

第二阶段教学计划-2

- 执行时间：2022年8月11日
 - 执行班级：H2204
 - 授课老师：杨标
 - 授课内容：面向对象程序设计，执行上下文，常用内置对象，函数高级部分，闭包
-

面向对象【一】

1. 对象的概念
2. 对象的创建
3. 使用Object来创建对象
4. 对象属性的调用
 - 通过 `.` 操作符调用
 - 通过 `[]` 操作符调用
5. 使用构造函数创建对象
6. 使用工厂模式创建对象
7. 构造函数与普通函数的区别
8. 工厂模式与构造函数模式区别
9. 基础篇总结
 - 理解面向对象以及对象的概念
 - 对象的创建的几种方式
 - 构造函数以及构造函数的特点
 - 对象属性的调用

面向对象【二】

1. 特殊属性对象的定义
2. `Object.defineProperty()` 定义对象属性
 - 数据属性
 - 访问器属性
3. `Object.defineProperties()` 定义对象属性
4. 获取对象属性的描述信息
5. 构造函数与特殊属性的结合
 - `Object.defineProperties` 与 `this` 关键字的结合
6. 变量名做为属性名
7. 对象的遍历
 - 通过 `for...in` 遍历对象
 - 通过 `Object.keys()` 来遍历对象
 - 通过 `Object.getOwnPropertyNames()` 来遍历
8. 判断某一个对象是否具备某一个属性
 - `getOwnPropertyNames` 结合 `indexOf` 方式

- 使用 `in` 关键字
- `hasOwnProperty` 的方式

9. 案例与练习

◦ 根据要求创建对象

定义一构造函数Person,它有5个属性,分别是姓名,性别,年龄,生日,身份证号,

用户在创建对象的时候,这个构造函数接收二个参数,分别是姓名,和身份证号

所有的属性都不能被delete,也不能更改它的初始值

所有的属性都可以被 `for...in` 遍历出来

根据身份证号来得到用户的性别

根据身份证号来得到用户的生日(身份证号里面是包含生日的,如420984199905041016)。

提示: 字符串可以通过索引取到里面的某个值 `var str = "abc"; str[0] == "a"`

根据生日来得到年龄

根据上面定义好的构造函数去创建一个对象,然后再去遍历这个对象,打印这个对象里面的属性名与当前属性的值

面向对象 (三)

在讲解此章节的时候,一定要多使用断点调试,让同学们也学会使用浏览器的debugger功能,去查看值的变化,查看this的指向

1. this关键字
2. 对象中的this
3. 构造函数里面的this
4. 函数的不同调用形式决定this指向【重点】
 - 通过call来调用方法
 - 通过apply来调用方法
 - 通过bind来调用方式

面向对象 (四)

1. 对象的继承关系
2. 对象的包含关系
3. 作业与练习
 - 现在现个动物分别小猫Cat和小狗Dog,Cat构造函数里面有性别,年龄,昵称和体重weight四个属性,而Dog构造函数里面有性别,年龄,昵称,和身高height四个属性,猫与狗有一个共同的方法叫sleep睡觉,但是猫有一个方法 `miao`,狗有一个方法叫 `wang`
 - 现在请列举出Cat与Dog的构造函数,并且提取公共部分使用继承
 - 突然之间又有一个小动画,农家田园犬Pastoral,它也是属于狗的类别,也具备Dog所有的属性,但是它还有一个方法是看门 `Janitor`
 - 现在请创建 `Pastoral` 的构造函数,并实现继承关系

对象在内存中的存储特点

1. 对象的深拷贝

- 简单对象的深拷贝
- 数组的深拷贝
- 通过 `Object.assign()` 实现简单对象拷贝

2. JavaScript 内存结构图

3. 复杂对象的深拷贝

```
1 function copyObj(oldObj) {
2     var newObj = {};
3     var propertyNamesArr = Object.getOwnPropertyNames(oldObj);
4     propertyNamesArr.forEach(function (item, index, _arr) {
5         if (typeof oldObj[item] == "object") {
6             //null检测出来也是object, 所以要判断一下是不是null
7             if (oldObj[item] != null) {
8                 newObj[item] = copyObj(oldObj[item]);
9                 //再次判断一下, 它是不是数组
10                if(Array.isArray(oldObj[item])){
11                    //我们还要将之前拷贝对象变成一个数组
12                    newObj[item] = Array.prototype.slice.call(newObj[item]);
13                }
14            } else {
15                newObj[item] = null;
16            }
17        } else {
18            newObj[item] = oldObj[item];
19        }
20    });
21    return newObj;
22 }
23
24 var a = copyObj(classInfo);    //将对象拷贝了一份
```

4. 作业与练习

- 现有如下对象, 请将它们拷贝一份

```
1 var classInfo = {
2     className:"H2001",
3     classAddress:"湖北省武汉市",
4     students:["张三","李四","王五","赵六","男七"],
5     teacher:{
6         userName:"标哥",
7         sex:"男",
8         age:18
9     }
10 }
```

```
9     }  
10  }  
11  Object.defineProperty(classInfo,"classId",{  
12      value:"Web前端2022",  
13      writable:false,  
14      enumerable:false,  
15      configurable:false  
16  });
```

使用你所知道的任何方法，将上面对象做一次深拷贝

解题思路：首先还是获取所有对象的属性名，然后去遍历，只是在遍历的时候，要判断一下，这个属性值是否是一个数组，如果是数组，应该怎么办？如果是基本数据类型又应该怎么办？如果是对象又应该怎么办？

通过call与apply跨对象调用方法

1. call与apply跨方法调用的优点

2. 案例

- 求数组元素中的最大值【使用apply】

```
1  var arr = [1,2,6,9,4,3,8,7];
```

- 将类数组转换成数组

```
1  var obj = {  
2      0:"a",  
3      1:"b",  
4      2:"c",  
5      3:"d",  
6      4:"e",  
7      length:5  
8  }
```

请将上面的对象变成 `arr = ["a","b","c","d","e"]`

分析过程

```
1 var arr = ["a", "b", "c", "d", "e"];
2 var arr1 = arr.slice();
3 //arr在调用slice的方法, 但是arr本身没有slice的方法, 调用的是它的父级的方法
4 //arr.slice()其实就是arr.__proto__.slice()它父级的方法
5 //但是还要注意arr.slice()里面的this指调用者arr, 但是arr.__proto__.slice()它这个方法里面的this如果指向调用者就是arr.__proto__
6 //所以我们需要将arr.__proto__.slice()里面的this重新指向arr
7 //这个时候我们就使用arr.__proto__.slice.call(arr) 结果 ["a", "b", "c", "d", "e"];
```

执行上下文栈

1. 什么是执行上下文栈
2. 执行上下文栈
3. 执行上下文建立阶段以及代码执行阶段
4. 作用域
5. 自由变量
6. 作用域与执行上下文
7. 变量提权
8. 案例

```
1 function a(){
2     var age=21;
3     var height=178;
4     var weight=70;
5     function b(){
6         console.log(age);
7         console.log(height);
8         var age=25;
9         height=180;
10        console.log(age);
11        console.log(height);
12    }
13    b();
14 }
15 a();
```

执行上下文练习【周考】

1. 写出下面代码的结果

```

1  var v = 123;
2  function foo(){
3      var v = 456;
4      function inner(){
5          console.log(v);
6      }
7      return inner;
8  }
9  result = foo()
10 console.log(result())

```

2. 写出下列代码的结果

```

1  var Name='张三';
2  var Age = 18;
3  function Foo(name,age){
4      this.Name = name;
5      this.Age = age;
6      this.Func = function(){
7          console.log(this.Name,this.Age);
8          function a(){
9              console.log(this.Name,this.Age);
10             //普通调用 this指向全局的window
11         }
12         a();
13     }
14 }
15 obj = new Foo('标哥',666);
16 obj.Func()

```

3. 请定出下列代码在控制台依次打印的结果

```

1  function f1(){
2      var tmp = 1;
3      this.x = 3;
4      console.log(tmp);
5      console.log(this.x);
6  }
7  var obj = new f1();
8  console.log(obj.x)
9  f1();
10 console.log(f1());

```

4. 看代码写结果

```
1  var User = {
2      count: 1,
3      getCount: function() {
4          return this.count;
5      }
6  };
7  console.log(User.getCount());
8  var func = User.getCount;
9  console.log(func());
```

常用内置对象

1. Math对象
2. Date对象
3. 延时调用与循环调用
 - 延时调用
 - 循环调用
4. 作业与练习
 - 产生一个20~100(包含100)之间的随机数10个，把它放在一个数组里面，数组里面的随机数不重复？
 - 现有一个产品，它的保持期只有18个月，它的生产日期就是当前时间，请计算这个产生在的过期时间是什么时候？
 - 现有一个产品，它的生产日期是 2020年1月18日，它的过期时间是 2022年6月30日，请计算这个产品保质期的天数是多少？
 - 现有一个从外地回小区需要进行《新冠疫情医学隔离》，隔离时间是14天，它回来的日期是 2020年1月18日，请计算一下他什么时间可以解除隔离？【输出的日期格式必须是xxxx年xx月xx日】
 - 在第4题的基础上做一次扩展，如果结束隔离的时间是**星期六或星期天**，则**顺延到星期一**

包装对象

1. 包含对象的概念
2. String字符串对象
 - 常用方法与属性
 - length 属性
 - 索引取值
 - concat() 方法
 - slice(start?,end?) 方法
 - substring(start,end?) 方法
 - substr(start,length?) 方法
 - indexOf/lastIndexOf 方法
 - split() 方法
 - charAt(index) 方法
 - charCodeAt(index) 方法
 - startsWith() 方法
 - endsWith() 方法
 - includes() 方法
 - trim()/trimLeft()/trimRight() 方法

- `replace()` 方法
- `toUpperCase()/toLowerCase()`
- `anchor/big/small` 方法
- `padStart(maxLength,fillString)/padEnd(maxLength,fillString)` 方法

3. Number对象

- 常用方法与属性

- `toString()` 方法
- `toFixed()` 方法

4. Boolean对象

5. 案例与练习

- 已知有字符串 "`get-element-by-id`",写一个 方法将其转化成 驼峰表示法 "`getElementById`"

闭包及闭包的应用