## **申論題**

* 請解釋什麼是 prototype inheritance

JavaScript 物件是一動態的屬性並擁有一個原型物件的鏈結，當物件試圖存取一個物件的屬性時，其不僅會尋找該物件，並且會單向由內向外尋找該物件的原型、原型的原型。

* 請解釋什麼是 closure 及如何使用

閉包為一函式建立時，該時間和位置時的環境變數。即使創建層的函式已執行完畢並結束，但對於內部建立的函示而言，但因為閉包的機制，其原型鍊依然存在

* 請解釋什麼是 ajax

一套綜合了多項技術的瀏覽器端網頁開發技術。

* 請解釋 event bubbling

當某個事件發生在某個DOM元件上，且各交集重疊時，事件觸發的順序會先依據cature phase由外向內，再行bubbling phase由內向外傳遞。

* == 和 === 有何不同

==用於非嚴格模式下，各類資料型別作判斷式時會自行轉換，如字串和數值會自動轉換；==用於嚴格模式下，各類資料型別作判斷式時不會自行轉換，如字串和數值不會自動轉換。

* 請解釋 same-origin policy

在跨領域請求時，瀏覽器會根據通訊協定、網域名稱、埠號做被判定，只要三者其中一個不相符，即為非同源，從目標資源取得的回應（Response）就會被攔截，此機制基於安全理由，避免不同源的網站相互發出請求。

* 什麼是 single page app，並怎麼讓它對 SEO 更友善。

單頁式應用（Single Page Application），藉由抓取html的元素的方式，頁面主要由javascript程式撰寫，可動態展現當前頁面來與使用者互動，而非傳統的從伺服器重新載入整個新頁面。

## **寫程式**

* 建個數到 100 的迴圈，當數字是 3 的倍數時輸出 “fizz”，當數字是 5 的倍數時輸出 "buzz"，當數字同時是 3 與 5 的倍數時輸出 “fizzbuzz”。

for (let i = 1; i <= 100; i++) {

  if (i % 3 === 0) {

    console.log(i, "fizz");

  } else if (i % 5 === 0) {

    console.log(i, "buzz");

  } else if (i % 10 === 0) {

    console.log(i, "fizzbuzz");

  }

}

* 用程式畫一個菱形如下

\*

\*\*\*

\*\*\*\*\*

\*\*\*

\*

* 實作符合下面的函式：  
  add(2, 5); // 7  
  add(2)(5); // 7

function add(x, y) {

  console.log(x + y);

}

add(2, 5);

function add(x) {

  return function (y) {

    console.log(x + y);

  };

}

add(2)(5);

**回答問題**

* var foo = 10 + '20';  
  foo 的值是什麼？

由於再var宣告為非嚴格模式，會自動將數值轉換為字串，故兩兩字串相加結果為“1020”

* "i'm a lasagna hog".split("").reverse().join("");  
  上面的陳述式會回傳什麼？

有先字串方法會由左向右執行，故會先以空白做為區隔，將各字詞存進陣列中，並且.reverse()方法將陣列順序顛倒，最後.join("")方法將各字詞合併並且轉成字串。"goh angasal a m'i"

* ( window.foo || ( window.foo = "bar" ) );  
  window.foo 的值是什麼？

此程式再包含在一個作用域中，故在該作用域下window.foo的值為’bar’字串

* 下面的兩個 alerts 的結果會是什麼？  
  var foo = "Hello";  
  (function() {  
   var bar = " World";  
   alert(foo + bar);  
  })();  
  alert(foo + bar);

第一個alert屬於為一個IIFE是一個定義完馬上就執行的立即函式， 會直接alert出Hello World的字串，第二個alert無法找到bar的值，因此reference error

* var foo = [];  
  foo.push(1);  
  foo.push(2);  
  下面 foo.length 的值是什麼？

陣列的Push方法會針對相同的陣列進行更新，故foo 為[1, 2]陣列，長度為2

* 下面這段會印出什麼？  
  console.log('one');  
  setTimeout(function() {  
   console.log('two');  
  }, 0);  
  console.log('three');

JS引擎為JIT即時編譯方式，會先由上往下讀取，故先印出"one"，接者執行排程時將timer webapi會將其放進event queue排隊，接著繼續執行callstack印出分別會印出"two"，最後等執行完畢後會執行排程印出"three"。"one"、"three"、"two"字串