

前端试卷1

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）

选择题

1. 以下哪个元素已被 HTML5 标准废弃？

A. base B. center C. legend D. time

答案：B 原因：html5中，对纯表现元素的标签进行了移除，使用css去控制。

2. 以下哪一个对象可以触发 HTML5 的应用程序缓存（Cache Manifest）事件？

A. window.localStorage B. window.appCache C. window.localStorage D. window.applicationCache 答案：D

3. 以下哪一种 CSS 代码可以使以 “https://qiniu.” 开头的超链接显示为蓝色？

A. a[href^='https://qiniu.']* { color: blue; } B. a { prepend: href='https://qiniu.'; color: blue; } C. a:before { content: attire(href) "https://qiniu."; color: blue; } D. CSS 无法做到 答案：A

4. 下面的 JavaScript 代码输出结果是？

```
var a = 10;
a.pro = 10;
console.log(a.pro + a);
```

```
var s = 'hello';
s.pro = 'world';
console.log(s.pro + s);
```

A. 10 'hello world'
B. 20 'helloworld'
C. NaN 'helloworld'
D. NaN 'undefinedhello'

答案：D

5. 使用 git 做版本管理，一下那些操作可以将远程分支（origin/develop）的代码更新到本地代码中？

A. git fetch origin develop B. git merge origin/develop C. git pull origin develop D. git rebase origin/develop 答案：C

6. 假设你使用 XMLHttpRequest (xhr) 向 “/resource” 发起一个请求，服务端返回了 301 或者 302 状态码，接下去的预期是什么？

A. readystate 返回了 3，你必须检查 301 或者 302 的 response header 并针对 301 或者 302 调用 xhr.continue() 处理。 B. XHR 对象将自动跳转至新的资源目标地址，除非目标在一个不同的域。 C. readystate 返回了 4，但是 responseText 和 responseXML 为 null。你必须重新发送一次请求至 xhr.getResponseHeader(“Location”) D. 视 XHR 对象是如何被设置的。如果 xhr.refirects=false，则你必须重新手动发送一次新请求，反之则自动跳转，除非目标在一个不同的域。 答案：B

7. 下面哪些不会生成BFC

A. 根元素 B. float属性不为none C. position为 absolute 或 fixed D. display为block 答案 D

8. 在cmd规范中，require(‘a package’) 表示：

A. 定义一个包 B. 同步引入一个包 C. 定义一个模块 D. 异步引入一个包 答案 B

9. 怎样查看当前文件夹所在磁盘还有多少空间？

A. df / B. du / C. df . D. du . 答案 C

10. 下面有关csrf的描述，说法错误的是？

A. CSRF则通过伪装来自受信任用户的请求来利用受信任的网站 B. xss是实现csrf的诸多途径中的一条 C. 在客户端页面增加伪随机数可以阻挡csrf D. 过滤用户输入的内容也可以阻挡csrf 答案 B (xss 是注入的一种)

填空题

1. 请写出一个能符合以下规则的正则表达式_____。

- 第一个字符必为除数字及字母外的字符
- 字符串中必须包含至少一个数字
- 字符串中必须包含至少一个字母
- 总字符数大于10位 参考答案: `^[^0-9A-Za-z]+(?=[A-Z]+)(?=[a-z]+)(?=.*\d+)\{9,$\}`, 匹配 %aaaaaaaaAaa9, 表达式意思为, 第一个非数字字母字符至少一个, 右侧字符必须含有A-Z、a-z、数字各至少一个, 长度至少为(1+9)个。此题考察正则表达式的应用能力(字符校验、长度校验、正/反向预搜索, 正/反向不匹配), 这种情况在前端密码校验中非常常见, 网上抄可以, 但是至少要知道抄过来的对不对, 至少我知道网上很多是有错的。

2. CSS 中_____方法可以让父元素与其第一个子元素不产生 margin 折叠 (margin collapse)。

答案: `overflow:hidden/scroll` (不包含visible即可)、`float:left/right`、`position:absolute`;
`display:inline-block;padding-top/border`不为0等, 答对其一即可。原理: 如上属性会导致元素margin不邻接, 所以不会合并。

3. 用 left 和 transform: translate 实现从右侧划入的 css 动画, 区别是_____

答案: left 会触发 layout, transform 不会, 性能更好, 动画更加平滑 (意思对即可)

4. 请写出隐藏电话号码中间4位的正则表达式方法 _____

'13100000000' -> '131****0000'

答案: `'13100000000'.replace(/(\w{3})\w{4}(\w{4})/, '$1...$2')`

5. 堆排序的时间复杂度是_____

答案: $O(n \log(n))$

6. CSS 中_____属性可以让父元素与其第一个子元素不产生margin折叠 (margin collapse)。

答案: `overflow:hidden/scroll` (不包含visible即可)、`float:left/right`、`position:absolute`;
`display:inline-block;padding-top/border`不为0等, 答对其一即可。原理: 如上属性会导致元素margin不邻接, 所以不会合并。

7. transform: translateY(-50%) 的意思是_____。

参考答案: 沿Y轴向上移动自身高度的50%。

8. 写一段 CSS 代码可以使以 “https://qiniu.” 开头的超链接显示为蓝色 _____。

参考答案: `a[href^='https://qiniu.'] { color: blue; }`

9. 使用 git 做版本管理, 那些操作可以将远程分支 (origin/develop) 的代码更新到本地代码中 _____。

参考答案: `git pull origin develop`

10. 七牛官网是_____。

答案: <https://www.qiniu.com>

简答题

1. 请描述你所知的在 CSS 中描述长度的单位, 并解释它们的区别?

参考答案: in, cm, mm, pt, pc, em, ex, px, 知道 px em pt 为最基本, 其他的算是加分项

区别可参考: <http://stackoverflow.com/questions/2385829/what-is-the-difference-between-px-em-and-ex>

2. 问题: 简述 JavaScript

的同源策略定义(什么情况属于同源, 什么情况不是)、同源策略的作用、需要跨域访问的常见情况, 以及对应的解决方案。

参考答案: 定义: 见https://developer.mozilla.org/zh-CN/docs/Web/Security/Same-origin_policy

作用: 限制不安全的非主动的跨站读取, 比如网页被注入, 此时同源策略可以限制数据被非法传递到注入者手中。 常见情况:

ajax与不同域server之间互访。解决方案: access control、jsonp

不同window之间互访(窗口与窗口、iframe与父窗口、iframe与iframe) 解决方案: document.name、Cross Frame、post message

父域名与子域名之间互访。解决方案: document.domain

编程题

1. JS实现两个值的比较(包括object(array, {})的深度比较)

参考答案:

```
/*
 *
 *number
 */
Array.prototype.unique = function () {
    this.sort ();
    var n = [this[0]];
    for (var i = 1, length = this.length; i < length; i++) {
        if (this[i] !== n[n.length - 1]) {
            n.push (this[i]);
        }
    }
    return n;
};
/*
 *
 *
 */
function SameValue (o1, o2) {
    //- false
    if (typeof o1 !== typeof o2) {
        return false;
    }
    /*
    *:undefined,object,number,boolean,string,function
    * Object String()
    */
    if (typeof o1 !== "object") {
        return String (o1) === String (o2);
    }

    /*
    * typeof  object
    *object  null,array,{}
    *
    */
}
```

```

if (o1 === null || o2 === null) {
    if (o1 === o2) {
        return true;
    }else {
        return false;
    }
}

if (Array.isArray (o1) || Array.isArray (o2)) {
    if (Array.isArray (o1) && Array.isArray (o2)) {
        /*
        *[]
        */
        if (o1.length === o2.length) {
            //-
            for (var i =0, length = o1.length; i < length; i++) {
                if (!SameValue (o1, o2)) {
                    //- false
                    return false;
                }
            }
            //- true
            return true;
        }else {
            //- false
            return false;
        }
    }else {
        return false;
    }
}

/*
* {}
*/
var bool = null,
    keyArr1 = [],
    keyArr2 = [],
    keyArr = [];
// key
for (var i in o1) {
    keyArr1.push (i);
}
for (var j in o2) {
    keyArr2.push (j);
}
// key1 key2
keyArr = keyArr1.concat(keyArr2);
//
keyArr = keyArr.unique();

// key value
for (var k = 0, klength = keyArr.length; k < klength; k++) {
    if (o1[k] && o2[k]) {
        bool = SameValue (o1, o2);
    }else {
        return false;
    }
}

```

```
    return bool;
}
```

2. 实现如下语法的功能: `var a = add(2)(3)(4);`

参考答案:

```
function add(a) {
  var temp = function(b) {
    return add(a + b);
  }
  temp.valueOf = temp.toString = function() {
    return a;
  };
  return temp;
}
var ans = add(2)(3)(4);
console.log(ans); // 9

(ES6)
function add(num){
  num += ~~add;
  add.num = num;
  return add;
}
add.valueOf = add.toString = function(){return add.num};
var ans = add(3)(4)(5)(6); // 18
alert(ans);

let add = (...a) => {
  let helper = (...b) => {
    return add(...[...a, ...b]);
  };

  helper.toString = helper.valueOf = () => {
    let sum = 0;
    for (let item of a)
      sum += item;
    return sum;
  };

  return helper;
};

alert(add(1)(2)) // 3
alert(add(1, 2, 3)(10)) // 16
alert(add(1)(2)(3)(4)(5)) // 15
alert(add(1)(2, 3)(4, 5)) // 15
```

论述题

1. 从打开浏览器输入网址, 到浏览器展示出网页内容, 这一过程中发生了哪些事情?
2. 你是如何看待目前的人工智能行业的发展?