

</html>

上述代碼寫出來後，做出來的頁碼在頁面居中顯示：



### （4）三角形強化



用代碼寫出如下圖像：



<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <style>

      .box1 {

        width: 0;

        height: 0;

        /\* 只保留右边的边框有颜色 \*/

        border-color: transparent red transparent transparent;

        /\* 边框式样都是solid \*/

        border-style: solid;

        /\* 上边框宽度大，右边框宽度稍小，其余边框为0 \*/

        border-width: 100px 50px 0 0;

      }

      .price {

        width: 160px;

        height: 24px;

        line-height: 24px;

        border: 1px solid red;

        margin: 0 auto;

      }

      .current-price {

        position: relative;

        float: left;

        width: 90px;

        height: 100%;

        background-color: red;

        text-align: center;

        color: #fff;

        font-weight: 700;

        margin-right: 8px;

      }

      /\* 用来做三角形 \*/

      .current-price i {

        position: absolute;

        right: 0;

        top: 0;

        width: 0;

        height: 0;

        border-color: transparent #fff transparent transparent;

        border-style: solid;

        border-width: 24px 10px 0 0;

      }

      .original-price {

        font-size: 12px;

        color: gray;

        text-decoration: line-through;

      }

    </style>

  </head>

  <body>

    <div class="box1"></div>

    <div class="price">

      <span class="current-price">

        $1650

        <i></i>

      </span>

      <span class="original-price">$5650</span>

    </div>

  </body>

</html>

### （5）CSS初始化

不同瀏覽器對有些標籤的默認值是不同的，為了消除不同瀏覽器對HTML文本呈現的差異，照顧瀏覽器的兼容，要對CSS初始化。

每個網頁都必須進行CSS初始化。

以京東CSS初始化代碼為例。

img{

border: 0; （照顧低版本瀏覽器，如果圖片外面包含了連接，會有邊框的問題。）

vertical-aligh: middle; （解決圖片底部有空白縫隙的問題）

}

body{

-webkit-font-smoothing: antialiased; （文字放大的時候，有時有鋸齒形。這個CSS3特性可以抗鋸齒形，讓文字顯示得更加清晰）

}

/\* 把我们所有标签的内外边距清零 \*/

\* {

  margin: 0;

  padding: 0;

}

/\* em 和 i 斜体的文字不倾斜 \*/

em,

i {

  font-style: normal;

}

/\* 去掉li 的小圆点 \*/

li {

  list-style: none;

}

img {

  /\* border 0 照顾低版本浏览器 如果 图片外面包含了链接会有边框的问题 \*/

  border: 0;

  /\* 取消图片底侧有空白缝隙的问题 \*/

  vertical-align: middle;

}

button {

  /\* 当我们鼠标经过button 按钮的时候，鼠标变成小手 \*/

  cursor: pointer;

}

a {

  color: #666;

  text-decoration: none;

}

a:hover {

  color: #c81623;

}

button,

input {

  /\* "\5B8B\4F53" 就是宋体的意思 这样浏览器兼容性比较好。如果写汉字，在很多浏览器里会是乱码显示。 \*/

  /\* 黑体 \9ED1\4F53

     宋体 \5B8B\4F53

     微软雅黑 \5FAE\8F6F\96C5\9ED1      \*/

  font-family: Microsoft YaHei, Heiti SC, tahoma, arial, Hiragino Sans GB,

    "\5B8B\4F53", sans-serif;

}

body {

  /\* CSS3 抗锯齿形 让文字显示的更加清晰 \*/

  -webkit-font-smoothing: antialiased;

  background-color: #fff;

  font: 12px/1.5 Microsoft YaHei, Heiti SC, tahoma, arial, Hiragino Sans GB,

    "\5B8B\4F53", sans-serif;

  color: #666;

}

.hide,

.none {

  display: none;

}

/\* 清除浮动 \*/

.clearfix:after {

  visibility: hidden;

  clear: both;

  display: block;

  content: ".";

  height: 0;

}

.clearfix {

  \*zoom: 1;

}

CSS3

## 圓角邊框

語法：

border-radius: length;

原理：length是一個圓的半徑長度，用於和邊框相切。切好了時，邊框就沿著圓弧走，不沿著邊框走了。如下圖，其它3個角同理。



length可以是數值，單位px；也可以是百分比。

把正方形設置為圓，把length修改為邊框一半，或者寫50%。

為長方形設置圓角矩形，把length改為高度的一半。

border-radius:後面可以寫4個值，分別代表左上角、右上角、右下角、左下角的圓半徑。

分開寫：border-top-left-radius，border-top-right-radius，border-bottom-left-radius，border-bottom-right-radius

兼容性：ie9+瀏覽器支持。不影響頁面佈局，可以放心使用。

## 盒子陰影

語法：

box-shadow: h-shadow v-shadow blur spread color inset;

|  |  |
| --- | --- |
| **值** | **描述** |
| h-shadow | 必須。水平陰影的位置。允許負值。 |
| v-shadow | 必須。垂直陰影的位置。允許負值。 |
| blur | 可選。模糊距離。（陰影是虛的還是實的） |
| spread | 可選。陰影的尺寸。（陰影大小） |
| color | 可選。陰影的顏色。可以寫半透明的陰影rgba(0, 0, 0, 0.3) |
| Inset | 可選。將外部陰影（outset）改為內部陰影。 |

注意：

默認的是外陰影（outset），但是不可以寫這個單詞，否則造成陰影無效。

盒子陰影不佔空間，不會影響其它盒子排列。

例子： div { box-shadow: 10px 10px 5px #888888; }

CSS3新特性

新增的CSS3特性有兼容性問題，ie9+才支持

移動端支持優於PC段

不斷改進中

應用相對廣泛

現階段主要學習：新增選擇器，盒子模型，其它特性。

## CSS3新增選擇器

### （1）屬性選擇器

屬性原則器根據元素特定屬性來選擇元素，這樣可以不用藉助於類或者id選擇器。

屬性選擇器，類選擇器，偽類選擇器，權重為10.

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>CSS3新增屬性選擇器</title>

    <style>

      /\* 1. 必須是input，同時具有value這個屬性 \*/

      input[value] {

        color: pink;

      }

    </style>

  </head>

  <body>

    <!-- 1. 利用屬性選擇器可以不用藉助於類或者id選擇器 -->

    <input type="text" value="請輸入用戶名" />

    <input type="text" />

  </body>

</html>

下面這個是重點

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>CSS3新增屬性選擇器</title>

    <style>

      /\* 2. 必須是input，同時value屬性的值為text \*/

      input[type="text"] {

        color: pink;

      }

    </style>

  </head>

  <body>

    <!-- 2. 屬性選擇器還可以選擇屬性=值的某些元素 -->

    <input type="text" /><input type="password" />

  </body>

</html>

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>CSS3新增屬性選擇器</title>

    <style>

      /\* 把class='icon'開頭的全部選出來 \*/

      /\* 首先是div，然後具有class屬性，並且屬性值必須是icon開頭的這些元素 \*/

      div[class^="icon"] {

        color: pink;

      }

    </style>

  </head>

  <body>

    <!-- 3. 屬性選擇器可以選擇屬性值開頭的某些元素 -->

    <div class="icon1">小圖標1</div>

    <div class="icon2">小圖標2</div>

    <div class="icon3">小圖標3</div>

    <div class="icon4">小圖標4</div>

    <div>醬油div</div>

  </body>

</html>

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>CSS3新增屬性選擇器</title>

    <style>

      section[class$="data"] {

        color: blue;

      }

      /\*5.  只要class這個屬性裡，有ico這個值，就都選出來 \*/

      section[class\*="ico"] {

        font-weight: 700;

      }

    </style>

  </head>

  <body>

    <!-- 4. 屬性選擇器可以選擇屬性值結尾的某些元素 -->

    <section class="icon1-data">清風不識字</section>

    <section class="icon2-data">何故亂翻書</section>

    <section class="icon3-ico">故國有明</section>

  </body>

</html>

### （2）結構偽類選擇器

結構偽類選擇器屬於偽類選擇器。

屬性選擇器，類選擇器，偽類選擇器，權重為10。

常用於根據父級選擇子元素

|  |  |
| --- | --- |
| 選擇符 | 簡介 |
| E:first-child | 匹配父元素中的第一個子元素 |
| E:last-child | 匹配父元素中最後一個子元素 |
| E:nth-child(n) | 匹配父元素中的第n個子元素   * n可以是數字，從1開始計數。 * n可以是even（偶數），odd（奇數） * n可以是公式：常見的公式如下（如果n是公式，則從0開始計數。但是第0個元素或者超出了元素的個數會被忽略。）   + 比如 ol li:nth-child(n) {background-color: pink;}把所有的子元素都選出來了。   + 2n 偶數   + 2n+1 奇數   + 5n 5 10 15……   + n+5 從第5個開始（包含第5個）到最後   + -n+5 前5個（包含第5個） |
| E:first-of-type | 指定類型元素的第1個 |
| E:last-of-type | 指定類型元素的最後一個 |
| E:nth-of-type(n) | 指定類型元素的第n個 |

例子：

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <style>

      /\* 1. 選擇ul裡面的第1個是li的孩子 \*/

      ul li:first-child {

        background-color: pink;

      }

      /\* 2. 選擇ul裡面第3個是li的孩子 \*/

      ul li:nth-child(3) {

        background-color: skyblue;

      }

    </style>

  </head>

  <body>

    <ul>

      <li>第1個li</li>

      <li>第2個li</li>

      <li>第3個li</li>

      <li>第4個li</li>

      <li>第5個li</li>

      <li>第6個li</li>

      <li>第7個li</li>

      <li>第8個li</li>

    </ul>

  </body>

</html>

#### nth-child和nth-of-type的區別

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Document</title>

    <style>

      /\* nth-child會把所有的子元素都排列序號 \*/

      section div:nth-child(1) {

        /\* 會先匹配:nth-child(1)，看後再匹配div。發現沒有這樣的div，所以無從修改樣式。 \*/

        background-color: pink;

      }

      /\* nth-of-type會把指定元素的盒子排列序號 \*/

      /\* 這個例子中，指定的元素盒子是div。這裡會先找section中的所有div，然後把第1個div選出來。 \*/

      section div:nth-of-type(1) {

        background-color: skyblue;

      }

    </style>

  </head>

  <body>

    <section>

      <p>醉眠海棠</p>

      <div>天朗氣清</div>

      <div>惠風和暢</div>

    </section>

  </body>

</html>

### （3）偽元素選擇器

偽元素選擇器：利用CSS創建新標籤元素，而不需要HTML標籤，從而簡化HTML結構。

::before 在元素內部的前面插入內容

::after 在元素內部的後面插入內容

before和after創建一個元素，屬於行內元素。

新創建的元素在文檔樹中不存在，所以稱為偽元素。

語法： element::before{}

before和after必須有content屬性

偽元素選擇器和標籤選擇器一樣，權重為1.

CSS書寫順序



例子：



切圖

## 常見圖片格式

* jpg（=JPEG）對色彩的信息保留較好，高清，顏色較多，產品類的圖片經常用jpg格式。
* gif 最多只能存儲256色，所以通常用來顯示簡單圖形及字體。可以保存透明背景和動畫效果，實際經常用於一些圖片小動畫效果。
* png 新興格式，結合了jpg和gif的優點，能夠保存透明背景。如果要切成背景透明的圖片，用png格式。
* psd 是Photoshop專用格式

PS常見貼圖方式：圖層切圖，切片切圖，PS插件切圖。

## 圖層切圖

右擊圖層 → 快速導出為PNG

同時導出2個圖層：

按住shift鍵，把2個圖層都選中→右擊圖層 → 快速導出為PNG

## 切片切圖

選擇切片工具選中圖片→文件菜單→導出→存儲為web設備所用格式→選擇圖片格式→存儲→選擇“選中的切片”來保存

如果要切透明背景的圖，先把PS圖層中的“背景”前面的小眼睛去掉（圖層不要包含背景），然後再開始上面的過程。

## Cutterman插件切圖

安裝cutterman時，PS必須是完整版，而非綠色版。查看PS是否為完整版：看窗口→拓展功能是否正常

打開cutterman：窗口→擴展功能→cutterman 切圖神器

可以導出單個圖層，也可以導出多個圖層，還可以導出選中的切片

工具

## PS

打開標尺：Ctrl+R或者 選擇 視圖→標尺。

右擊標尺，把標尺單位改為像素。

Ctrl+加號（+）放大視圖，Ctrl+減號（-）縮小視圖。

在右邊工具欄選擇矩形選框工具，可以測量大小。

Ctrl+D可以取消選區，或者在旁邊空白處點擊一下，也可以取消選區。

吸管工具可以取色。

## Snipaste

① F1截圖，測量大小，在圖上做標記，書寫文字。

② F3在桌面置頂顯示。

③ 點擊圖片，alt可以取色（按下shift可以切換取色模式）。

④ 按下esc取消圖片顯示。

項目——學成網

## 佈局

1. 確定頁面版心（可視區）。

2. 確定行模塊，以及每個行模塊中的列模塊。

列模塊經常浮動佈局。先確定每個列的大小，之後確定列的位置。

3. 製作HTML結構。

4. 運用盒子模型原理，CSS美化頁面。

## header要點



寫3（首頁、課程、職業規劃）的時候，在實際開發中，不會直接用a，而是用li包含a的做法，原因有二：

① li+a語義更清晰，一看就是有條理的列表型內容。

② 如果直接用a，容易被搜索引擎判定為堆砌關鍵字（搜索引擎對鏈接很敏感），進而被搜索引擎降低被搜索到的權重，進而影響網站排名。

## banner要點

做banner左邊的課程菜單（subnav）時遇到一個疑問，答案在於【嵌套塊元素垂直外邊距的塌陷】和【相鄰塊元素垂直外邊距的合併】



.banner .subnav li {

  margin: 30px 0 0 22px;

  line-height: 14px;

  font-size: 14px;

}

.header {

  height: 42px;

  margin: 30px auto; /\*此地层叠w里面的margin\*/

}

疑問：去掉subnav的左浮動時，第1個li的上邊距直接頂在了header中學成在線logo的下面。

原因：

①【嵌套塊元素垂直外邊距的塌陷】3次：li的上邊距30px變成了ul的，然後變成了w的，然後變成了banner的。

②【相鄰塊元素垂直外邊距的合併】：header下邊距是30px，banner上邊距是30px，合併成了30px

## 精品推薦要點

這裡10個圖是10個li。每個li會有個右邊距，除了最右邊的那些li右邊距為0。寫樣式的時候，給每個li加個margin-right: 15px。不必把最右邊的margin-right設為0。可以把ul寫寬些，寫得超出版心的寬度。ul的寬度=（li的寬度+margin-right的值）X每一行li的個數。

即：（228px+15px）\*5=1215px。

