**第一章**

1. javascript引入方式

答：1、嵌入式 2、外链式 3、行内式

二、注释

答：1、单行注释“//”2、多行注释“/ \* \* /”

1. 标识符

答：1、由字母、数字、下划线和美元符号（$）组成 2、不能以数字开头

3、严格区分大小写 4、不能使用javascript中的关键字命名

5、要尽量要做到“见其名知其意”

**第二章**

关键字p22 变量的使用p23

一、数据类型分类

数据类型：

|  |  |
| --- | --- |
| 基本数据类型 | Boolean(布尔型) |
| String（字符型） |
| Number(数值型) |
| Null（空型） |
| Undefined(未定义型) |
| 引用数据类型 | Object(对象) |

二、数值类型

|  |  |  |  |
| --- | --- | --- | --- |
| 待转数据 | Number() | parseInt() | parseFloat() |
| 纯数字字符串 | 转成对应的数字 | 转成对应的数字 | 转成对应的数字 |
| 空字符串 | 0 | NaN | NaN |
| 数字开头的字符串 | NaN | 转成对应的数字 | 转成对应的数字 |
| 非数字开头的字符串 | NaN | NaN | NaN |
| null | 0 | NaN | NaN |
| undefined | NaN | NaN | NaN |
| false | 0 | NaN | NaN |
| true | 1 | NaN | NaN |

三、isNaN()函数

答：可以通过isNaN()函数来判断一个值是否是NaN，如果该值是NaN则返回true，否则返回false。(判断输入的内容是否为数字)

四、隐式转换

答：1、+可以对两个值进行加法运算，并将结果返回，如果对两个字符串进行加法运算，则会做拼串，会将两个字符串拼接为一个字符串，并返回，任何的值和字符串做加法运算，都会先转换为字符串，然后再和字符串做拼串的操作。

2、-可以对两个值进行减法运算，并将结果返回，可以对两个值进行乘法运算

3、\*可以对两个值进行乘法运算  
4、/可以对两个值进行除法运算  
5、%取模运算（取余数）

我们可以利用这一特点，来将一个任意的数据类型转换为String，我们只需要为任意的数据类型+一个" "即可将其转换为String，这是一种隐式的类型转换，由浏览器自动完成，实际是它也是调用了String()函数。任何值做- \* / 运算时都会自动转换为Number，我们可以利用这一特点做隐式的类型转换，可以通过为一个值-0 \*1 /1来将其转换为Number，原理和Number()函数一样，使用起来更加简单

五、比较运算符

答：

1. 相等运算符用来比较两个值是否相等，如果相等返回true，否则返回false。  
   2、使用==来做相等的运算，当使用==来比较两个值时，如果值的类型不同，则会自动进行类型转化，将其转换为相同的类型， 然后再比较。  
   3、不相等，不相等用来判断两个值是否不想等，如果不相等返回true，否则返回false，使用!=来做不相等的运算，不相等也会对变量进行自动的类型转换，如果转换后相等它也会返回false。
2. ===全等，用来判断两个值是否全等，它和相等类似，不同的是它不会自动进行类型转换，如果两个值的类型不同，直接返回false。  
   5、!==不全等，用来判断两个值是否不全等，和不等类似，不同的是它不自动进行类型转换，如果两个值的类型不同，直接返回true。

三元运算符P33

第三章

1. 创建数组

答：

1、创建一个数组：var arr = new Array();  
2、使用字面量来创建数组 语法:[] var arr = [];

二、遍历数组

答：

例一：

//创建一个数组  
 var arr = ["孙悟空","猪八戒","沙和尚","唐僧","玉兔精","白骨精"];  
 for(var i = 0 ; i < arr.length ; i++){  
 console.log(arr[i]);  
 }

例二：

//创建一个数组  
var arr = ["孙悟空","猪八戒","沙和尚","唐僧","玉兔精","白骨精"];  
for(var i in arr){  
 console.log(arr[i]);  
}

例三：

//创建一个数组  
var arr = ["孙悟空","猪八戒","沙和尚","唐僧","玉兔精","白骨精"];  
for(var i of arr){  
 console.log(i);  
}

1. 栈和队列的相关方法
2. push()，该方法可以向数组的末尾添加一个或多个元素，并返回数组的长度，可以将要添加的元素作为方法的参数传递，这样这些元素将会自动添加到数组的末尾，该方法会将数组新的长度作为返回值返回。
3. pop()，该方法可以删除数组的最后一个元素,并将被删除的元素作为返回值返回。
4. unshift()，向数组开头添加了一个或多个元素，并返回新的数组长度，向前边插入元素以后，其他的元素索引会依次调整。
5. shift()，可以删除数组的第一个元素，并将被删除的元素作为返回值返回。

四、检索方法

1、includes():用于确定数组中是否含有某个元素，含有返回true，否则返回false。

2、Array.isArray():用于确定传递的值是否是一个Array，是返回true，不是返回false。

3、indexOf():返回在数组中可以找到给定值的1个索引,如果不存在，则返回-1。

4、lastIndexOf():返回指定元素在数组中的最后一个的索引，如果不存在则返回-1。

五、其他方法

1、 join()，该方法可以将数组转换为一个字符串，该方法不会对原数组产生影响，而是将转换后的字符串作为结果返回，在join()中可以指定一个字符串作为参数，这个字符串将会成为数组中元素的连接符。

2、toString(),返回一个字符串，表示指定的数组及其元素。

3、sort(),对数组的元素进行排序，并返回数组。

4、fill(),用一个固定值填充数组中指定下标范围内的全部元素。

5、 revese(), 该方法用来反转数组（前边的去后边，后边的前边。

6、splice(),对一个数组在指定下标范围内删除或添加元素。

7、slice(),从一个数组的指定下标范围内拷贝数组元素到一个新数组中。

8、concat()可以连接两个或多个数组，并将新的数组返回。

第四章

一、作用域

作用域指一个变量的作用的范围，在JS中一共有两种作用域：  
 1、全局作用域， 直接编写在script标签中的JS代码，全都在全局作用域，全局作用域在页面打开时创建，在页面关闭时销毁，在全局作用域中有一个全局对象window，它代表的是一个浏览器的窗口，它由浏览器创建我们可以直接使用。在全局作用域中：创建的变量都会作为window对象的属性保存，创建的函数都会作为window对象的方法保存，全局作用域中的变量都是全局变量，在页面的任意的部分都可以访问的到。  
 2.函数作用域，调用函数时创建函数的作用域，函数执行完毕以后，函数作用域销毁，每调用一次函数就会创建一个新的函数作用域，他们之间是互相独立的，在函数作用域中可以访问到全局作用域的变量，在全局作用域中无法访问到函数作用域的变量，当在函数作用域操作一个变量时，它会在自身作用域中寻找，如果有就直接使用， 如果没有则向上一级作用域中寻找,直到找到全局作用域，如果全局作用域中依然没有找到，则会报错ReferenceError，在函数中要访问全局变量可以使用window对象。

第五章

1. 对象的定义

对象属于一种复合的数据类型，在对象中可以保存多个不同数据类型的属性  
 对象的分类：

1、内建对象,由ES标准中定义的对象，在任何的ES的实现都可以使用，比如：Math String Number Boolean Function Object......  
 2、宿主对象，由JS的运行环境提供的对象，目前来讲主要指由浏览器提供的对象，比如BOM DOM  
 3、自定义对象，由开发人员自己创建的对象

创建对象：使用new关键字调用的函数，是构造函数constructor，构造函数是专门用来创建对象的函数，使用typeof检查一个对象时，会返回object。  
var obj = new Object();

在对象中保存的值称为属性,向对象添加属性,语法：对象.属性名=属性值;  
向obj中添加一个name属性  
obj.name = "孙悟空";  
向obj中添加一个gender属性  
obj.gender = "男";  
向obj中添加一个age属性  
obj.age = 18;

使用对象字面量来创建一个对象  
var obj = {};

1. 构造函数

答：创建一个构造函数，专门用来创建Person对象的，构造函数就是一个普通的函数，创建方式和普通函数没有区别，不同的是构造函数习惯上首字母大写。  
构造函数和普通函数的区别就是调用方式的不同， 普通函数是直接调用，而构造函数需要使用new关键字来调用  
构造函数的执行流程  
1.立刻创建一个新的对象2.将新建的对象设置为函数中的this，在构造函数中可以使用this来引用新建的对象3.逐行执行函数中的代码4.将新建的对象作为返回值返回  
使用同一个构造函数创建的对象，我们称为一类对象，也将一个构造函数称为一个类，我们将通过一个构造函数创建的对象，称为是该类的实例。

1. 函数中的this指向
2. 当以函数的形式调用时，this就是window
3. 当以方法的形式调用时，谁调用方法this就是谁

3.当以构造函数的形式调用时，this就是新创建的那个对象

四、String对象的常用属性和方法

length属性，可以用来获取字符串的长度  
console.log(str.length);console.log(str[5]);

charAt()，可以返回字符串中指定位置的字符，根据索引获取指定的字符

var str = "中hello Atguifu";var result = str.charAt(6);

indexOf()，该方法可以检索一个字符串是否含有指定内容，如果字符串中含有该内容，则会返回其第一次出现的索引，如果没有找到指定的内容，则返回-1，可以指定一个第二个参数，指定开始查找的位置

str = "hello atguigu";result = str.indexOf("h",1);  
  
lastIndexof(), 该方法的用法和indexof()一样,不同的是indexOf是从前往后找，而lastIndexOf是从后往前找,也可以指定开始查找的位置  
str = "hello atguigu";result = str.lastIndexOf("h",5);

substring(),可以用来截取一个字符串，可以sllice()类似。  
参数：第一个：开始截取位置的索引（包括开始位置）,第二个：结束位置的索引（不包括结束位置）,不同的是这个方法不能接受负值作为参数,如果传递了一个负值，则默认为0,而且他还自动调整参数的位置，如果第二个参数小于第一个，则自动交换。  
result = str.substring(1,-1);  
substr()，用来截取字符串，参数：1.截取开始位置的索引2.截取的长度  
str = "abcdefg";result = str.substr(3,2);

toUpperCase()， 将一个字符串转换为大写并返回  
str = "abcdefg";result = str.toUpperCase();  
toLowerCase()， 将一个字符串转换为小写并返回  
str = "ABCDEFG"; result = str.toLowerCase();console.log(result);  
 split()，可以将一个字符串拆分为一个数组，参数：需要一个字符串作为参数，将会根据该字符串去拆分数组  
str = "abc,bcd,efg,hig";result =str.split("d")；  
如果传递一个空串作为参数，则会将每个字符都拆分为数组中的元素  
result =str.split(" ");console.log( Array.isArray(result));console.log(result[0]);

五、Number对象

MAX\_VAULUE在javascript中所能表示的最大数值（静态成员）

MIN\_VALUE 在javascript中所能表示的最小正值（静态成员）

toFixed(digits)使用定点表示法来格式化一个数值

六、Math对象  
Math，Math和其他的的对象不同，它不是一个构造函数，它属于一个工具类不用创建对象，它里边封装了数学运算相关的属性和方法，比如：Math.PI表示的圆周率。  
 console.log(Math.PI);   
 abs()可以用来计算一个数的绝对值  
 console.log(Math.abs(-1));  
 Math.ceil（）可以对一个数进行向上取整，小数为只有有值就自动进1  
 Math.floor() 可以对一个数进行向下取整,小数部分会被舍掉  
 Math.round()可以对一个数进行四舍五入的取整  
 console.log(Math.ceil(1.4));console.log(Math.floor(1.2))；console.log(Math.round(1.4));  
 Math.random()，可以用来生成一个0-1之间的随机数，生成一个0-10的随机数，生成一个0-x之间的随机数，Math.round(Math.random() x);生成一个1 - 10的 生成一个x-y之间的随机数Math.round(Math.random() (y - x)))+x  
 for(var i = 0 ; i< 100 ;i ++){  
 console.log(Math.round(Math.random() 10));console.log(Math.round(Math.random() 20));console.log(Math.round(Math.random() 9)+1);  
 console.log(Math.round(Math.random() 8)+2);  
 生成1-6之间的随机数  
 console.log(Math.round(Math.random() 5)+1);  
 }   
 max()可以获取多个数中的最大值  
 min()可以获取多个数中的最小值  
 var max = Math.max(10,40,50,100);  
 var min = Math.min(10,40,50,100);  
 console.log(min);  
 Math.pow(x,y);返回x的y次幂  
 console.log(Math.pow(2,3));  
 Math.sqrt()，用于对一个数进行开放运算  
 console.log(Math.sqrt(9));

七、Date对象

getFullYear()：获取表示年份的4位数字

setFullYear(value)：设置年份

getMonth()： 获取月份，范围0～11

setMonth(value)：设置月份

getDate()：获取月份中的某一天，范围1- 31

setDate(value)：设置月份中的某一天

getDay()：获取星期，范围0-6

getHours()：获取小时数，返回0-23

getMinutes():获取分钟数，范围0-59

setMinutes(value):设置分钟数

getSeconds:获取秒数，范围0-59

setSeconds(value):设置秒数

getMilliseconds():获取毫秒数，范围0～999

setMilliseconds():设置毫秒数

getTime()：获取从1970-1-1 00:00:00距离Date对象所代表时间的毫秒数

getTime(value)：通过从1970-1-1 00:00:00计时的毫秒数来设置时间

1. 原型链

原型prototype  
我们创建的每一个函数，解析器都会向函数中添加一个属性prototype，这个属性对应着一个对象，这个对象就是我们所谓的原型对象。如果函数作为普通的函数调用prototype没有任何作用。当函数通过以构造函数调用时，它所创建的对象中都会有一个隐含的属性，  
指向该构造函数的原型对象，我们可以通过\_proto\_来访问该属性，原型对象就相当于一个公共的区域，所有同一个类的实例都可以访问到这个原型对象，我们可以将对象中共有的内容，统一设置到原型对象中， 当我们访问对象的一个属性或方法时，它会先在对象自身中寻找，如果有则直接使用，如果没有则会去原型对象中寻找，如果找到则直接使用。  
以后我们创建构造函数时，可以将这些对象共有的属性和方法，统一添加到构造函数的原型对象中，这样不用分别为每一个对象添加，也不会影响到全局作用域，就可以使每个对象都具有这些属性和方法了。

第六章  
一、BOM对象

Window（核心对象）：document(DOM)、history、location、navigator、screen

P127弹出对话框的窗口相关的属性和方法

1. 输入对话框

答：例：var str = prompt(“请输入测试的选项”)；

三、确认对话框

答：通过confirm()方法实现

1. 定时器

答：setTimeout() ：在指定的毫秒数后调用函数或执行一段代码

setInterval() ：按照指定的周期（以毫秒计）来调用函数或执行一段代码

clearTimeout() ：取消由setTimeout()方法设置的定时器

clearInterval() ：取消由setInterval()设置的定时器

1. 对象

答：

|  |  |  |  |
| --- | --- | --- | --- |
| location对象 | 方法 | reload（） | 重新载入当前文档 |
| 属性 | href | 返回完整的URL |

|  |  |  |  |
| --- | --- | --- | --- |
| history对象 | 属性 | length | 返回历史列表中的网址数 |
| 方法 | back() | 加载hisroty列表前中的前一个URL |
| forward() | 加载hisroty列表前中的下一个URL |
| go() | 加载hisroty列表前中的某个具体页面 |

|  |  |  |  |
| --- | --- | --- | --- |
| navigator对象 | 属性 | appName | 返回浏览器的名称 |
| userAgent | 返回由客户端发送服务器的User-Agent头部的值 |

第七章

一、p149DOM HTML节点树