1. **What is different between undefined and null？**

Undefined: when a variable is declare, but no value has been assigned. Typeof undefine is undefine

Null: null is an assign value, you can assign it to a variable. Null is similar to Boolean, string and integer. The type of null is object

1. **why use && operator**

&& is logical operators in JS, it will return the first operand that is false. If no false was found, it return the latest operand.

1. **Why use || operator**

|| is logical operator is JS, it was used to prevent some variable is undefined. It will return the first truthy operand. If no true was found, return the latest one.

1. **DOM是什么**

DOM is Documental Object Module. When browser read the HTML document, it will create a big tree structure object.

Document -> html -> head

-> body

1. **What is different between preventDefault vs. stopPropagagtion.**

* **stopPropagation:** prevent propagation of the same event from being called. Propagation means bubbling up to parent elements or capturing down to child element.
  + I have two div, div1 and div2. Div2 is div1’s child. They both has an onclick alert function. Normally, if I click div2, then div1 & div2 alert function will be triggered. If I add event.stopPropagation() in div2, then only div2 alert function will be triggered.
* **PreventDefault:** prevent default action occur, like the submit button in the form
  + Button inside the form, it’s default action will submit. If you don’t want to use it’s default action, then you preventDefault to stop it occur.

1. **Why `obj.someprop.x` will cause an error**

Const obj = {}

Console.log(obj.somprop.x)

because obj.someprop is undefined. Therefore, undefined don’t have the properties x, which will cause an error.

1. **What is different between == and ===**

==: is more used to compare the value with different data type.

===: is more used to compare the value with same data type. this one is more strict comparation.

Special case:

let a = {a: 1};

let b = {a: 1};

let c = a;

Console.log(a === b) // false

Console.log(a === c) // true

The reason is, === is more likely to compare the memory address. Variable a & b has same value, but different address. Variable a & c pointing at same memory address.

=== rule

1. Data type different, return false
2. If two variable refer same object or function, then return ture.
3. **!! operator why use it.**

!! was used to mandatory switch right side value to Boolean

1. **What is Hoisting**

Hoisting mean you can use the function or variable before declare it.

For example, I have a function call addOne, which I didn’t declare it yet. But I can still use this function before I declare. That is because function declaration will move forward and save in cache, and then call the function.

Var can hoisting, but let and const can’t

Num = 6

Console.log(num)

Var num;

1. **Explain what is the scope? What is the difference between var, let, and const in JavaScript?**

Scope is whether you have the accessibility of the variable and function. In the other word, what variables you can access to. Two kind of scope, global scope and local scope.

Global scope: if a variable is declared outside all function or curly braces { }， it is said to be defined in the global scope.

Local scope: two kinds of local scope, one is function scope, and the other one block scope.

- Function scope is declare inside the function

- Block scope is declare inside the curly braces.

**Const**: const definition, one you define, you can change it. Most used on global scope. But there is some exception, like object, array. You can still change its’ properties. Due to the properties contain another memory address that pointing at different region. If you don’t want the object or array change anymore, you can use Object.free()

**Var and let**: let is kind of update of var. let is more used on block scope, like for loop, which is a perfect sample. Var can use on global and function scope.

Var will have hoisting, which mean you can use the variable before declare it.

1. **Closures**

Closures mean the variable inside the function can be read by its child function. Each function is a close package. You can’t read the variable inside the function directly.

Why we use closure?

1. If you want to read the internal function variable, then you can return child function, which contain the function variable.

2. If you want some function variable don’t be destroy, after finish the function

1. **What is falsyValues in JS**

FalsyValue mean when you mandatory switch the data type to Boolean, falsy value will show false.

Const falsyValue = [“”, 0, null, undefined, NaN, false]

1. **What is “use strict” for?**

“use strict” module add some limitation when our coding. It can prevent some bug earlier. It can make developer easier to write secure JS.

1. **What is the keyword of “this” in JavaScript?**

This:

* In global environment, this is referred to the window, but in strict module, this will be undefined.
* This is refers to an object. Every js function while executing has a reference to its current execution context, called this.

function bike() {

console.log(this.name);

}

var name = "Ninja";

var obj1 = { name: "Pulsar", bike: bike };

var obj2 = { name: "Gixxer", bike: bike };

bike(); // "Ninja"

obj1.bike(); // "Pulsar"

obj2.bike(); // "Gixxer"

1. **What is IIFE, why use it?**

IIFE: 'Immediately Invoked Function Expression', use racket to include a function, another racket include the parameter of the function.

Like this (function (a, b) { dosomething… } ) (aVal, bVal);

Why use it?

1. A variable defined in an IIFE can’t be accessed from the outside. It can be accessed in the enclosing block, which mean is safe
2. Loop index, it could executing asynchronous tasks inside a loop. Solve the async problem without using Promise and async/await
3. **Why use Function.prototype.bind?**

**最主要的原因是，function里面通常是有this.state，如果传到别的component里面，this所指就会改变，改成当前的this. 所以要绑定一下**

This was most use in React, in class. Bind() is used to bind the function to the class. Then you can use the function by this.function(). The reason why use .bind, because what keyword “this” pointing will change in different components.

If inside the function , it don’t use any keyword “this”, then that will be fine without using Function.prototype.bind

You can use arrow function to save this step.

1. **What is High-order functions?**

Functions that taking functions as arguments or return a function. Callback function is kind of high-order function.

Like map, filter and reducer, this is js internal high-order functions.

1. **Array.prototype.map & Array.proptotype.filter & Array.proptotype.reduce**

Map() create a new array, calls the provided function once for each element in an array, in order

Filter() create a new array, filter the element not fit for functions.

Reduce() executes a provided function for each value of the array (from left-to-right).

1. **What is arrow function, when you should not use it?**

Arrow function is more concise and convenience to use compare to the regular function expression.

In classic function expression, the `this` keyword is bound to different value based on the context in which it is called. With arrow functions, you don’t need to bind, because it bind automatic.

When should use arrow function

- When you always need to bind the function to this.

- Simple callback function, which save you lots of space and readable.

1. **什么是REST**

REST is one api module. The most common use data type is JSON. RestFul API is simple, easy reading and easy testing.

1. **Different between forEach and Map**

* forEach and map support 3 parameter, item, index and arr
* map will return new array, forEach don’t return anything, undefined.
* forEach can’t use callback function, but map can.

1. **JS的垃圾回收系统**
2. **ES6导入和导出的区别**
3. **What is set, how it work**
4. **… what different between them?** rest参数和扩展运算符
5. **What is default value**
6. **What’s different between implicit conversion and explicit conversion.**
7. **What is NaN, how to check**
8. **How to check whether it is an Array**
9. **Why use New keyword?**
10. **What’s different between Object.freeze() and const**

--- 异步

1. Most frequently use request method

AXIOS: create http request from NodeJS. It support Promise API.

axios.get(***url***, {  
 headers,  
})

AJAX: Asynchronous JavaScript and XML, it use XMLHttpRequest. It can send and receive JSON, XML and HTML file. AJAX is part of JQuery. It will have the callback hell problem.

Advance:

* Don’t need to refresh the HTML page
* Quick response

$.ajax({  
 url: "save.php",  
 type: "POST",  
 data: {  
 name: ***name***,  
 email: email  
 },  
 cache: false,  
 success: function(dataResult){  
 var dataResult = ***JSON***.parse(dataResult);  
 if(dataResult.statusCode==200){  
 $('#success').html('Data added successfully !');  
 }  
 else if(dataResult.statusCode==201){  
 alert("Error occured !");  
 }  
  
 }  
});

Fetch: Fetch is used to replace ajax. It support Promise API. You can use async/await to solve callback hell problem.

fetch('https://api.github.com/orgs/nodejs', {  
 headers: new ***Headers***({  
 'User-agent': 'Mozilla/4.0 Custom User Agent'  
 })  
})  
 .then(response => response.json())  
 .then(data => {  
 ***console***.log(data)  
 })  
 .catch(error => ***console***.error(error))

1. **How to handle callback hell**

* The solution of callback hell: use Promise to encapsulate function. Resolve and reject is two callback method in the Promise. One is for success callback and the other is for failure callback. .then is for resolve function, .catch is for reject function. 使用resolve(data)就会返回这个数据，return是返回function
* And then we can use async/await method to handle promise. Remember, 1. await only return promise. 2. Await and async need to appear together.

1. **What’s the relationship between axios and promise.**

Axios is a HTTP lib that based on Promise. When you get and post using axios, it will return a Promise.

Promise is one of solution for async programing. It was used to solve callback hell problem.

1. **What’s different between Promise and Callback?**

Promise is kind of callback function, two argument of Promise, resolve and reject are two function. Resolve is successful callback, reject is failure callback.

Promise represent an async operation, use .then method to handle callback function. Promise is one of method to handle callback hell.

1. **Why JS is single thread, but it can handle asynic**

There is an example to explain this. Image JS is a person, everyday he woke up, he need to wash his face, but he also feel hungry. He opened the microwave and push the food inside the microwave. And then he went to wash his face. When he done he go back to kitchen and get and eat the food.

How JS work, we have 4 tasks, 1,2,4 are synch tasks, 3 is async task. This 4 tasks will put into queue. When the JS main thread running, the first and second tasks are sync task, then they will run one by one. After the first one finish, then the second one.

The third is async task, it will also run on JS main thread. but it need some time to get the response, therefore, JS main thread will hand it up, put it to the latest position of queue, then go to the forth tasks.

After main thread finish all the tasks, then it will check the queue whether there is some task in there. If yes, then it will run it. Otherwise, keep looping, which is event loop.

Single Threaded mean that only JS can do all the process. Once JS have async operation, then JS will let other to handle the async operation, JS will handle the result once it returns back.

1. **Where does JS execute**

Javascript is most often run-on webpages inside the browser, but it can also run server-side, which is NodeJS

1. **What’s different between Get and Post**

Get and Post is used to transfer data from client to server in HTTP protocol.

Get carries request parameter appended in URL string,

Post carries request parameter in body which makes it more secure way of transferring data from client to server

1. **Please introduce what is 304**

301-303 the website page move to new position, 304 mean website didn’t change since last request. Server won’t response webpage content

1. **What is callback function**

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some action

1. **What is promise**

It is kind of callback function. Promise is used to solve async coding and callback hell. It more like a container, contain the data send from the server. Promise has three status: pending, fulfilled and rejected.

1. **What is Async/Await, how it work?**

Await can only be used inside the async function. Async return a promise object.

During the development of front-end development, sometime we might have serval request, the second request depends on the first request’s data. If use promise, use .then or .catch., which might be a little bit complex. But if use async/await, code is more consice.

1. **What is AJAX**

**A**synchronous **Ja**vaScript and **X**ML, it was used to update website without reloading it. AJAX is one of the lib in jQuery.

--- HTML

1. **What happens when you type a URL in the browser and press enter?**

1. Analysis the URL, check whether the URL is legal URL, or just key word search

2. DNS search, in this step, First check the browser and OS cache to check whether there is IP address relative to this URL. If no, then go to router cache, if still can’t find it, then go to root DNS, until find the final result

3. TCP connection, once we get the IP address, then we can build the connection between server and browser. TCP three hand shake connection.

4. After connected the server successfully, the request will be sent to the server side. Server side handle the request based on the request and return response to the client side

5. Browser receive the response, check the header, cache the resource, handle the data

6. Render website page

1. **What’s different between TCP and UDP**

TCP is more safety, transfer data both sides. TCP need three hand shake connection

UDP: is more faster, transfer data single sides, high likely lost package. Can’t resend it lost package.

1. **What is cookie and session**

Cookie can auto fill the username and password for website. Then user can login in directly. It stored in the client side

Session is stored in server side. It was used to mark and check the user. An example is shop list, remember what user put the product into the shop list.

1. **The step that browser render**
2. After go through HTML and create a dom tree,
3. Build the css ruler tree base on css file
4. Build the render tree
5. Layout
6. Render the website
7. **How to optimize the page**
8. Eliminate the http request
9. Use DNS
10. Delete repeat script
11. Use local cache
12. **HTTP status code**

2xx: handle request successfully

3xx: use on redirection

4xx: request get error response

5xx: something wrong on server side.

1. **Request head and response head**

Request head:

* Accept: text/html
* Accept-encoding
* Accept-language
* Connection
* Cookie

Response:

* Content-type
* Content-encoding
* Date
* Expires
* Last-modified

**Explain what is the scope? What is the difference between var, let, and const in javascript?**

Scope is whether you have the accessability of the variable and function. In the other word, what variables you can access to. Two kind of scope, global scope and local scope.

Global scope: if a variable is declared outside all function or curly braces { }， it is said to be defined in the global scope.

Local scope: two kinds of local scope, one is function scope, and the other one block scope.

* Function scope is declare inside the function
* Block scope is declare inside the curly braces.

Const: const definition, one you define, you can change it. Most used on global scope.

Var and let: let is kind of update of var. let is more used on block scope, like for loop, which is a perfect sample. Var can use on global and function scope.

**What happens when you type a URL in the browser and press enter?**

1. Analysis the URL, check whether the URL is legal URL, or just key word search
2. DNS search, in this step, First check the browser and OS cache to check whether there is IP address relative to this URL. If no, then go to router cache, if still can’t find it, then go to root DNS, until find the final result
3. TCP connection, once we get the IP address, then we can build the connection between server and browser. TCP three hand shake connection.
4. After connected the server successfully, the request will sent to the server side. Server side handle the request based on the request to return response to the client side
5. Browser receive the response, check the header, cache the resource, handle the data
6. Render website page

**Where does javascript execute?**

Javascript is most often run on webpages in side the browser, but it can also run server-side, which is NodeJS

**If Javascript Is Single Threaded, How Is It Asynchronous?**

There is an example to explain this. Image JS is a person, everyday he wake up, he need to wash his face, but he also feel hungry. He open the microwave and push the food inside the microwave. And then he go to wash his face. When he done he go back to kitchen and get and eat the food.

Single Threaded mean that only JS can do all the process. Once JS have async operation, then JS will let other to handle the async operation, JS will handle the result once it return back.

The other example to explain JS must be a single threaded is that if JS is multi-thread, then one thread create DOM and other delete DOM. The browser have no idea which thread he should listen for.

**Hoisting?**

Hoisting mean you can use the function or variable before declare it.

For example, I have a function call addOne, which I didn’t declare it yet. But I can still use this function before I declare. That is because function declaration will move forward and save in cache, and then call the function.

Num = 6

Console.log(num)

Var num;

**JavaScript — is it Compiled or Interpreted?**

JavaScript is an interpreted language, not a compiled language. A program such as C++ or Java needs to be compiled before it is run.

JavaScript don’t have compilation step. Instead, an interpreter in the browser reads over the JavaScript code, interprets each line, and runs it.

If use ts, we need compile. It will translate the TS code to Nodejs code.

**Closures?**

Closures mean the variable inside the function can be read by its child function. Each function is a close package. You can’t read the variable inside the function directly.

Why we use closures?

1. If you want to read the internal function variable, then you can return child function, which contain the function variable.
2. If you want some function variable don’t destroy, after finish the function