JS Questions LIJIA XU (61)

1. null is also considered as object

```
console.log((bar !== null) && (typeof bar === "object"));
```

2.
```
(function(){
  var a=b=3
})();
console.log("a defined?"+(typeof a !== 'undefined'));
console.log("b defined?"+(typeof b !== 'undefined'));
```

var a=b=3 ⇒ equal
b=3 // is not using 'use strict'
b will be global
var a=b
so ⇒ a defined? false
b defined? true

* use strict is important

3.
```
var myObject={
  foo:"bar",
  func:function(){
    var self=this;
    console.log("outer func: this.foo="+this.foo);
    console.log("outer func: self.foo="+self.foo);
    (function(){
      console.log("inner func: this.foo="+this.foo);
      console.log("outer func: self.foo="+self.foo);
    }());// inner func
  }// outter func
};// object
myObject.func()
```

outputs:
bar
bar
undefined
bar

4. wrapping the entire content of a JS source file in a function Block?

name spacing
easier to reference

5. why 'use strict' important?

① show errors failed silently

② prevents accidental globals (normal mode)
assign a value to an undeclared variable auto create a global variable

③ eliminates this coercion

④ disallows duplicate parameter values

⑤ make eval() safer

⑥ throws error on invalid usage of delete

6. will they both return the same thing?

```
function foo1()
{
  return{
    bar:"hello"
  };
}
function foo2()
{
  return
  {
    bar:"hello"
  };
}
```

foo1 ⇒ object {bar:"hello"}

due to semicolon is auto inserted after return
// foo2 will not throw errors

foo2 ⇒ undefined

7. what is NaN?

NaN ⇒ typeof NaN ⇒ "Number"

NaN compared to anything even itself is false

:isNaN() ⇒ not perfect

value !== value // only produce

8. what will be the output?

```
console.log(0.1+0.2); //0.300...0004
console.log(0.1+0.2==0.3); //false
function checkEqual(num1,num2){
  return Math.abs(num1-num2)<
    Number.EPSILON;
}
```

9. check is Int

```
function isInteger(x){return(x^0)===x;}
function isInteger(x){return Math.round(x)===x;}
```
↓ also works
```
function isInteger(x){
  return Math.round(x)===x;}
```

10. what order will the number 1-4 be logged to the console

```
(function(){
  console.log(1);
  setTimeout(function(){console.log(2)},1000);
  setTimeout(function(){console.log(3)},0);
  consde.log(4);
})(); // will prints
```

1
4
3
2

11. if a string is a palindrome

```
function isPalindrome(str){
  str=str.replace(/\W/g,"").toLowerCase();
  return (str===str.split('').reverse().join(''));
}// review regular expression
```

12. make a function:

```
console.log(sum(2,3)) //outputs 5
console.log(sum(2)(3)) //outputs 5
```

Method 1:

```
function sum(x){
  if(arguments.length==2){
    return arguments[0]+arguments[1]
  }else{
    return function(y){return x+y;}
  }
}
```

method 2
```
function(x,y){
  if(y!==undefined){
    return x+y
  }else{return function(y){return x+y}}
}
```