申論題

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

非同步即是當一段程式在未執行完成前，可同時再去處理其他程式，像是去櫃檯辦事，等待期間可再去洗手間

最常見的像是API的發查，回傳時間會受到許多因素影響，例如伺服器地理位置，網路的訊號強度及速度等等，在等待回傳的結果期間，

如果可以利用非同步概念去完成一些不需要依回傳結果去處理的行為，能加快程式邏輯的處理

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

SPA是一頁式網頁應用程式，將所有此網站上的畫面及其元件都在同一位址顯示，在點擊畫面上的按鈕等等都能夠不需要重新讀取，再藉由ajax可更好的實現

通常SPA都是CSR，使用者體驗上雖然會比較快，但遇到SEO的問題是需要去SERVER取得資料，再取得資料回client由CSR去處理，這段等待的期間可能就會影響SEO的效能

3.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)。

一.利用Webpack將前端程式包成大小不一的bundle，能縮小整體檔案大小，亦可藉由將第三方元件放入vendor區塊來減少重複下載，甚至某些常用元件可以使用他CDN版本

二.可藉由像是Lazy Loading方式，起始只先去抓取網頁上半部資源，再藉由判斷網頁右方滾輪位置，到達指定位置才去讀取後續資料

程式題

1.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()

let i, result

for (i = 0; i <= 100; i++) {

if (i % 15 === 0) result = 'fizzbuzz'

else if (i % 5 === 0) result = 'buzz'

else if (i % 3 === 0) result = 'fizz'

else result = ''

console.log(result)

}

2.實作符合下面的函式：

add(2, 5); // 7

add(2)(5); // 7

一.const add = (a, b) => {

console.log(a + b)

}

二.const add = a => {

return b => console.log(a + b)

}

簡答題

1."i'm a lasagna hog".split("").reverse().join("");

上面的陳述式會回傳什麼？

"goh angasal a m'i"

2.( window.foo || ( window.foo = "bar" ) );

window.foo 的值是什麼？

"bar"

3.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);

一."Hello World"

二.錯誤:bar is not defined

4.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？

一.[2, 1]

二.2

5.下面這段會印出什麼？

console.log('one');

setTimeout(function() {

console.log('two');

}, 0);

console.log('three');

"one"

"three"

"two"

Figma

