## **申論題**

* 請簡述對 Javascript 非同步概念(Asynchronous)的理解與常見情境

JavaScript在程式執行上分成了同步(Synchronous)和非同步(Asynchronous)兩種方式，由於JavaScript屬於單一執行續(Single-Thread)的語言，所以程式碼會由上往下逐行執行，也就是說當這一行還沒執行完畢時不會執行下一行，一次只能做一件事情，這樣的方式就是同步執行，若所有程式都採用同步執行很容易發生阻塞(Blocking)。

為了解決這個問題就需要採用非同步的執行方式，非同步的特點在於可以同時處理多個要求不會被阻塞，較常被用在向外部請求資料時，由於無法得知請求的時長，所以會同時進行資料請求同時執行後續的程式碼，而JavaScript能達成非同步的概念主要是利用了瀏覽器所提供的Web APIs，例如setTimeout、Fetch等等。

* 什麼是 Single Page Applications？在 SEO 上會遇到什麼問題？

Single Page Applications(SPA)為單頁式網站，也就是所有的網頁渲染過程不會涉及到網頁跳轉或是重新整理的動作，SPA的好處就在於頁面更新的方式是透過AJAX獲取資料進行局部更新，提高頁面的加載速度，提供更快更流暢的用戶體驗，由於首次連線到網頁就會下載完所有資源，減少了頻繁對server發送request的負擔。

但由於SPA大多數的資料都是透過AJAX而來，以React來說，只有一個內含<div id=”root”></div>的元素，因此會被系統判定為網頁內容空泛，不利於SEO的表現。

* 請列舉兩項最佳化網站效能與體驗的方式，並簡單說明 (不限於前端程式)。

網站的效能經常會被容量較大的檔案影響，例如較大型的圖片或是JavaScript檔案，因此可以透過lazy-loading的方式載入圖片，減少首次加載頁面所需耗費的資源，或是透過Webpack打包等方式壓縮JavaScript的檔案。

另外一個方式是透過後端資料庫的設計，例如資料庫正規化，或是在資料庫加上索引的方式降低資料獲取的時間，提高網站效能。

## **程式題**

* 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”。**注意，你只能在程式中使用一次console.log()**

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

let output = "";

if (i % 3 === 0) {

output += "fizz";

}

if (i % 5 === 0) {

output += "buzz";

}

console.log(output || i);

}

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

function add(x, y) {

if (args.length === 1) {

return function(y) {

return x + y;

}

}

return x + y;

}

**簡答題**

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

"goh angasal a m'i"

該程式碼會先將原字串的每一個字元分割成一個array後，將整體array的順序做倒轉，再將array中的物件組合成新字串，因此會回傳倒反的原字串。

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

沒有window.foo因此執行window.foo = “bar”。

* 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);  
  “Hello World”

Uncaught Reference Error: bar is not defined

var屬於function scope，因此在function外部無法取得內部宣告的變數

* var foo = [];  
  foo.push(1);  
  foo.push(2);

foo.unshift(foo.pop());

印出 foo 與 foo.length 的值分別是什麼？  
[2, 1]  
2

去除末端的2再加入foo.pop()的回傳值2到首端。

* 下面這段會印出什麼？  
  console.log('one');  
  setTimeout(function() {  
   console.log('two');  
  }, 0);  
  console.log('three');  
  ‘one’  
  ‘three’  
  ‘two’  
  setTimeout即使設定0ms執行，依然會先被放到callback queue中，所以會等到stack中的項目處理完才會執行。