



張睿玓

基礎前端《壹》

第一版

前言.....	5
講義編寫原因.....	5
關於程式碼編號.....	5
壹、簡略介紹瀏覽器	6
貳、概述前端	7
貳-1 前端是什麼？	7
貳-2 前端如何構建	7
參、前端網頁 《HTML》	8
肆、前端網頁 《JavaScript》	9
肆-1 JavaScript的前世今生	9
肆-2 JavaScript 能使用在哪裡？	9
BF-1-E-1 範例程式碼 《§4-2 練習模板》	10
肆-3 JavaScript的變數型態	11
Number	11
String	11
Boolean	12
Null	12
Undefined	12
Object	12
BigInt	13
Symbol	13
肆-4 風格指南及變數命名法	13
變數命名法.....	13
風格指南.....	14
肆-5 變數宣告及作用域.....	14

var	14
不使用宣告關鍵字	15
let.....	16
const	16
附錄壹、推薦IDE	19
1. HBuilder.....	19
2. Brackets	19
3. WebStorm.....	19
4. Atom	20
5. 令人著迷的插件「TabNine」	20

前言

講義編寫原因

Web技術近來十分火紅

關於程式碼編號

本講義中不論是HTML,JavaScript,甚至是CSS檔案我都會加以編號，以利整理。

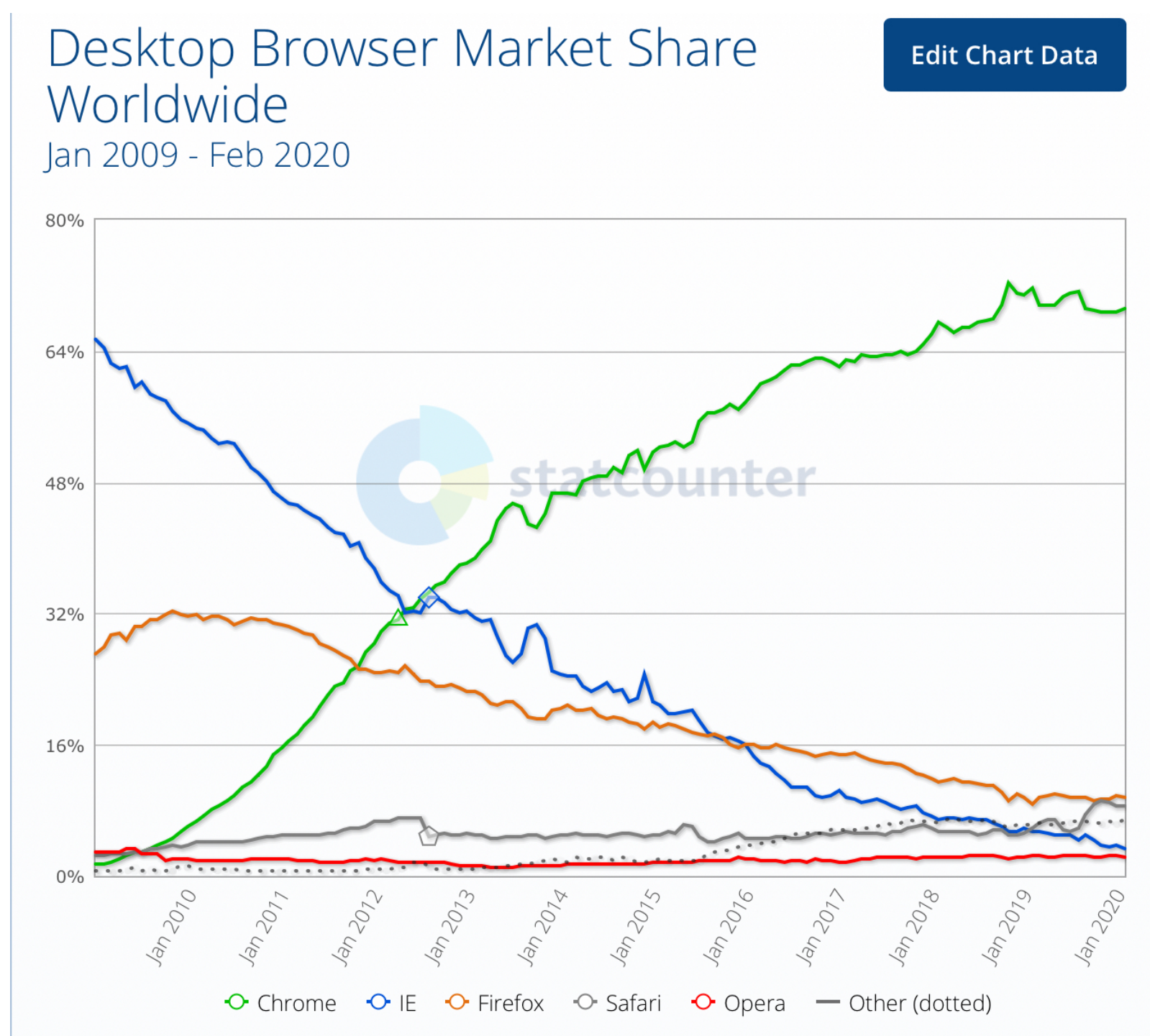
編號格式應為 「書籍-版本-程式類型-第幾個」

程式類型： C講義內需要，E範例程式碼

壹、簡略介紹瀏覽器

自從有網際網路及網頁以來，瀏覽器就是我們與「視覺化」網路連結的窗口，而瀏覽器則順勢發展，百家爭鳴，自一開始的IE稱霸市場，到現在Chrome引領全球，網路不停的在進步中，任何工具隨時都可能被淘汰。

下面這張圖是Statcounter製作的調查，從2010至現在2020瀏覽器的市佔率有怎麼樣的變化。



我們可以發現，網際網路發展初期，是由令人詬病的IE稱霸了市場，但等到Chrome出現後，Chrome的市佔率是節節攀升，而IE則像溜滑梯般直直落，昔日輝煌的IE現在卻只能當一個用來下載其他瀏覽器的軟體，真是諷刺丫！

貳、概述前端

貳-1 前端是什麼？

前端？那有後端、中端、北端、南端……嗎？所謂網頁前端是指一個網頁的組成包含兩個部分：「前端」與「後端」，那麼怎麼判別呢？前端就是你能在瀏覽器看到、取得的一切東西，像是描述結構的HTML，描述樣式、美化的CSS，讓網頁活起來的JavaScript都是前端的語言。而後端呢？我們舉網路上常看到的PHP為例，PHP是一種後端語言，正常來說，你無法取得它原始的程式碼，例如網站上的login.php你雖然都能看到其中有什麼樣的結構、長什麼樣子，但你不知道，在網頁的背後還有一大堆PHP程式碼，用來連接資料庫、寫入資訊等等，而在此講義，我們偏向教學的是前端的程式語言，當你把前端摸熟後，可以嘗試玩玩後端，會有完全不一樣的感受喔！

貳-2 前端如何構建

前端「目前」包含了下列這三樣程式語言

- HTML 宣告網頁結構
- JavaScript 與DOM、Web API互動
- CSS 宣告網頁樣式表

注意：這些都是「目前」的技術，誰也不知道什麼時候會有新技術淘汰舊技術。

藉由這三種語言相互配合，即可做出一個有模有樣的網頁了！

參、前端網頁 《HTML》

肆、前端網頁 《JAVASCRIPT》

肆-1 JAVASCRIPT的前世今生

JavaScript 是一個基於原型的物件導向程式語言，他是由Netscape所開發，到了1996年，Netscape 向ECMA提交新語言標準，ECMA也制定了ECMAScript標準，從此之後JavaScript就一直採用著ECMAScript這個標準。

現今瀏覽器支援度最廣的莫過於ECMAScript3標準而近日最多人熟知的是ES6(ES2016)這個新版本標準(雖然有更新的，但這個版本算是支援度廣的了) 這個標準加入了許多十分便利的功能，像是「Array.prototype.forEach」或「fetch」大大改進了JavaScript的方便性。

肆-2 JAVASCRIPT 能使用在哪裡？

JavaScript 能在下列這些地方使用

1. HTML的<script> 標籤中
2. javascript: 協定的連結
3. 另外寫成檔案(.js) 並用<script> 標籤的src屬性載入
4. 部分標籤的 「on+事件」 屬性

真的不得不說，JS與Html是配合的很好的

現在，請看一下下一頁的範本檔案，我們將說明如何使用它來執行JavaScript。

本章節【肆-2】之範例程式碼為《BF-1-E-1》

BF-1-E-1 範例程式碼 《§4-2 練習模板》

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>§4-2 練習模板</title>
    <script type="text/javascript">
      //下面可以寫程式

    </script>
    <script src=""></script>
  </head>
  <body>
    <a href="javascript">連結</a>
    <button onclick="">按鈕</button>
  </body>
</html>
```

這個範例程式碼是一個HTML檔案

(詳情可見【參、前端網頁〈HTML〉】)

你需要做的是將程式碼貼近你常使用的IDE或文字編輯器，將其存為「.html」檔案

(關於推薦的IDE可以參考【附錄壹、推薦IDE】)

肆-3 JAVASCRIPT的變數型態

JavaScript 是一個弱型別的語言，但他還是有幾種變數類型（雖然說到最後還是全部都是物件），我們來看看有哪幾種吧！

根據ECMAScript標準，定義了八種類型

在這本講義中，我們只會使用到基礎的類型（下面的A-F），另外的BigInt、Symbol可能等以後再研究吧！

NUMBER

首先，先講JavaScript的Number型態。

大家寫過C++，知道C++中，關於「數字」的型態有很多，像是「int,long long int,double,short」等等的，在JavaScript 中只需要一個Number就可以代表一切。

Number可以表示 $-2^{53}-1$ 到 $2^{53}-1$ 之間的數字（包括小數），另外，除了一般數字外，還包括三種特別的值「+Infinity，-Infinity，NaN」，分別是正無限大、負無限大、非數字。

NaN是一個十分特別的值，讓我們看看下面這個程式碼，大家猜猜輸出會是什麼呢？

```
console.log(('b'+ 'a'+ 'a'*10+ 'a').toLowerCase())
```

(BF-1-C-1)

答案：_____（請自行填入）

STRING

string和C++的string 是一模一樣的，都是字串，有一點需要注意，JavaScript 中沒有char的概念，每一個在string中的字元都還是string。另外一個需要注意的坑是，JavaScript 中你不能透過類似str[2] = "H"這樣修改，只能透過replace,concat,substr等等的String原型方法修改。

BOOLEAN

boolean等價於C++ bool，沒有什麼好說的，就是true和false。

貼心小提醒：0為false、非0為true喔！（在JavaScript中null,NaN,undefined,空字串也都是false）

NULL

Null 「這裡該有值，卻沒有值」

UNDEFINED

未定義！當你存取一個物件中未定義的屬性會回傳undefined。

OBJECT

物件是JavaScript 中最重要的型態，因為萬物皆是物件，上面提到的所有型態也都是物件，function也是一種物件。

參考下面的程式碼就可以創造一個「物件」

```
var me = {  
  ownedCars:null,  
  age:15,  
  birthday:new Date("2004-11-17T19:00:00"),  
  schools: [  
    "TNFSH",  
    "NNKIEH"  
  ]  
}
```

這段程式碼會建立一個名叫「me」的物件，你可以看到其中有許多的「屬性」像是「ownedCars,age,birthday」等。要如何存取這些屬性呢？我們可以用「.」或「[]」來存取，如果我今天要「年齡」，我就會寫「me.age」或「me["age"]」這樣就能取得年齡了！

另外一點提醒：在JavaScript 中，「屬性」定義時不一定要加上「"」，直接寫即可。

BIGINT

略

SYMBOL

略

肆-4 風格指南及變數命名法

變數命名法

常見的變數命名法有三種

- 蛇形命名法

蛇形命名法是C++中十分常見的命名法，std許多函式皆是使用這種命名法命名，像是「make_pair」、「sync_with_stdio」等等，這種命名法的特點是將不同單字使用下底線（_）連接，因為長得像蛇故得其名。

- 駝峰式命名法

駝峰式命名法是在JavaScript 中常見的，像是Array的原型方法中有一個「indexOf」方法，再舉一個更常見的例子—操作DOM的函式「document.getElementById」、「document.getElementsByTagName」等等的，皆是使用駝峰式命名法，因此在Javascript 中建議大家使用此命名法來為變數命名。

- 匈牙利命名法

匈牙利命名法的習慣是在變數名稱之前加上該變數之型態，讓人能一目了然該變數的內容。

風格指南

接下來是JavaScript 的各種程式風格指南整理，雖然是有這些指南，但究竟是有多少人遵守（？）

1. AirBnB的風格指南 <https://github.com/airbnb/javascript/blob/master/README.md>
2. Google的風格指南 <https://google.github.io/styleguide/jsguide.html>
3. jQuery的風格指南 <https://contribute.jquery.org/style-guide/js/>
4. NPM的風格指南 <https://docs.npmjs.com/misc/coding-style.html>

（是說我想寫怎樣就怎樣xd，標準重要嗎？）

肆-5 變數宣告及作用域

讀到這裡時，建議大家先喝杯茶，放鬆一下心情，把腦袋淨空，等等你就會看到程式最不可思議之處。

在JavaScript 中有四種變數宣告方式

VAR

先講var這個最邪教的變數宣告法。

用var宣告變數是自JavaScript 誕生以來就如此了，但它具有非常奇葩的作用域，我們來深入探討吧！

以「是否在Function內」為界，如果今天你在函數「外」宣告，變數將會把自己納入global物件（在前端叫做window）下成為它的屬性，成為全域變數。而如果今天在函數「內」宣告，會成為區域變數，他的作用範圍就僅僅在函數內，函數外是取不到它的值的。

但注意一點：是「函數」喔，今天如果我們是用for迴圈可就不是如此了，看看下一頁的例子

```
for(var i=0;i<10;i++){  
  console.log(i)
```

(BF-1-C-2)

答案：_____（請自行填入）

大家在這裡可以打開Chrome自己練習看看，試試看JavaScript 的語法。

筆記欄🖋️

不使用宣告關鍵字

沒錯，你眼睛不是壞掉，在JavaScript 中你可以不用宣告關鍵字就宣告一個變數。

但注意一點，若宣告「不使用宣告關鍵字」的變數，會自動變成全域變數，等同是不論在函數內或外宣告都可以存取到該變數。

因為這種奇怪的宣告，建議非必要不要偷懶乖乖寫var比較好。

LET

接下來的let和const都是ES6的新語法，解決var的奇怪問題，還變數一個乾淨的空間。

let宣告的變數是以「區塊 (block)」為界，什麼是區塊呢？簡單來說就是由{}括起來的程式，像是迴圈、函數、if判斷，let並不會在全域物件建立一個屬性，其實就是var的加強版，拿掉var的各種奇葩處。

注意：前面講的var是「可以」被重複宣告的，而let若在同一區塊則是「不能」重複宣告，若重複宣告會跳出Syntax Error

那麼到底實際上let與var有什麼差呢？MDN做了解釋：「let 可以宣告只能在目前區塊、階段或表達式中作用的變數。而 var 則是定義了一個全域變數，或是在整個 function 而不管該區塊範圍。」

CONST

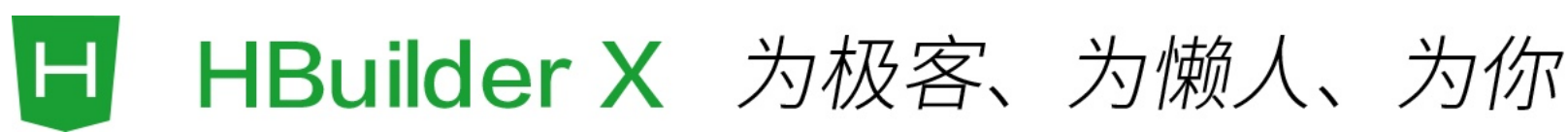
附錄壹、推薦IDE

在寫程式時，IDE是開發者的絕佳助手，雖然說前端的開發根本不需要編譯（純HTML,CSS,JS）但一個好的IDE能讓你在開發時快上數倍，也能寫出較為整齊，可讀性較高的程式。

關於前端IDE我推薦下列幾種（依照好用程度排列，1為最推薦）

1. HBuilder

HBuilder: <https://www.dcloud.io/hbuilderx.html>



HBuilder是由DCloud製作的一個十分優秀的IDE，體積小、開啟快速，功能也十分齊全，還能透過「插件安裝」的功能，將HBuilder改造成更強大的工具。

對於Vue.js框架，開發起來是十分方便的，整合的非常好，還能透過其內建的App封裝功能直接發布為iOS,Android的App，一份程式碼能在好幾個地方跑。是一個十分優秀的工具。

支援Git,支援Live Preview 幾乎其他IDE擁有的特點他都擁有了。

2. Brackets

Brackets: <http://brackets.io/>

Brackets是一個由Adobe開發的開源IDE，豐富的插件讓Brackets能夠變成一個好用的IDE

Brackets最吸引人的特點是Live Preview當你在編輯器上修改時，對應的瀏覽器頁面也會顯示更新的內容。

3. WebStorm

WebStorm: <https://www.jetbrains.com/webstorm/>

WebStorm是一個由知名IDE開發廠商JetBrains製作的網頁開發IDE，繼承了IntelliJ Idea對於JavaScript的強大功能，且是一個十分完全的IDE，缺點就是他的體積太大，十分佔空間又耗資源，對於小型電腦不是那麼適合。

WebStorm是一套付費軟體，但對於學生，我們有教育方案可以使用，申請一個JetBrains的教育帳號就能免費授權了。

4. Atom

Atom: <https://atom.io/>

Atom是一個由Github開發的IDE，體積小，打開就是編輯器。

推薦他的理由是可擴充性非常大。且有一大堆插件能使用，對於Git的整合十分好，大型程式在製作時若需要版本控制可以使用它。

像我們在寫程式競賽時，若安裝兩個插件「PDF Viewer」、「C++ Compiler」瞬間Atom就可以變成一個厲害的C++IDE，可以推廣到其他狀況，想做什麼，安裝插件就行了。

5. 令人著迷的插件「TABNINE」

他不是IDE啦，只是推薦一下這個十分邪教的TabNine，你可以在Atom、VS Code、IntelliJ 平台、Vim、Sublime Text等等的IDE安裝他，他能幹嘛呢？當你輸入文字時，他會幫你「自動補齊」利用AI，比對每個工程師常用的語法，為你自動補齊，是一個超級方便的插件

官網: <https://tabnine.com/>

