

前端试卷3

填空（2分/空*10空=20分）

简答（10分/题*2题=20分）

编程（20分/题*2题=40分）

论述（20分/题*2题=40分）

一、选择

1、快排的平均时间复杂度 A

A、 $n \log n$ B、 n^2 C、 n^3 D、 $\log n$

2、

```
var name = "window";
var obj = {
  name: "obj",
  func1: function() {
    console.log(this.name)
  },
  func2: function() {
    setTimeout(function() {
      console.log(this.name)
    })
  }
}
```

```
obj.func1()
obj.func2()
var f = obj.func1
f()
```

上述代码输出 B

A、obj window obj B、obj window window C、obj obj obj D、obj obj window

3、http 状态码中表示权限受限制的是 C

A、401 B、302 C、403 D、304

4、下列 url 与 <http://www.qiniu.com> 同源的是 C

A、<https://www.qiniu.com> B、<http://support.qiniu.com> C、<http://www.qiniu.com/products/kodo> D、<http://www.qiniu.com:9000/kodo>

5、下面属于网络层协议的是 B

A、TCP 协议 B、IP 协议 C、HTTP 协议 D、DNS 协议

6、对名为fido的文件用chmod 551 fido 进行了修改，则它的许可权是D。

A -rwxr-xr-x
B -rwxr - r -
C -r - r - r -
D -r-xr-x-x

7、比较下列 css 选择器优先级正确的是 D

A、div#test1 > div.test .link > div.test a > div.test [val="qiniu"]
B、div.test .link > div#test1 > div.test a > div.test[val="qiniu"]
C、div.test .link > div#test1 > div.test[val="qiniu"] > div.test a
D、div#test1 > div.test .link > div.test[val="qiniu"] > div.test a

8、'13100000000'.replace(/(\w{3})\w{4}(\w{4})/, '\$1***\$2') 的结果是 A

A、131***0000 B、131\$1***\$20000 C、\$1***\$2 D、131*0000

9、

```
function fn(a) { console.log(a);
  var a = 2; function a() {} console.log(a);
}

fn(1);
```

上述代码输出:

A、function a() {}, 2

B、2, 2

C、function a() {}, function a() {}

D、undefined, function a() {}

10、关于 http 说法错误的是 A

A、http 是有状态协议

B、http 是基于 TCP/IP

C、http 建立连接需要经过 3 次握手

D、http 可以通过设置 keep-alive 设置长连接

一、填空

1、列举5个行内元素: _____

答案: a b span img input select strong

2、盒模型包含内容: _____

答案: 内容(content)、填充(padding)、边界(margin)、边框(border)

3、说出5个 css 选择器: _____

答案: 1.id选择器 (# myid) 2.类选择器 (.myclassname) 3.标签选择器 (div, h1, p) 4.相邻选择器 (h1 + p) 5.子选择器 (ul > li) 6.后代选择器 (li a) 7.通配符选择器 (*) 8.属性选择器 (a[rel = "external"]) 9.伪类选择器 (a:hover, li:nth-child)

4、[1, 2, 3, 4, 5].map(parseInt) 的输出是什么: _____

答案: [1, NaN, NaN, NaN, NaN]

5、var re = /(\w+)\s(\w+)/;

```
var str = 'John Smith';  
console.log(str.replace(re, '$2, $1'));
```

上述代码的输出:

答案: Smith, John

6、判断 true or false

null == undefined:_____ (答案: true)

null === undefined:_____ (答案: false)

7、http 状态码中, 304 表示_____, 401 表示_____

答案: 304 Not Modified 401 Unauthorized

8、console.log(a);

```
let a = 2;
```

上述代码的执行结果是什么: _____

答案: 抛出异常, a 未定义

二、简单

1、请描述一下 cookies, sessionStorage 和 localStorage 的区别?

答案:

```

cookieClient Side
  cookiehttp
  sessionStoragelocalStorage

  cookie4k
  sessionStoragelocalStorage cookie5M

  localStorage
  sessionStorage
  cookie      cookie

```

2、什么是闭包 (closure) ?

答案:

,

- 1.
- 2.
- 3.

三、编程题

1、如何实现数组的随机排序?

答案:

```

var arr = [1,2,3,4,5,6,7,8,9,10];
function randSort1(arr){
  for(var i = 0,len = arr.length;i < len; i++){
    var rand = parseInt(Math.random()*len);
    var temp = arr[rand];
    arr[rand] = arr[i];
    arr[i] = temp;
  }
  return arr;
}
console.log(randSort1(arr));

var arr = [1,2,3,4,5,6,7,8,9,10];
function randSort2(arr){
  var mixedArray = [];
  while(arr.length > 0){
    var randomIndex = parseInt(Math.random()*arr.length);
    mixedArray.push(arr[randomIndex]);
    arr.splice(randomIndex, 1);
  }
  return mixedArray;
}
console.log(randSort2(arr));

var arr = [1,2,3,4,5,6,7,8,9,10];
arr.sort(function(){
  return Math.random() - 0.5;
})
console.log(arr);

```

2

```

/**
 * 延迟执行, 在 t 毫秒后执行 fn
 * @param fn function
 * @param t 延迟时间, 单位 ms
 */

```

```
function debounce (fn, t) {  
}  
  
function A() {  
  console.log('A')  
}  
  
var tA = debounce(A, 1000)  
tA() // 1s 后输出 'A'
```

答案:

// 更完整的答案请参考 <https://github.com/lodash/lodash/blob/4.17.4/lodash.js#L10321>

// 只需要能通过闭包写出基本的就可以, 如下

```
function debounce (fn, t) {  
  var interval = null;  
  return function () {  
    var args = arguments  
    var context = this  
    if (interval) {  
      clearTimeout(interval)  
    }  
    var later = function () {  
      interval = null  
      fn.apply(context, args)  
    }  
    setTimeout(fn, later)  
  }  
}
```

四、论述

1、说说 es6 增加的新特性和对这些的理解 (大致说出 3 点)

答案: 参考<http://es6.ruanyifeng.com/#docs/destructuring>

- 1、let const 块级作用域
- 2、变量的解构赋值
- 3、promise
- 4、for ... of

2、谈谈关于前端的发展前景