# JS面向对象

## 课程目的：

1. 面试
2. 让大家体会什么叫面向对象编程
3. 为了后续的框架封装课程做准备 jquery dom on ajax

## 如何学习

1. 学习面向对象入门知识，学会使用面向对象方式进行开发
2. 面向对象进阶 面向对象三大特征：封装、继承、多态
3. 用面向对象角度重新认识一下整个JS世界 window document {} location

## 今天学习内容

1. 复习JS基础语法，让大家基础更加扎实
2. 了解什么叫面向对象思想
3. 学习一种最简单的对象——对象字面量
4. 使用对象字面量形式开发出一个简单的商品展示

# 复习

JS：ECMASCRIPT DOM BOM

ES6:2015 ecmascript 2015

ES5

JS如何编写

记事本 .js

.html 引用.js

JS引擎：

1. chrome v8
2. webkit edge

ie 9- ie9-11 edge

# JS语法

## 变量：存储数据的一个名称

### 1.1、基本使用

声明：var name;//\_ $ 字母数字 其中首字母不能是数字

赋值：name=”成先程”;

var name=”成先程”;

使用：

alert(name);

### 1.2、全局变量和局部变量

全局作用域：整个JS执行环境

局部作用域：通过创建一个函数就开辟出了一个局部作用域

全局变量：在全局作用域都可以访问的变量

局部变量：只能在当前局部作用域访问的

function test1(){

var name=”张三”;

console.log(name);

console.log(age);

function test2(){

var age=18;

console.log(name);//”张三”

}

}

### 1.3、变量声明提升

function test(){

a=3;

a+=1;

console.log(b);//undefined

b=2;

var a;

var b;

console.log(b);//2

}

functioin test(){

var a;

var b;

a=3;

a+=1;

}

### 1.4、变量搜索机制

a、有没有局部作用域，如果有的话，先查找是不是这个局部作用域定义的变量

b、寻找上一级作用域，，。。。。

c、找到全局作用域，如果全局作用域也找不到这个变量，这个变量就是未定义的 undefined

## 数据类型

提供6种基本数据类型(ES6之前)

2.1、数字

var age=18;

整数、浮点数：就是带有小数部分的

var pai=3.14;

2.2、字符串：

var className=”2期就业班”;

var className2=’2期’;

var c=”2’;//语法错误 syntax error

var d=’a”;//语法错误

2.3、布尔

只有2种值：true和false

常用于判断，当然也可以用于只有2种值的情况

//性别只有2种，就可以使用布尔值

//这样定义了gender，根本无法判断出男和女

var gender=true;

var gender2=”男”;

gender2=”女”；

var isNumber=true;//isNumber一眼就能看出用来判断是不是数字的

var canUse=false;//意思就是不能使用

2.4、null

2.5、undefined

出现情形：a、一个变量没有被定义

b、变量定义了没有被赋值，就直接使用

2.6、object类型

数组:

var arr=[1,2,3];

console.log(arr.length);

arr[3];//undefined

arr[5]=10;

console.log(arr.length);//6

[1,2,3,undefined,undefined,10]

正则表达式：

1、/[0-9]{10}/

“1231231230”

2、new RegExp(“[0-9]{10}”);

Regular Expression

## 运算符

### 3.1、算术运算符：

+ - \* / %（取模，求余）++ --

%：var a=10%3;//1

/：var b=10/3;//3

++：

var i=1;

i++;

console.log(i);//2

console.log(i++);2 这一步并没有完成i自加的操作

console.log(++i);//4 这一步i先进行自加，再输出i

--

### 3.2、比较运算符：> >= < <= == ===

== ：不会进行数据类型的比较，直接比较它们的值

=== ：先进行比较数据类型，再判断它们的值

console.log(null==undefined);//true

console.log(1==”1”); //true

console.log(1===”1”); //false

### 3.3、逻辑运算符

&&： 用来判断运算符左右两边表达式是不是都是true，如果都是true，结果就是true，如果不是，结果false

||：判断运算符左右两边只要有一个true，那么结果就是true，否则才是false

!：(取反)判断运算符右边表达式是不是true，如果是true，结果是false，否则就是true

以下数值是false的：

1. 数字0 NaN
2. 空字符串：””
3. null
4. undefined
5. false

### 3.4、三元运算符

简化if操作

常用于赋值

var a=10>3?”张三”:”李四”;

var a;

if(10>3){

a=”张三”;

}else{

a=”李四”;

}

var num1=1000;

var num2=”1000”:

console.log(num1+num2);//”10001000”

数据类型检测重要性

## 数据类型检测

#### typeof

typeof 10;//’number’

typeof “hello”;//”string”

typeof undefined;//”undefined”

typeof true;//”boolean”

**typeof function(){} //”function”**

typeof {};//”object”

typeof null;//”object”

typeof [1,2,5]//”object”

typeof /[abc]/; ‘object’;

typeof可以用来检测数字、字符串、布尔、函数、undefined

如果用来检测数组、null、object类型、正则就显得无能为力了

更多数据类型检测的方法

instanceof

Array.isArray();

toString.call();

## 数据类型转换

parseInt()//把一个数据转换为整数

parseInt(“123”);//123

parseInt(3.14);//3

parseInt(“123hello”);//123

parseInt(“123hello123”);//123

parseInt(true);//1

parseInt(false);//0

parseInt(“”);//NaN

parseInt(“hello”);//NaN

parseFloat();//把一个数据转换为浮点数

parseFloat(“3.14”);//3.14

parseFloat(“3.14abc”);//3.14

parseFloat (true);//1

parseFloat (false);//0

parseFloat (“”);//NaN

parseFloat (“hello”);//NaN

toString(); //把数据转换为一个字符串

null undefined;//type error

var isNan=true;

isNan.toString();//”true”

var c=123;

c.toString();//”123”

## 表达式

### 6.1、逗号表达式

常用于变量声明合并：

var a=10;

var b=”张三”;

var c=18;

var a,b,c;

a=10;

b=”张三”;,

var a=10,b=a+10,c=b+10;

console.log(c);//30

逗号表达式是从左往右依次执行

var a,b,c;

a=10;

b=a+10;//20

c=b+10;//30

### 6.2、短路表达式

var name=null || 0;

function test(a,b,c){

b=b || 0;

b.toString();//如果此时b是undefined，就会报错

}

test(1)

## 函数

#### 函数声明：

function testFn(){

console.log(“hello world”);

}

调用：

testFn();

#### 函数参数

arguments是对实参的判断

arguments.length：实参个数

arguments[索引]：获取指定索引的实参的值

function test2(name){//name是形参

//获取函数参数：arguments[0]//获取name的值

arguments[1] //23

//name=”hello”;

console.log(name);

}

test2(“hello”,23);//”hello”是实参

#### 函数返回值:

function add(num1,num2){

return num1+num2;

}

console.log(add(1,2));//3

function add2(num1,num2){

console.log(num1+num2);

}

var result=add2(2,5);

console.log(result);//result没有被赋值

函数声明提前

在一个作用域中，都会优先执行函数声明以及变量声明

#### 匿名函数：一个没有函数名称的函数

function (){}

不能通过函数名称调用，因为没有函数名称

只能自己执行自己

(function (){})()

##### 匿名函数使用场景：

1、常用于杜绝变量污染

div1.onclick=function(){

};

2、还用于只需要执行一次的函数

#### 函数表达式

fn(); //报错，此时fn还是一个没有值的变量，所以说无法当成函数来调用

var fn=function(name){}

fn(“蒋小聪”);

函数表达式不具有声明提前的特性

# 下午教学内容：

## 了解面向对象思想

## 提炼对象、属性、方法

## 3、讲解对象字面量方式进行面向对象编程

## 4、使用对象字面量开发一个商品展示页面

# ５、面向对象思想

javascript是一门面向对象语言

java c++ c#

蕴含了面向对象思想

案例１：卖煎饼：

1. 自己去煎饼摊买一个回来
2. 让同桌帮我带一个

好处：节省了时间，让我可以有更多的时间敲代码

让别人帮我完成买煎饼这么一事儿

案例２：

ａ、我第一次买手机

　１５００酷派

ｂ、买第二个手机，让我表哥在小米官网上抢购了一台小米２ｓ

区别：第二个手机是找一个懂行的人或者经验丰富的人买的手机

物有所值，没有花冤枉钱

面向对象思想就是找个人帮我完成一件事儿

# ６、对象：

万物皆对象

长城

熊猫

企鹅

北极熊

描述一个对象？

我们通常通过对象的动作和特征来描述一个对象

人：吃喝拉撒睡

动作：在软件编程中，把一个对象的动作称之为**方法，事情过程**

特征：把一个对象的特征称之为对象的**属性，一定不是事儿**

**其实所谓的面向对象的编程，其实就是提炼出一个个的对象，以及他们的属性和方法，并对他们的属性和方法进行一些操作**

**面向对象编程：ＯＯＰ：object oriented programming**

**买手机：**

**对象：表哥**

**方法：买手机**

# 7、JS使用面向对象编程

对象字面量

JSON：javascript对象表示法

javascript对象总是属性和方法的集合

var person={

name:”张三”,

age:18,

gender:”女”，

/\*表白\*/

sayLove:function(nameMale){

//买花,巧克力

//找到那个男孩纸

//在跟他说一些心里话

} //称之为对象的方法

};

千万不要实例化

正确的使用对象的属性:

person.name//”张三”

person[“name”]//”张三”

正确的获取对象的方法：

typeof person.sayLove//”function”

person.sayLove(“李四”);//实现了让张三去表白这件事

练习：用对象字面量的方法实现猫叫这件事

对象字面量

new Object();

手动给对象添加属性和方法

var cat={};

cat.color=”white”;//让猫具有了color

console.log(cat.color);//”white”

cat.say=function(){};//让猫具有了say这个方法

cat.say();//让猫叫

# 使用对象字面量开发一个商品展示页