

前端试卷2

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：

- JavaScript定义var a="40",var b=7,则执行a%b会得到(b)
a. "5" b. 5 c. undefined d. null
- +new Array(017) 输出 (a)
a. NaN b. 17 c. 15 d. error
- display:none 和 visibility:hidden的说法, 正确的是? (a,b)
a. display:none与visibility:hidden都可以用来隐藏某个元素
b. display:none在隐藏元素的时候, 将其占位空间也去掉; 而visibility:hidden只是隐藏了内容而已, 其占位空间仍然保留。
c. display:none是不符合html5标准的
d. visibility:hidden是不符合html5标准的
- 元素的alt和title有什么异同, 选出正确的说法? (b)
a. 不同的浏览器, 表现一样
b. alt和title同时设置的时候, alt作为图片的替代文字出现, title是图片的解释文字
c. alt和title同时设置的时候, title作为图片的替代文字出现, alt是图片的解释文字
d. 以上说法都不正确
- 以下代码符合html5规范的是(b)
a. <div></div>
b. <input type=" text" disabled>
c. <label id=" mylabel" ></label>
d. <div myname=" javk" ></div>
- 具有12个关键字的有序表, 折半查找的平均查找长度(a)
a. 3.1
b. 4
c. 2.5
d. 5
- 序列[2, 1, 4, 9, 8, 10, 6, 20] 是某排序算法第二轮排序的结果, 则该算法只能是(a)
a.
b. 选择排序
c.
d. 插入排序
- 超文本的含义是 (c)
a.
b.
c.
d.
- 下面哪个文件定义了网络服务的端口? (b)
a. /etc/netport
b. /etc/services
c. /etc/server
d. /etc/netconf
- 在掩码为255.255.224.0条件下, 下面哪些ip地址属于同一网段 (a c)
a. 192.168.235.25
b. 192.168.188.99
c. 192.168.246.187
d. 192.168.67.28

填空：

- HTTP协议是建立在 _____ 协议之上的一种应用
答案： TCP
- HTTP状态码中, 表示永久性重定向的是_____表示临时重定向的是_____
答案： 301 302
- Restful API 常用的HTTP动词_____两个不常用的HTTP动词_____

答案: get post put patch delete , head options

4. 浏览器在发送请求时, _____ 头信息代表客户端系统、浏览器类型等信息; _____ 头信息表示提交的数据类型
答案: User-Agent, Content-Type
5. Web存储常见的有哪些_____
答案: cookie, sessionStorage, localStorage
6. CSS 的 position 属性值有_____
答案: inherit, static, relative, absolute, fixed
7. CSS 盒模型包含的属性有哪些? _____
答案: box-sizing, height, max-height, min-height, width, max-width, min-width, padding, border, margin
8. JavaScript 的基本数据类型 _____
答案: string number null undefined boolean
9. JavaScript Ajax 是使用哪个对象来向服务端发异步请求? _____
答案: XMLHttpRequest
10. Nginx 可以搭建若干个 server, 每个 server 是通过 _____ 和 _____ 属性来区分
listen 和 server_name

简答

1. css定位属性absolute, fixed相同点与区别
参考答案:
absolute和fixed的相同点, 都会脱离文档流
不同点: absolute相对是第一个有定位的祖元素进行定位, fixed相对于浏览器进行定位。
在有滚动条的页面中, absolute会跟着祖元素进行移动, fixed固定在页面的具体位置
2. js延迟加载的方式有哪些?
参考答案:
1. defer和async
2. 动态创建DOM方式 (创建script, 插入到DOM中, 加载完毕后callBack)
3. 按需异步载入js

编程

1. 数组去重, 例如输入['1', 1, 2, 2, NaN, NaN, null, undefined], 输出['1', 1, 2, NaN, null, undefined]
参考答案 主要考察NaN!=NaN

```
function unique (arr){
  var r = [],
  hash = {},
  flag = false;

  for (var i = 0, length = arr.length; i < length; i++) {
    if (!hash[arr[i]]) {
      //1'1'
      if (typeof arr[i] == "string") {
        hash[arr[i]+"string"] = true;
      }else {
        hash[arr[i]] = true;
      }
      //NaN
      if (arr[i].toString() === "NaN") {
        flag = true;
      }else {
        r.push(arr[i]);
      }
    }
  }

  //NaN
  if (flag) {
    r.push(NaN);
    return r;
  }else {
    return r;
  }
}
```

2. JS""

参考答案

```
// stack
function Stack () {
  this._size = 0;
  this._storage = {};
}
Stack.prototype.Push = function (data) {
  this._storage[++this._size] = data;
}
Stack.prototype.Pop = function () {
  if (this._size) {
    var deleteData = this._storage[this._size];
    delete this._storage[this._size--];
    return deleteData;
  }
  return false;
}
Stack.prototype.IsEmpty = function () {
  if (this._size) {
    return false;
  } else {
    return true;
  }
}
Stack.prototype.Size = function () {
  return this._size;
}
// Stack -> realize -> Queue
function Queue () {
  this._stack1 = new Stack();
  this._stack2 = new Stack();
  this._size = 0;
  this._storage = {};
}
Queue.prototype.Enqueue = function (data) {
  this._stack1.Push(data);
  this._storage = this._stack1;
}
Queue.prototype.Dequeue = function () {
  if (this._stack1.IsEmpty() && this._stack2.IsEmpty()) {
    throw new Error("空队列");
  }
  if (this._stack2.IsEmpty() && !this._stack1.IsEmpty()) {
    while (!this._stack1.IsEmpty()) {
      this._stack2.Push(this._stack1.Pop());
    }
    var deleteData = this._stack2.Pop();
    this._storage = this._stack2;
    return deleteData;
  }
}
Queue.prototype.IsEmpty = function () {
  if (this._stack1.IsEmpty() && this._stack2.IsEmpty()) {
    return true;
  } else {
    return false;
  }
}
Queue.prototype.Size = function () {
  return this._storage.Size();
}

var q = new Queue();
q.Enqueue("121");
q.Enqueue("321");q.Size()
```

```

function Stack () {
  this._size = 0;
  this._storage = {};
}
Stack.prototype.Push = function (data) {
  this._storage[++this._size] = data;
}
Stack.prototype.Pop = function () {
  if (this._size) {
    var deleteData = this._storage[this._size];
    delete this._storage[this._size--];
    return deleteData;
  }
  return false;
}
Stack.prototype.IsEmpty = function () {
  if (this._size) {
    return false;
  }else{
    return true;
  }
}
Stack.prototype.Size = function () {
  return this._size;
}
// Stack -> Realize -> Queue
function Queue () {
  this._stack1 = new Stack();
  this._stack2 = new Stack();
  this._size = 0;
  this._storage = {};
}
Queue.prototype.Enqueue = function (data) {
  this._stack1.Push(data);
  this._storage = this._stack1;
}
Queue.prototype.Dequeue = function () {
  if (this._stack1.IsEmpty() && this._stack2.IsEmpty()) {
    throw new Error("空队列");
  }
  if (this._stack2.IsEmpty() && !this._stack1.IsEmpty()) {
    while (!this._stack1.IsEmpty()) {
      this._stack2.Push(this._stack1.Pop());
    }
    var deleteData = this._stack2.Pop();
    this._storage = this._stack2;
    return deleteData;
  }
}
Queue.prototype.IsEmpty = function () {
  if (this._stack1.IsEmpty() && this._stack2.IsEmpty()) {
    return true;
  }else{
    return false;
  }
}
Queue.prototype.Size = function () {
  return this._storage.Size();
};var q = new Queue();q.Enqueue("121");q.Enqueue("321");q.Size()
< 2

```

论述：

1. Cookie, sessionStorage, localStorage区别

参考答案：sessionStorage, localStorage统称webStorage；

Cookie的作用是与服务器进行交互, 作为HTTP规范的一部分而存在, 而Web Storage仅仅是为了在本地“存储”数据而生

(1) 存储大小：

cookie, 4k; sessionStorage, localStorage, 5M或更大；

(2) 过期时间：

cookie, 过期时间前一直有效，默认cookie（没有设置expires的cookie）失效时间直到关闭浏览器，设置方式

保留COOKIES一个小时-----Response.Cookies("MyCookie").Expires = DateAdd("h", 1, Now());

sessionStorage关闭即删除，只要同源的同窗口（或tab）中，刷新页面或进入同源的不同页面，数据始终存在

localStorage永不丢失

(3) 服务器交互带宽使用情况：

cookie放在请求头发送给服务端，浪费带宽

sessionStorage, localStorage无于服务端交互

(4) 存储位置：

cookie以键值对形式存储在http请求的header中发送给服务端，同时从服务端获得的cookie是 键=值分号隔开的形式存储

sessionStorage, localStorage存储在本地

(5) 共享：

cookie受同源策略限制

sessionStorage同一窗口共享

localStorage受同源限制

(6) 用法：

cookie需要自己封装方法

sessionStorage, localStorage有接口: setItem, getItem, removeItem, clear

2. Get, post区别

参考答案:

GET - 从指定的资源请求数据

POST - 向指定的资源提交要被处理的数据

GET方式需要使用 Request.QueryString 来取得变量的值

POST方式通过 Request.Form 来获取变量的值

也就是说 Get 是通过地址栏来传值, 而 Post 是通过提交表单来传值。

大小限制

get 1024字节

post 无限制, 但可以自行设定

安全性

post较get安全

数据存放

get传输的参数附在URL后

post传输的数据放在HTTP请求体中

幂等

get是幂等的, post不是

缓存, 收藏, 历史

get都可以, post没有。