

ecmascript

变量类型

原型和原型链

执行上下文

this

作用域

作用域链

```
function F1() {  
  var a = 100;  
  return function () {  
    console.log(a) // 自由变量，与作用域有关  
  }  
}  
var f1 = F1()  
function F2(f1) {  
  var a = 200;  
  f1()  
}  
F2(f1)
```

函数作为返回值

```
function F1() {  
  var a = 100;  
  return function () {  
    console.log(a) // 自由变量，与作用域有关  
  }  
}  
var f1 = F1()  
function F2(f1) {  
  var a = 200;  
  f1()  
}  
F2(f1)
```

函数作为参数传递

变量定义

函数声明（注意和函数表达式的区别）

说一下对变量提升的理解

说明this几种不同的使用场景

```
1 // 写一个错误的函数！  
2 var i, a  
3 for (i = 0; i < 10; i++) {  
4   a = document.createElement('a')  
5   a.innerHTML = i + '<br>  
6   a.addEventListener('click', function (e) {  
7     e.preventDefault()  
8     alert(i)  
9   })  
10  document.body.appendChild(a)  
11 }
```

```
1 // 修正函数写法！  
2 var i  
3 for (i = 0; i < 10; i++) {  
4   (function () {  
5     var a = document.createElement('a')  
6     a.innerHTML = i + '<br>  
7     a.addEventListener('click', function (e) {  
8       e.preventDefault()  
9       alert(i)  
10    })  
11    document.body.appendChild(a)  
12  })()  
13 }
```

创建10个<a>标签，点击的时候弹出来对应的序号

思考题

自由变量

如何理解作用域

作用域链，即自由变量的查找

闭包的两个场景

```
1 // 模拟实际应用中主调用子封装变量，收敛权限  
2 function isFirstLoad() {  
3   var _list = []  
4   return function (id) {  
5     if (_list.indexOf(id) == 0) {  
6       return false  
7     } else {  
8       _list.push(id)  
9       return true  
10    }  
11  }  
12 }  
13  
14 // 调用  
15 var firstLoad = isFirstLoad()  
16 firstLoad(10) // true  
17 firstLoad(10) // false  
18 firstLoad(20) // true
```

实际开发中闭包的应用

异步和单线程