

JS Questions LIJIA XU (41)

45. test string as a literal and as an object?

```
function isString(str){
  return typeof(str) === 'string' || str
```

instance of String; 46 typical use of the anonymous function in JS?

1. No name is needed if function only used in one place

```
setTimeout(function(){
  alert("Hi");
}, 1000);
```

2. use the event handler, inline functions have advantage in case can access vars in parent scopes

3. passing anonymous function as a parameter to calling function

```
function process(callback){
  if(typeof callback === 'function'){
    callback();
  }
}
```

4. process(callback, function){}

5. advantages: 1. reduce code

2. avoid needless global namespace populations

47. How to set default parameter value?

```
function sendEmail(config, provider){
  provider = typeof provider !== 'undefined' ?
    provider : 'Gmail';
}
```

```
function sendEmail(config, provider){
  provider = provider || 'Gmail';
}
```

48. merge two JavaScript Object dynamically

1. const merge = (toObj, fromObj) => Object.assign(toObj, fromObj);

2. without using build-in function

```
function merge(toObj, fromObj){
  if(typeof toObj !== 'object' && typeof fromObj !== 'object'){
    for (var prop in fromObj){
      if (fromObj.hasOwnProperty(prop)){
        toObj[prop] = fromObj[prop];
      }
    }
  }
}
```

3. if (typeof toObj !== 'object' && typeof fromObj !== 'object')

4. for (var prop in fromObj){

5. if (fromObj.hasOwnProperty(prop)){

6. toObj[prop] = fromObj[prop];

7. } // throw "only on Objects";

49 non-enumerable property

```
var person = {name: 'Bob'}
```

```
Object.defineProperty(person, 'phoneNo', {
  value: '88888',
  enumerable: false});
```

changing non-enumerable property value will not success

50. function binding

```
var clickHandler = {
  message: 'click event handler',
  handleClick: function(event){
    console.log(this.message);
  }
}
```

3; var btn = document.getElementById(btn);

```
btn.addEventListener('click',
  clickHandler.handleClick.bind(clickHandler));
```

4. coding Questions

1. Objects, including arrays are passed by reference

2. strings, booleans and numbers are passed by value

```
1. var strA = "A";
   var strB = strA;
   strB = "C";
   console.log(strA) // "A"

2. var objA = {prop1: 2};
   var objB = objA;
   objB.prop1 = 90;
   console.log(objA) // {prop1: 90}
```

3. // reassign

```
var objA = {prop1: 2};
var objB = objA;
objB = {3};
console.log(objA) // {prop1: 2}
```

5. reference different arrays

```
var arrA = [0, 1, 2];
var arrB = arrA.slice();
arrB[0] = 42;
console.log(arrA) // [0, 1, 2]
```

by value

```
var arrA = [{prop1: "value of A"},
  {anotherProp: "also value of A"}];
```

```
var arrB = arrA.slice();
```

```
arrB[0].prop1 = 42 // pull request to change the code
arrB[3] = 22
console.log(arrA)
```

```
// [{prop1: 42}, {anotherProp: "also value of A"},
  3, 4, 5]
```

Hoisting

1. console.log(employeeId)

// Reference Error: employeeId is not defined

2. console.log(employeeId)

var employeeId = '19000';

// undefined

3. var employeeId = '123456'

```
(function(){
  console.log(employeeId);
  var employeeId = '123456';
})();
```

// undefined

Questions from other reference

1. var y = 1;

```
if (function fun() {}){
  y += typeof fun;
}
```

3 console.log(y) // undefined

2. single line statement to remove all duplicate elements

```
var array = [1, 3, 3, 5, 7];
array = [...new Set(array)];
```

// spread operator

3. function add(num1) {

return function(num2) {

return function(num3) {

return num1 + num2 + num3;

} console.log(add(1)(2)(3)) // "6"

3. do shallow copy, objects in array will be passed by reference

4. number, string or Boolean will be passed by value