

面试题

- JS中使用 `typeof` 的类型
 - 考点: `js基本类型`

```
`number`  
`string`  
`object`  
`boolean`  
`undefined`  
`function`
```

- `===` 和 `==` 何时使用
 - 考点: `强制类型转换`

```
if (obj.a == null) {  
  // 相当于obj.a=== null||obj.a=== undefined, 简写形式  
  // 这是jquery源码中推荐的写法  
  // 其他情况全部使用 ===  
}
```

- `window.onload` 和 `DOMContentLoaded` 的区别
 - `浏览器渲染过程`
- 用js创建10个 `<a>` 标签, 点击时弹出对应的序号
 - 考点: `作用域`

```
// 全局作用域  
var i;  
for (i = 0; i < 10; i++) {  
  (function(i) {  
    // 函数作用域  
    var a = document.createElement('a');  
    a.innerHTML = i + '<br>';  
    a.addEventListener('click', function(e) {  
      e.preventDefault();  
      alert(i); // i 是自由变量, 需要去父级作用域寻找, 定义一个自执行函数, 形成一个函数作用域  
    }, false)  
    document.body.appendChild(a);  
  })(i)  
}
```

- 简述如何实现一个 `模块加载器`, 实现类似 `require.js` 的基本功能
 - 考点: `JS模块化`
- 实现数组的 `随机排序`
 - 考点: `JS基础算法`
- 对 `变量提升` 的理解
 - 变量定义提升
 - 函数声明提升, 执行代码前先读取函数声明

- 说明 `this` 几种不同的使用场景

1. 作为构造函数执行
2. 作为对象执行
3. 作为普通函数执行
4. `call, apply, bind`

- 如何理解 `作用域`

- `作用域` 即变量与函数的可访问范围，即作用域控制着变量与函数的可见性和生命周期。在JavaScript中，变量的作用域有全局作用域和局部作用域两种，局部作用域又称为函数作用域。

1. 自由变量
2. 作用域链，即自由变量的查找
3. 闭包

- 实际开发中 `闭包` 的应用

```
function isFirstLoad() {  
  var _list = [];  
  return function(id) {  
    var index = _list.indexOf(id);  
    if (index > -0) {  
      return false;  
    } else {  
      _list.push(id);  
      return true;  
    }  
  }  
}  
  
var firstLoad = isFirstLoad();  
firstLoad(1); // true  
firstLoad(1); // false  
firstLoad(2); // true  
firstLoad(2); // false  
//在 isFirstLoad函数外面，不可能修改 _list 的值
```

- 同步和异步的区别是什么?分别举一个同步和异步的例子

- 同步会阻塞代码的运行，异步不会阻塞

```
// 异步  
console.log(100);  
setTimeout(function(){  
  console.log(200);  
},1000)  
console.log(300);  
// 执行顺序 分别打印 100, 300, 200
```

```
// 同步  
console.log(100);  
alert(200); // 会等待，点击之后再打印 300  
console.log(300);
```

- 一个关于 `setTimeout` 的笔试题

```
// 异步
console.log(1);
setTimeout(function(){
  console.log(2);
},0)
console.log(3);
setTimeout(function(){
  console.log(4);
},1000)
console.log(5);
// 执行顺序 分别打印 1,3,5,2,4
```

- 前端使用异步的场景有哪些
 - 定时任务: `setTimeout` , `setInterval`
 - 网络请求: `ajax` 请求, 动态 `` 加载
 - 事件绑定

- 获取随机数, 要求是长度一致的字符串

```
var random = Math.random();
random = random + '0000000000';
random = random.slice(0,10);
```

- 写一个能遍历对象和数组的通用 `forEach` 函数

```
function(obj, fn) {
  var key;
  if (obj instanceof Array) {
    obj.forEach(function(item, index) {
      fn(index, item)
    })
  } else {
    for (key in obj) {
      fn(key, obj[key])
    }
  }
}
```