JavaScript導論

JavaScript包括：JavaScript基礎語法，WebAPI（DOM操作），JavaScript面向對象

jQuery基於JavaScript封裝。

JavaScript基礎語法

## 1. 編程語言

### （1）編程

編程：讓計算機為解決某個問題而使用某種程序設計語言編寫程序代碼，並最終得到結果的過程。

計算機程序：計算機所執行的一系列的指令集合。

這裡所定義的計算機是任何能夠執行代碼的設備，可能是智能手機、ATM機、服務器等。

### （2）計算機語言

計算機語言指用於人與計算機之間通訊的語言，它是人與計算機之間傳遞信息的媒介。

計算機語言種類非常多，總的來說可以分成機器語言，彙編語言和高級語言三大類。

計算機最終所執行的都是機器語言，它是由0和1組成的二進制數，二進制是計算機語言的基礎。

### （3）編程語言

可以通過類似於人類語言的“語言”來控制計算機，讓計算機為我們做事情，這樣的語言就叫做編程語言（Programming Language）。

編程語言是用來控制計算機的一系列指令，它由固定的格式和詞彙（不同編程語言的格式和詞彙不一樣），必須遵守。

常用的編程語言有兩種形式：彙編語言和高級語言。

彙編語言和機器語言實質是相同的，都是直接對硬件操作，只不過指令採用了英文縮寫的標識符，容易識別和記憶。

高級語言主要是相對於低級語言而言，它並不是特指某一種具體的語言，而是包括了許多編程語言，常用的有C語言、C++、Java、C#、Python、PHP、JavaScript、Go語言、Objective-C、Swift等。

### （4）翻譯器

高級語言所編制的程序不能直接被計算機識別，必須經過轉換才能被執行，為此，我們需要一個翻譯器。翻譯器可以將我們所編寫的源代碼轉換為機器語言，這也被稱為二進制化，記住1和0。



### （5）編程語言和標記語言區別

編程語言有很強的邏輯和行為能力。在編程語言裡，有很多if else、for、while等具有邏輯性和行為能力的指令，這是主動的。

標記語言（HTML）不用於向計算機發出指令，常用於格式化和鏈接。標記語言被計算機讀取，是被動的。

## 2. 計算機基礎

### （1）計算機組成



### （2）數據存儲

計算機內部使用二進制0和1來表示數據。

所有數據，包括文件、圖片等最終都是以二進制數據（0和1）的形式存放在硬盤中的。

所有程序，包括操作系統，本質都是各種數據，也以二進制數據的形式存放在硬盤中。平時所說的安裝軟件，就是把程序文件複製到硬盤中。

硬盤、內存都是保存的二進制數據。

### （3）數據存儲單位

bit < byte < kb < GB < TB…

位（bit）：1bit可以保存一個0或者1（最小的存儲單位）

字節（Byte）：1B=8b

千字節（KB）：1KB=1024B

兆字節（MB）：1MB=1024KB

吉字節（GB）：1GB=1024MB

太字節（TB）: 1TB=1024GB

……

### （4）程序運行



step① 打開某個程序時，先從硬盤中把程序的代碼加載到內存中

step② CPU執行內存中的代碼

需要內存的原因，CPU運行速度非常快，如果只從硬盤中讀取數據，會浪費CPU性能，所以使用存取速度更快的內存來保存運行時的數據。（內存是電，硬盤是機械。）

## 3. 初識JavaScript

### （1）JavaScript歷史

布蘭登·艾奇 Brendan Eich （1961年出生）在1995年用10天設計了JavaScript。

網景公司最初將其命名為LiveScript，後來與Sun（famous for Java）合作之後將其改名為JavaScript

JavaScript和Java沒有關係。

### （2）JavaScript是什麼

JavaScript是世界上最流行的語言之一，是一種運行在客戶端的腳本語言（script是腳本的意思）。

腳本語言：不需要編譯，運行過程中由js解釋器（js引擎）逐行來進行解釋並執行。

現在也可以基於Node.js技術進行服務器端編程。



客戶端（自己的電腦） 遠程服務器

### （3）JavaScript作用

* 表單動態校驗（密碼強度檢測）（JS產生最初的目的）
* 網頁特效
* 服務端開發（Node.js）
* 桌面程序（Electron）
* App（Cordova）
* 控制硬件-物聯網（Ruff）
* 遊戲開發（cocos2d-js）

### （4）HTML / CSS / JS 的關係

HTML決定網頁結構和內容（決定看到什麼），相當於人的身體。

CSS決定網頁呈現給用戶的模樣（決定好不好看），相當於給人穿衣服、化妝。

JS腳本語言（編程類語言）實現業務邏輯和頁面控制（決定功能），相當於人的各種動作。

### （5）瀏覽器執行JS

瀏覽器分成兩部分，渲染引擎和JS引擎。

渲染引擎：用來解析HTML和CSS，俗稱內核，比如Chrome瀏覽器的blink，老版本的webkit。

JS引擎：也稱為JS解釋器。用來讀取網頁中的JavaScript代碼，對其處理後運行，比如Chrome瀏覽器的V8（號稱“最快的JS引擎”）。

瀏覽器本身並不會執行JS代碼，而是通過內置JavaScript引擎（解釋器）來執行JS代碼。JS引擎執行代碼時逐行解釋每一句源碼（轉換為機器語言），然後由計算機去執行，所以JavaScript語言歸為腳本語言，會逐行解釋執行。

### （6）JS的組成



ECMAScript是由ECMA國際（原歐洲計算機製造商協會）進行標準化的一門編程語言，這種語言在萬維網上應用廣泛，它往往被稱為JavaScript或Jscript，但實際上這兩者都是ECMAScript語言的實現和拓展。



ECMAScript規定了JS的編程語法和基礎核心知識，是所有瀏覽器廠商共同遵守的一套JS語法工業標準。

更多参看MDN: [MDN手册]

<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/JavaScript_technologies_overview>

DOM（Document Object Model）（文檔對象模型），是W3C組織推薦的處理可拓展標記語言的標準編程接口。通過DOM提供的接口可以對頁面上的各種元素進行操作（大小、位置、顏色等）。

BOM（Browser Object Model）（瀏覽器對象模型），提供了獨立於內容的，可以與瀏覽器窗口進行互動的對象結構。通過BOM可以操作瀏覽器窗口，比如彈出框、控制瀏覽器跳轉、獲取分辨率等。

### （7）書寫JS

JS有3種書寫位置，分別為行內、內嵌和外部。

**① 行內JS**

可以將單行或少量JS代碼寫在HTML標籤事件屬性中（已on開頭的屬性），如：onclick。

注意單雙引號：在HTML中推薦使用雙引號，在JS中推薦使用單引號。

可讀性差，在HTML中編寫大量JS代碼時，不方便閱讀。

引號易錯，引號多層嵌套匹配時，非常容易弄混。

特殊情況下使用。

**② 內嵌JS**

可以將多行JS代碼寫到<script>標籤中。

內嵌JS式學習時常用的方式。

**③ 外部JS文件**

利於HTML頁面代碼結構化，把大段JS代碼獨立到HTML頁面之外，既美觀，也方便文件級別的複用。

引用外部JS的script標籤中間不可以寫代碼。

<script src="my.js"></script> 兩個script中間不要寫代碼。

適合於JS代碼量比較大的情況。

以下為例子。

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

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

    <title>Document</title>

    <!-- 2. 內嵌式js -->

    <script>

      alert("天邊彎月");

    </script>

    <!-- 3. 引入外部js 注意這裡是雙標籤-->

    <script src="my.js"></script>

  </head>

  <body>

    <!-- 1.行内式的js 直接写到元素的内部-->

    <input type="button" value="物華天寶" onclick="alert('人傑地靈')" />

  </body>

</html>

## 4. JavaScript注釋

    <script>

      // 單行注釋   快捷鍵 ctrl + /

      /\* 多行注釋   快捷鍵 shift + alt + a 修改為shift + ctrl + /

      多行注釋 \*/

    </script>

## 5. JavaScript輸入輸出語句

為了方便輸入輸出信息，JS提供了一些輸入輸出語句。常用語句：

    <script>

      //  給用戶的輸入框

      prompt("請輸入您的用戶名");

      // 展示給用戶看的

      alert("計算的結果是");

      // 控制台輸出 給程序員測試用的

      console.log("給程序員看的");

    </script>

## 6. 變量概述

### （1）什麼是變量

變量是用於存放數據的容器。通過變量名獲取數據、修改數據。

通俗解釋：變量是一個裝東西的盒子。

### （2）變量在內存中的存儲

本質：變量是程序在內存中申請的一塊用來存放數據的空間。

類比：變量名=酒店房間號碼，變量=酒店房間；聲明了1個變量=在酒店開了一間房。

## 7. 使用變量

使用變量有2步：①聲明變量，②賦值。

第①步：聲明變量

var age; //聲明一個名稱為age的變量。

var是一個JS關鍵字，用來聲明變量（variable）。使用該關鍵字聲明變量後，計算機會自動為變量分配內存空間。

age是程序員定義的變量名，程序員需要通過變量名來訪問內存中分配的空間。

第②步：賦值

age = 10；//給age這個變量賦值為10。

=用來把右邊的值賦給左邊的變量空間中，此處代表賦值。

變量值是程序員保存到變量空間裡的值。

8. 變量語法拓展

9. 變量命名規範

10. 交換變量案例

WebAPI（DOM操作）

jQuery

面向對象編程

ES6語法

ECharts

熟悉Echarts即可，因為有的公司用，有的公司不用。

編程邏輯思維能力

案例：攜程移動端輪播圖，京東移動端輪播圖，品優購輪播圖，基於ECharts的數據可視化項目。