# 对象

* 1. 什么是对象
     + 1. 在现实世界中，任何(实体/所有的东西)都可叫对象，比如“人”可看作一个对象。“人”具有姓名、性别、年龄、身高、体重等特征，“人”可以吃饭、可以开车、可以运动等动作(具有这些功能)。

一个杯子就是一个对象，杯子有什么特征：形状、用途、花纹。。。 杯子：装水；

圆、喝水、无 功能：装水。

* + - 1. 简单的讲，对象就是一组属性与方法的集合，比如：“人”的属性(特征)有身高、体重，“人”的功能有吃饭、开车、睡觉等。
      2. 对象是一种**复合数据类型**，或者称为**引用类型**。

基本数据类型只能用于描述单个数据，而引用类型是一个可以描述多个数据的集合。

比如：

基本数据类型描述数据，每个变量只能描述某个事物的一个特征；

var name = ”老板”;

var age = 18;

var height = 1.78;

引用数据类型，则具有描述事物的能力；

var user = new Object(); //创建一个对象来描述一个事物，或者一个事物在js中就是一个对象。

//以上对象描述的是一个人，那么我们就可以给这个对象添加一些车的属性（特征）

user.name = “赵菊儿”;

user.age = 18;

user.height = 1.80;

//当然对象应该拥有一些行为（方法、功能），比如：拥有一个自我介绍的方法。

user.display=function(){

console.log(“各位好!,我是”+this.name+”,今年”+this.age+”岁”);

};

对象：对象由属性和方法组成；

属性：描述对象自身的特征；比如人的姓名，猪的体重；

方法：对象具有的行为和功能，比如人可以拥有一个“自我介绍”功能；

* 1. 对象的分类
* **JavaScript（内置对象）内建对象**
  + 由JavaScript语言规范定义的一系列对象，这些对象提供了完成各种任务的方法，比如Math对象,String对象,Date对象等

var date = new Date();//日期对象

console.log(date.getSeconds());

console.log(date.getHours());

* **自定义对象**

**自定义对象：由程序员创建描述特定事物的对象；**

编程语言中，创建对象往往比现实生活中找对象来的快的多，你只需要通过**new 类型()**就ok了。

比如：var user = new Object(); //创建了一个对象

user.name = “王花花”; // 为对象添加属性（特征）

user.sex = true;

user.showInfo = function(){

//this指向user. this==user;

console.log(“我是：”+this.name+“，是一个”+this.sex?”男人”：”女人”);

}；

var myGirl = {

name:'小华',

age:18,   
height:2.8,  
run:function(){  
 alert('跑步走...');

}

};

* 1. 方法中的this

当一个函数作为对象的属性存在时，并且通过对象调用这个方法，那么函数中的this就指向调用函数的这个对象；

function sayName(){

console.log(this.name);

}

var ergou = {

name :”二狗”,

say:sayName

};

ergou.say();//二狗

var juer = {

name :”菊儿”,

say:sayName

};

juer.say();//菊儿

this指向对象的好处在于，可以更加方便的对象自身的内部成员。

* 1. 基本类型与引用类型区别
     1. 内存存储

1. 基本类型和引用类型，都是数据类型；
2. 数据的值都存储在内存中；
3. 程序在运行时，会把内存划分不同区域，其中最重要的两个区域，是栈和堆。
4. 栈中的值占用空间固定，而堆中是动态分配的内存；
5. 所有的变量都存储在栈
6. 基本数据类型的值直接存储在栈中，而引用类型的值存储在堆中，变量只是一个指向堆内存的地址引用。
7. String对象
   1. 什么是字符串对象

JS中字符串可以是基本数据类型，也可以是引用数据类型，具体是哪一种根据我们创建的方式而定；

var s1 = “aaa” ; //基本数据类型

var s2 = new String(“bbb”) //引用数据类型

var str1 = “abcd”; //简单常用的定义形式

var str2 = ‘abcd’;

var str3 = new String(“abcd”); //标准定义形成

var str4 = new String(‘abcd’);

以上都是字符串，只是不同的数据类型而已，及存储的位置也不同。

注意点：JS中基本数据类型（存储字符串值的能力）与JS引用数据类型（存储字符串值、长度、方法（功能））虽然从类型的角度来讲，他们的确不同，但是使用的时候其实差异不大，简单理解：就是说JS中的基本数据类型可以当成对应的引用类型来使用，后面章节中会有直观体会；

* 1. 为什么使用字符串对象

因为在JavaScript中为字符串对象提供了很多现成对字符串处理的方法和属性,借助于现有的方法和属性可以对对象中的字符串进行处理. 我们在工作中遇到的数据最多也是字符串类型的数据.

* 1. 字符串的属性和方法

JavaScript为字符串对象定义了很多方法和属性,通过对象可以直接是调用方法和数据对自身的字符串进行处理.

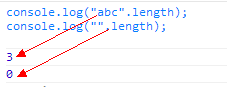
语法:

字符串对象.属性

字符串对象.方法名()

* + 1. 属性length

length属性字符串对象字符串长度;

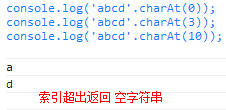


* + 1. stringObject.charAt(index)

|  |  |
| --- | --- |
| index | 必需。表示字符串中某个位置的数字，即字符在字符串中的下标。 |

返回在指定位置的字符。

注意:字符串中第一个字符的下标是0。如果参数 index 不在 0 与 string.length 之间，该方法将返回一个空字符串。



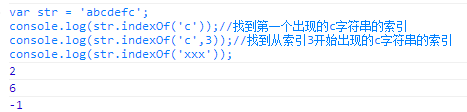
* + 1. stringObject.indexOf(searchvalue,fromindex)

|  |  |
| --- | --- |
| searchvalue | 必需。规定需检索的字符串值。 |
| fromindex | 可选的整数参数。规定在字符串中开始检索的位置。它的合法取值是 0 到 stringObject.length - 1。如省略该参数，则将从字符串的首字符开始检索。 |

方法可返回某个指定的字符串值在字符串中首次出现的位置。

注释：indexOf() 方法对大小写敏感！

注释：如果要检索的字符串值没有出现，则该方法返回 -1。



* + 1. stringObject.lastIndexOf(searchvalue,fromindex)

|  |  |
| --- | --- |
| searchvalue | 必需。规定需检索的字符串值。 |
| fromindex | 可选的整数参数。规定在字符串中开始检索的位置。它的合法取值是 0 到 stringObject.length - 1。如省略该参数，则将从字符串的最后一个字符处开始检索。 |

返回一个指定的字符串值最后出现的位置，在一个字符串中的指定位置从后向前搜索。

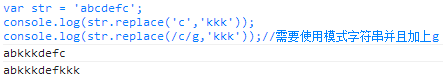


* + 1. stringObject.replace(substr,replacement)

|  |  |
| --- | --- |
| substr | 必需。需要被替换的字符串值. |
| replacement | 必需。替换后的字符串值。 |

字符串中用一些字符替换另一些字符.

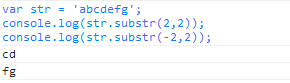
默认情况下使用replacement替换找到的第一个substr的值. 如果需要全部替换需要加上全局标志 g



* + 1. stringObject.substr(start,length)

|  |  |
| --- | --- |
| start | 必需。要抽取的子串的起始下标。必须是数值。如果是负数，那么该参数声明从字符串的尾部开始算起的位置。也就是说，-1 指字符串中最后一个字符，-2 指倒数第二个字符，以此类推。 |
| length | 可选。子串中的字符数。必须是数值。如果省略了该参数，那么返回从 stringObject 的开始位置到结尾的字串。 |

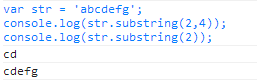
可在字符串中抽取从 *start* 下标开始的指定数目的字符。



* + 1. stringObject.substring(start,stop)

|  |  |
| --- | --- |
| start | 必需。一个非负的整数，规定要提取的子串的第一个字符在 stringObject 中的位置。 |
| stop | 可选。一个非负的整数，比要提取的子串的最后一个字符在 stringObject 中的位置多 1。  如果省略该参数，那么返回的子串会一直到字符串的结尾。 |

用于提取字符串中介于两个指定下标之间的字符。



* + 1. stringObject.toLowerCase()

把字符串转换为小写。



* + 1. stringObject.toUpperCase()

把字符串转换为大写。



* + 1. stringObject.charCodeAt()

返回在指定的位置的字符的 Unicode 编码。

var s = “我爱你”;

console.log(s.charCodeAt(1));//获取“爱”的unicode编码。

* + 1. String.fromCharCode()

可接受一个指定的 Unicode 值，然后返回一个字符串。

var s = String.fromCharCode(97,98,99);

console.log(s);

* 1. 字符串方法实例

点击上传图片按钮检测上传文件后缀是否支持指定的上传后缀



<html>

<script type=**"text/javascript"**>

**function** checkFile**(){**

**var** allowExt **=** **[**'jpg'**,**'png'**,**'gif'**,**'jpeg'**];**

//>>1.找到html中的上传文件对象

**var** file **=** document.getElementById**(**'pic'**);**

//>>2.得到上传文件的值

**var** value **=** file.value**;** //加入说 D:/xxx/yy/xxxx.jpg

//>>3.获取文件的后缀名,先找到.的位置

**var** pos **=** value.lastIndexOf**(**'.'**);**

//>>4.截取.后面的jpg,将后缀名统一小写,因为上传的后缀名可能有大写的

**var** ext **=** value.substr**(**pos**+**1**).**toLowerCase**();**

//>>5.检测后缀名是否在数组中

**var** tag **=** **false;**

**var** len **=** allowExt.length**;**

**for(var** i **=** 0**;** i **<** len**;**i**++){**

//使用数组中的每个元素和文件后缀做对比

**if(**allowExt**[**i**]==**ext**){**

//只要对比上说明文件后缀已经出现在数组中,将tag设置为true

tag **=** **true;**

**break;** //就没有必要继续循环了

**}**

**}**

**if(**tag**){**

alert**(**ext**+**'后缀名支持上传'**);**

**}else{**

alert**(**ext**+**'后缀名不支持上传'**);**

**}**

**}**

</script>

</head>

<body>

<!--id在当前文档中代表唯一的表示-->

<input type=**"file"** id=**'pic'**/><br/>

<!--当点击上传按钮触发方法,将方法加到onclick上-->

<input type=**"submit"** value=**'上传图片'** onclick=**"checkFile()"**/><br/>

</body>

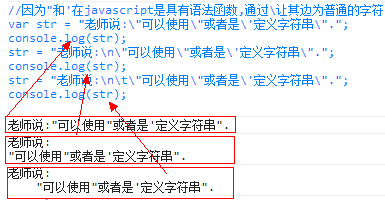
</html>

* 1. 特殊字符串

**你可以在 JavaScript 中使用反斜杠来向文本字符串添加特殊字符。**

|  |  |
| --- | --- |
| **代码** | **输出** |
| \' | 单引号 |
| \" | 双引号 |
| \& | 和号 |
| \\ | 反斜杠 |
| \n | 换行符 |
| \r | 回车符 |
| \t | 制表符 |
| \b | 退格符 |
| \f | 换页符 |

以上特殊的字符串是不能够单独存在在字符串中,需要通过\来转义.



1. Number对象

**查阅文档学习**

|  |  |  |  |
| --- | --- | --- | --- |
| [toFixed](mk:@MSITStore:E:\\WEB前端\\文档w3capi\\w3c官方教程.chm::/www.w3school.com.cn/js/jsref_tofixed.asp.htm) | 把数字转换为字符串，结果的小数点后有指定位数的数字。 | 1.0 | 5.5 |

1. Math对象

JavaScript自身定义了很多数学运算的方法被定义在Math对象中. Math自身就是一个对象无需创建直接使用自身中的方法和属性

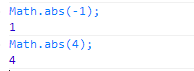
* 1. Math的属性PI

PI属性代表圆周率



* 1. Math.abs(x)

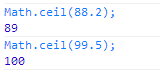
返回数x的绝对值。

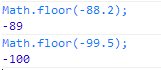


console.log是可以省略的.

* 1. Math.ceil(x)

对一个数x进行上舍入

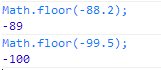




* 1. Math.floor(x)

对一个数x进行下舍入

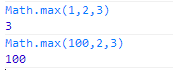




* 1. Math.max(x)

返回指定数中带有较大的值的那个数。

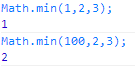
…表示可以传递无限个参数



* 1. Math.min(x,y)

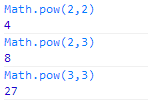
返回指定数中带有较小的值的那个数。

…表示可以传递无限个参数



* 1. Math.pow(x,y)

返回 x 的 y 次幂的值。



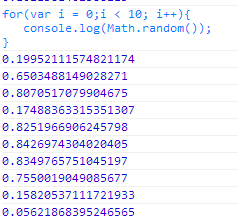
2\*2 = 4

2\*2\*2 = 8

3\*3\*3 = 27

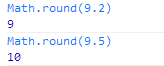
* 1. Math.random()

返回介于 0 ~ 1 之间的一个随机数,包含0不包含1。



* 1. Math.round(x)

可把一个数字舍入为最接近的整数。(四舍五入)



* 1. 获取指定范围的随机整数

获取指定范围的随机整数的方法,该方法不是系统的方法是系统,该方法必须记住写法,后面经常应用.

**function** randomInt**(**min**,**max**){**

**return** Math.floor**(**Math.random**()\*(**max**-**min**+**1**)+**min**);**

**}**

//需求:获取10到30之间的随机整数

//逆向分析法:

//>>1. 如果能够获取0到20的随机数,再随机上加10

//>>2. 如果能够获取0到1的随机数,再乘以20可以实现,

//因为通过Math.random()生成的随机数不包含1. 所以说在20的基础上加一

//即 Math.random()\*(20+1) 计算出来的值0--21之间(不包含21)

//返过来分析

//>>1. Math.random()\*(20+1) 计算出来的值0--21之间(不包含21),要得到10--30

//>>2. 在以上的基础上加10 即: Math.random()\*(20+1)+10, 得到10--31之间(不包含31)

//>>3. 在 Math.random()\*(20+1)+10公式中需求给出没有20,实际上20可以通过30-10得到

// 即: Math.random()\*(30-10+1)+10 (10--31之间的数),但不包含31.并且我们需要的是整数

//>>4. Math.floor(Math.random()\*(30-10+1)+10),得到10到30之间的整数

//>>5. 如果将以上需求中的10,30分别作为需求的最小数min,最大数max表示的话,

//得到以下公式:

//Math.floor(Math.random()\*(max-min+1)+min)

//>>6.可以将该公式看作获取指定范围随机整数的方法如上方法:

1. Date对象

Date 对象用于处理日期和时间。

* 1. 创建Date对象
     + 1. **new Date()**
       2. **new Date(***dateVal***)**
       3. **new Date(***year*, *month*, *date*[**,** *hours*[**,** *minutes*[**,** *seconds*[**,***ms*]]]]**):**创建date代表指定当前系统时间  
          
       4. **new Date(***dateVal***):**根据dateVal指定的时间创建date对象  
          dateVal可以是字符串的类型的参数:  
          

字符串的格式:  
短日期可以使用“/”或“-”作为日期分隔符，但是必须用月/日/年的格式来表示，例如"7/20/96"。  
小时、分钟、和秒钟之间用冒号分隔，尽管不是这三项都需要指明。"10:"、"10:11"、和 "10:11:12" 都是有效的。

dateVal也是数字类型的参数,数字类型的参数代表毫秒值,该毫秒值是从1970 年 1 月 1 日 到指定日期的毫秒数.



* + - 1. **new Date(***year*, *month*, *date*[**,** *hours*[**,** *minutes*[**,** *seconds*[**,***ms*]]]]**)**

根据指定年月日时分秒创建date对象



注意month中的值从0开始, 0—11代表12个月

* 1. Date对象的常用方法
     1. dateObject.getFullYear()

可返回一个表示年份的 4 位数字。



* + 1. dateObject.getMonth()

可返回表示月份的数字。

使用本地时间。返回值是 0（一月） 到 11（十二月） 之间的一个整数。



* + 1. dateObject.getDate()

可返回月份的某一天。

使用本地时间。返回值是 1 ~ 31 之间的一个整数。



* + 1. dateObject.getHours()

可返回时间的小时字段。

以本地时间显示。返回值是 0 （午夜） 到 23 （晚上 11 点）之间的一个整数。



* + 1. dateObject.getMinutes()

以本地时间显示。返回值是 0 ~ 59 之间的一个整数。



* + 1. dateObject.getSeconds()

可返回时间的秒。

以本地时间显示。返回值是 0 ~ 59 之间的一个整数。



* + 1. dateObject.getMilliseconds()

可返回时间的毫秒。

以本地时间显示。返回值是 0 ~ 999 之间的一个整数。



* + 1. dateObject.getDay()

可返回表示星期的某一天的数字。

使用本地时间。返回值是 0（周日） 到 6（周六） 之间的一个整数。



* + 1. dateObject.getTime()

可返回距 1970 年 1 月 1 日之间的毫秒数。



* + 1. dateObject.toLocaleString()

可根据本地时间把 Date 对象转换为字符串，并返回结果。



以上的方法是获取时间中的内容.Date对象也提供了设置Date对象内容的方法

|  |  |
| --- | --- |
| [setDate()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setDate.asp.htm) | 设置 Date 对象中月的某一天 (1 ~ 31)。 |
| [setMonth()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setMonth.asp.htm) | 设置 Date 对象中月份 (0 ~ 11)。 |
| [setFullYear()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setFullYear.asp.htm) | 设置 Date 对象中的年份（四位数字）。 |
| [setYear()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setYear.asp.htm) | 请使用 setFullYear() 方法代替。 |
| [setHours()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setHours.asp.htm) | 设置 Date 对象中的小时 (0 ~ 23)。 |
| [setMinutes()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setMinutes.asp.htm) | 设置 Date 对象中的分钟 (0 ~ 59)。 |
| [setSeconds()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setSeconds.asp.htm) | 设置 Date 对象中的秒钟 (0 ~ 59)。 |
| [setMilliseconds()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setMilliseconds.asp.htm) | 设置 Date 对象中的毫秒 (0 ~ 999)。 |
| [setTime()](mk:@MSITStore:E:\\文档\\电子书\\w3school-2014.CHM::/jsref_setTime.asp.htm) | 以毫秒设置 Date 对象。 |

* 1. 时间案例
     + 1. 计算一个人活了多少天?

/\*\*

@param birthday: 指定出生日期

@return 截止当前时间出生天数

\*/

**function** f**(**birthday**){**

//>>1.生日时间对象

**var** d1 **=** **new** Date**(**birthday**);**

//>>2.当前时间对象

**var** d2 **=** **new** Date**();**

//>>3.得到生日时间距1970.07.07的毫秒值

**var** t1 **=** d1.getTime**();**

//>>3.得到当前时间距1970.07.07的毫秒值

**var** t2 **=** d2.getTime**();**

**return** Math.round**((**t2**-**t1**)/**1000**/**60**/**60**/**24**)**

**}**

**var** days **=** f**(**'1980/06/26'**);**

alert**(**days**);**

* + - 1. 计算一个人活到60岁还需要多少天?

/\*\*

@param birthday: 指定出生日期

@param num: 指定岁数

@return 截止当前时间出生天数

\*/

**function** f1**(**birthday**,**num**){**

//>>1.生日时间对象

**var** d1 **=** **new** Date**(**birthday**);**

//>>2.设置岁数

**var** year **=** d1.getFullYear**();**

year**+=**num**;**

//>>3.设置num岁后的日期

d1.setYear**(**year**);**

**var** d2 **=** **new** Date**();**

//>>4.未来num岁后距1970.07.07的毫秒值

**var** t1 **=** d1.getTime**();**

//>>5.得到当前时间距1970.07.07的毫秒值

**var** t2 **=** d2.getTime**();**

**return** Math.round**((**t1**-**t2**)/**1000**/**60**/**60**/**24**)**

**}**

**var** days **=** f1**(**'1980/06/26'**,**60**);**

alert**(**days**);**

**附录（脏话过滤关键词）：**

最毒的脏话骂人经典句子（热门篇）

　　1、 这么不要脸，这么没心没肺，你的体重应该会很轻吧?

　　2、 我没认识你之前，我真没发现原来我有以貌取人这毛病。

　　3、 你他妈就是被强奸避孕失败生出你这个没闭眼的畜生

　　4、 这么多年谁一直照顾你啊?我很佩服他的胆量。

　　5、 你出生时是不是被扔上去过3次而只被接住过2次?

　　6、 真想立马把我37码的鞋bia到你42码的脸上

　　7、 你是个可爱的，迷人的辛勤的，白色的小小的总在为人民服务的新型杂交鱼种

　　8、 长的很科幻，长的很抽象!

　　9、 大哥，把你脸上的分辨率调低点好吗?

　　10、 你的长相突破了人类的想象...

　　11、 你妈生你时是不是把人扔了把胎盘养大

　　12、 别跟我说话，我有洁癖。

　　13、 我忍屎忍尿也忍不下你呀。

　　14、 一巴掌把你打到墙上扣都扣不下来。

　　15、 你的牙如同天上的繁星，色泽鲜艳，相距甚远。

　　16、 这两片嘴唇，切切倒有一大碟子。

　　17、 听说你傍大款了，认二郎神当主人了。

　　18、 你玩劈腿，劈那么开，不怕蛋蛋受凉啊

　　19、 别人要说，我c你妈，你可以回答我阉你爸。

　　20、 别人骂你装13，你可以回，恩，你真13。

　　21、 别人骂你吵，你回说，我炒shi给你吃啊。

　　22、 你妈带你逛街，别人都问：大姐，你这猴多少钱买的。

　　23、 瞅你长得姥姥不疼，舅舅不爱的。

　　24、 jian人永远都是jian人，就算经济危机了，你也贵不了。

　　25、 请你卷成一团圆润的离开。

　　26、 有病你治病，你别找我啊，我又不是兽医。

　　27、 你内张脸长地比盆骨都标志。

　　28、 一脸兴冲冲的，跟喝了尿糖似的。

　　29、 当初你傲成那个样子，现在又是玩哪出呢。

　　30、 我给狗扔块骨头它都知道冲我摇摇尾巴，你算什么啊。

　　31、 把你的眼屎擦掉了看清楚是谁说话再行不。

　　32、 麻烦你看清楚什么叫货在说话好吗。

　　33、 我怎么敢碰你呢，我怕我买洗手液买穷自己。

　　34、 缺狗粮才知道来找我?

　　35、 别以为你名门什么的，你以为你爸是李刚啊。

　　36、 别人若是骂你，看p啊or看毛啊，你可以回，看你呢啊。

　　37、 你妈生你的时候调成了随机状态了吧。

　　38、 你爸是731部队的吧，研究病毒没研究明白，把你研究出来了。

　　39、 你的长相和智商都挺哈韩的。

　　40、 别以为你长的稀有样我们就应该物以稀为贵。

最毒的脏话骂人经典句子（个性篇）

　　1、你他妈整个脑残了，不惜拿自己的父母家人来骗别人的精液。

　　2、跟一个你视为他为狗的人将素质，你真心逗B，脑子也秀逗了吧!

　　3、贱人的生活方式就是，只许自己吃屎，不许别人放屁。

　　4、你爹有点多，我忍不住笑了下，不好意思啊，你不要去想你那悲伤的母亲。

　　5、我们骂不骂关你个婊子什么事，草你吗的瞎眼白内胀!

　　6、你他妈那么喜欢被人擦，我想说我退出我不喜欢兽交。

　　7、我只是你祖宗而已，你怎么可以这样，你个傻逼怎么能把你祖宗都忘记了?

　　8、你要是不扒一扒脸皮，我还真不晓得原来你的脸皮如此的厚呢!

　　9、对不起哈，当年老子我没忍住，一个屁把这二逼给蹦出来了!

　　10、你是脑子有病还是怎么?你以为你是谁?不就是傻逼一个嘛!

　　11、整天把逼脸摆给别人看，是想要告诉别人你是黄世仁他妈吗?

　　12、你是不是每次拉完屎，然后又自己舔干净了，还到处炫耀自己拉屎很香。

　　13、你妈是靠卖逼把你养大的吧，到你这代还能继续卖逼，真的是母业女继承啊。

　　14、是用屁眼子打字的?你是你麻痹，说这个说那个还以为你是一个野种孤儿呢。

　　15、傻B卵蛋、脑残孙子，有种去外星球射啊，活在地球上太浪费了。

　　16、你他妈妈卖批能像个人样么?学什么不好偏偏要学下贱。

　　17、整体做些缺德的事，小心遭报应被雷劈，脑残二货。

　　18、你个畜生级别的，还轮不到你在人界指手画脚。

　　19、瞧你那长相就知道你是一笔袋，也只能用来装笔了。

　　20、怪只怪我太年轻，是人是狗都分不清!

　　21、就你这飞禽走兽的长相，国家没给你颁个奖实在是太可惜了。

　　22、别整天显着那菊花脸样的淡定，让人给你操两下你就会很爽了吧!

　　23、从猴子变成人需要成千上万年，而你从人变成猴子也就立马的事情。

　　24、你他妈是不是结婚当晚，初夜让人用肚脐眼给骗了?

　　25、像你这种贱人就是明骚易躲，暗箭难防了!

　　26、小贱人每天都自拍，真担心哪天你的手机会愤怒到自曝而亡。

　　27、瞧来瞧去漫山遍野的傻逼，原来你也在这里!

　　28、就算你用了伟哥，照样是直不起的料，注定此生不性福啊!