js 基本语法汇总

1、分类

ECMAScript js 基本语法与标准

DOM Document Object Model 文档对象模型

BOM Browser Object Model 浏览器对象模型

tips：DOM 和 BOM 都是一套 API（Application programing interface）

2、注释方式

```
style   /*  */body    <!-- --!>
script  //
        /* */
        /**
         *   js 说明文档注释
         */
```

3、简单指令

alert(""); 提示框；confirm(""); 确认框，点击后会响应返回 true 或 false； prompt(); 弹出一个输入框；
document.write("");console.log(""); 在控制台打印相应的信息；
console.dir(""); 在控制台打印出该对象的所有信息；

4、变量命名

数字（0-9）、字母（a-z，A-Z）、下划线（_）；　　tips:应避免保留字和关键字；

5、NaN 和 isNaN

isNaN(number),如果 number 不是数字，则为 true；Number(number),在转换为数字类型时，若 number 不是数字，则返回 NaN；

6、转义字符

\

\r　回车

\n　空格

\t　缩进

\\　反斜杠

7、逻辑短路、逻辑中断

true || 6;　　逻辑或短路，左边为 ture 返回右值；6　&&　true;　逻辑与短路，左边 false 返回右值；

8、优先级

```
* / %
+   -
&&
||
?
```

tips：自上而下优先级越来越高

9、类型转换（type）

parseInt("12a3"); 转为数字，尝试强转；parseFloat("123.123");

data.toString();String(data);

tips:变量声明未赋值，其值为 undefined;

对象为空，其值为 null;

10、三元表达式

eg ： a>b?a=1:a=2;

格式:

判断条件？true 的时候执行的操作：false 的时候执行的操作;

11、数组 Array

（1）、定义法

构造函数:

```
var arr = new Array("123","abc","xxx");
```

字面量:

```
var arr = ["123","646","abc"];
```

数组长度:

```
var arr = new Array(6);(数组长度为 6);
```

（2）、赋值

```
arr[0]=1;
```

12、形参和实参

定义函数时，function funcA(a,b,c){}，其中的 a、b、c 即为形参；

调用函数时，funcA(1,2,3);其中的 1、2、3 即为实参；

tips：function 里面有一个 arguments 对象，里面存有所有传进函数的实参；

13、函数 function

（1）、函数命名

1、 可以使用字符、数字、下划线、$；

2、 不能以数字开头；

3、 不能使用关键字和保留字；

4、 区分大小写；

5、 建议要有意义 -- 动词+名字结构；

6、 驼峰命名法；

7、 函数名不能重名，后面写的重名函数会把前面写的函数给覆盖掉；

（2）、函数的返回值

返回值：

当函数执行完毕之后，所得到的结果就是一个函数返回值

任意函数都有返回值

1、 在函数内部没有显示的写有 return 的时候，函数的返回值是 undefined；2、 当函数内部有 return，但是 return 后面没有跟着任何内容或者数据的时候，

函数的返回值是 undefined，并且 return 后面的代码不会执行；3、 当 return 后面跟着内容或者数据的时候，函数的返回值就是这个跟着的内容或者数据；

（3）、函数的四种形式：

1、没有参数，没有 return；

通常在于封装一段过程；

2、没有参数，有 return；

通常用于内部封装引用其他函数（闭包，回调）；

3、有参数，没有 return；

通常用于执行操作的封装；

4、有参数，有 return；

常见形式；

（4）、匿名函数

匿名函数的 name 属性值为 anonymous；

函数仅用一次的情况，即用即废；

eg:

```
setTimeout(function(){
    console.log(this.name);
```

```
},1000);
```

tips:在 1 秒后在控制台打印出本函数的名称；

（5）、回调函数

在一个函数当中，另一个函数作为参数传入该函数中，另一个的这个函数即为回调函数；

eg:

```
function atack(callback){
    return callback;
}
```

tips:在调用该函数时，指定 callback 是哪个函数；

```
atack（func）;
```

（6）、短路运算

作用：防止传入函数的数据不足，造成无法运行；

eg:

```
function getResult(a,b,fn) {
    fn && fn();
}
```

（通常使用逻辑与的短路来决定是否执行回调函数；）

```
function getResult_2(a,b){
    a || 0;
}
```

（通常用逻辑或的短路来防止实参不足的情况，强行赋值；）

（7）、自执行函数

```
(function func2(){


})()
```

tips:在函数定义的结束最后写入一个（），该函数定义完成后直接被调用执行；

（8）、递归

在函数执行的最后再一次的调用自身；

tips:递归是一种非常耗资源的做法，通常为了简化运算，还会结合缓存进行；

并且注意，递归必须要有结束判断条件（if），否则该函数被调用后就是死循环；

14、数据类型

（1）、简单数据类型

string、number、boolean

（2）、复杂数据类型

String、Number、Boolean、Array、Math、Date、Obeject、function、RegExp(正则表达式)

（3）、空数据类型

* Null ---→Null 的数据类型会返回一个 Object

* undifined

tips：用 typeof 可以进行判断数据类型；

tips：定义的简单数据类型变量，其数据保存在变量中；

而复杂数据类型，其变量保存的是数据所在的内存地址；

15、内置对象

Array、Date、Math、String；

16、（Math）数学对象

向上取整　　Math.ceil(number);

向下取整　　Math.floor(number);

四舍五入　　Math.round(number);

求多个数字之间的最大值　　Math.max();

求多个数字之间的最小值　　Math.min();

求 x 的 y 次幂　　Math.pow(x,y);

求正弦值　　Math.sin(x);

example:

求一个角度的正弦值，要求 x 必须是一个额弧度值

角度和弧度的转换公式：

```
弧度 = 角度 * 2 * Math.PI / 360;
```

```
Math.sin(30*2*Math.PI/360)
```

```
Math.abs(x);
```
得到一个数字的绝对值

17、（Array）数组对象

（1）、arr1.concat(arr2);

数组拼接，结果为将 arr2 拼接到 arr1 的最后；

（2）、arr.join();

数组字符串输出，括号内可以指定元素连接的符号；

eg:

```
arr=["a","b","c","d"];
console.log(arr.join("|"));
```
(结果为"a|b|c|d")

（3）、arr.pop();

切除数组的最后一个元素，返回值为该元素；

（4）、arr.slice(start,end)

获取，获取数组的指定片段，start 必须有，如果参数为负数则从末尾开始选取；

返回值为该片段组成的，一个新的数组；

（5）、arr.push

添加，用于向数组的末尾添加新的元素，参数可以是多个；

返回值为数组的新长度；

（6）、arr.splice

1、用于向数组中指定的索引添加元素；

```
arr.splice(2, 0, "William","asdfasdf");
```

在第 2 个元素开始，删除的元素个数（可以为 0，为 0 到结尾），

加入元素为"William"、"asdfasdf"；

2、用于替换数组中的元素；

```
arr.splice(2,1,"William");
```

3、用于删除数组中的元素；

```
arr.splice(2,2);
```

（7）、arr.indexOf(element);

查找，在数组中查找 element，返回值为索引，如果没有该元素返回-1；

（8）、arr.sort(function);

排序，function 为一个函数；

eg:

```
function sortNumber(a,b){
```

```
        return a-b;
    }
    arr.sort(sortNumber);(从小到大排序)
```

tips：如果 a-b 改成 b-a，那么执行的操作为从大到小；

tips:字符串对象（String）的方法与 Array 的方法类似；

18、（Date）日期对象

date.getTime()date.getMilliseconds()date.getSeconds()date.getMinutes()date.getHours()date.getDay()date.getDate()date.getMonth()date.getFullYear()

tips:很多，查文档

19、（String）对象

charAt(index)

str[index]　　获取字符串指定位置的字符

concat()　　拼接字符串----------

slice(start,end)/substring(start,end)　　截取从 start 开始，end 结束的字符，返回一个新的字符串，若 start 为负数，那么从最后一个字符开始；

substr(start,length)　　截取从 start 开始，length 长度的字符，得到一个新的的字符串----------

indexOf(char)　　获取指定字符第一次在字符串中的位置

lastIndexOf(char)　获取指定字符最后一次出现在字符串中的位置

trim()　　去除字符串前后的空白----------

toUpperCase()

toLocaleUpperCase()　　转换为大写

toLowerCase()

toLocaleLowerCawse()　　转换为小写----------

replace()　　替换字符 split()　　分割字符串为数组

20、自定义对象

对象：无序属性的集合；

特征：属性（key）；

行为：方法（value）；

js 是基于对象的弱类型语言；

继承：基于类，子类可以从父类得到的特征；

工厂模式：定义一个 function 构造函数，作为对象，要创建对象直接调用该构造函数，加 new 关键字；

构造函数：定义对象的函数，里面存有该对象拥有的基本属性和方法；

命名首字母大写，this 会自动指代当前对象；

访问对象属性：

```
obj[key];

obj.key;
```

遍历对象：

```
for(key in obj){
    key         为属性名；
    obj[key]    为属性值（value）；
}
```

21、JSON

```
{
  "name" : "李狗蛋",
  "age" : 18,
  "color" : "yellow"
}
```

1、 所有的属性名，必须使用双引号包起来；

2、 字面量侧重的描述对象，JSON 侧重于数据传输；

3、 JSON 不支持 undefined；

4、 JSON 不是对象，从服务器发来的 json 一般是字符串，

通过 JSON.parse(jsonDate.json)可以将其转换成 js 对象；

22、JS 解析

（1）、作用域

全局作用域：整个代码所有地方都可以调用；

局部作用域：在函数内部声明的变量，只可以在函数内部使用；

（2）、变量提升和函数提升

预解析：在解析的时候，var 和 function 都会被提升到代码的最顶端；

但是赋值操作不会被提升，定义和函数才会被提升；

if 里面的变量定义也会被提升，但是赋值操作不会；

23、其他细节（tips）

(1)、元素由对象组成的数组进行排序

eg：

```
var data = [
    {title: "老司机", count: 20},
    {title: "诗人", count: 5},
    {title: "歌手", count: 10},
    {title: "隔壁老王", count: 30},
    {title: "水手", count: 7},
    {title: "葫芦娃", count: 6},
];
    //该数组的元素都为对象。我们需求为根据 count 的值给数组重新排序。
    //解决方案：使用 sort 方法，对传入的函数做手脚。
```

```
function sortArr(a,b){
    return a.count > b.count;
}
data.sort(sortArr);

    //原本的 a 和 b 的比较方法变成 a.count 和 b.count;
    //原本的比较方法可以参见 17，数组对象
    //至此，data 将以 count 值从小到大排列。
```

tips:Array 对象的 sort 方法传入的为比较函数，比较函数里 return 排序比较的方法；

原始的 sort 方法传入的函数内部 return 的值为 a>b，

通过修改 sort 传入的函数，可以实现对元素为对象的数组的排序！