Javascript

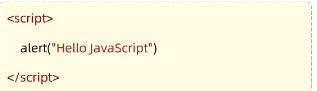
Javascript和java是一门跨平台、面向对象的脚本语言。用来控制网页行为的，它能使网页可交互。Javascript和java语言是完全不一样的，不论是概念还是设计，但是基础语法类似

Javascript引入方式

内部脚本：将JS代码定义在HTML页面中

* JavaScript代码必须位于<script><script>
* 在HTML文档中，可以在任意地方，放置任意数量的<script>
* 一般会把脚本置于<body>元素的底部，可以改善显示速度

语法形式如下图：



外部脚本：将js代码定义到外部js文件中，然后引入到HTML页面中

* 外部js文件中，只包含js代码，不包含<script>标签
* <script>标签不能自闭合

内部文本调用js语句：



在js文档里面的编写仍然是要按内部脚本的方式编写。

Javascript基础语法

书写语法：

* 区分大小写;与java一样，变量名、函数名以及其他一切东西都是区分大小写的
* 每行结尾的分号可有可无
* 注释：

单行注释：//注释内容

多行注释：/\*内容\*/

* 大括号表示的是代码块



输出语句：

* 使用window.alert()写入警告框
* 使用document.write（）写入HTML输出
* 使用console.log()写入浏览器控制台

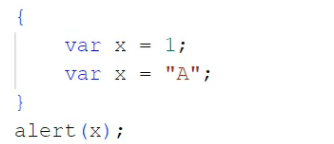


变量：

* Javascript中用var关键字来声明变量。
* Jacascript是一本弱语言，变量可以存放不同类型的值
* 变量名需要遵循如下规则：
* 组成字符可以是任何字母、数字、下划线（\_）或美元符号
* 数字不能开头
* 建议使用驼峰命名

特点：1.作用域大，为全局变量

2.可以重复定义的



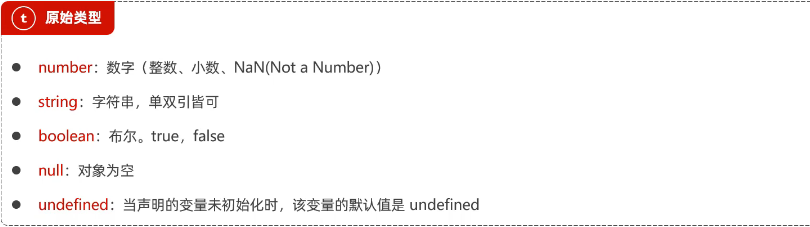
* JavaScript中用let关键字来定义变量。它的用法类似与var但是他在代码块内定义，只能在代码块中生效，并且不允许重新声明
* JavaScript中用const关键字，用来声明一个只读的常量。一旦声明，常量的值就不能改变

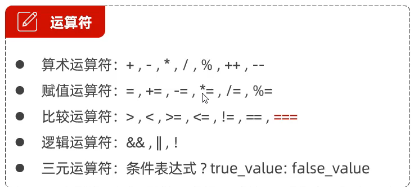
数据类型、运算符、流程控制语句：

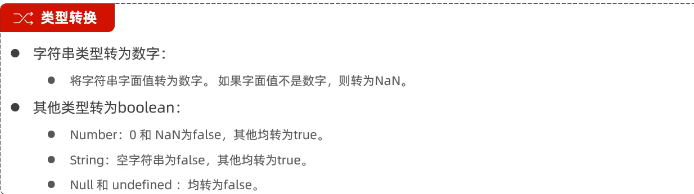
数据类型：

JavaScript中分为：原始类型和引用类型

Typeof用于获取数据类型









函数：

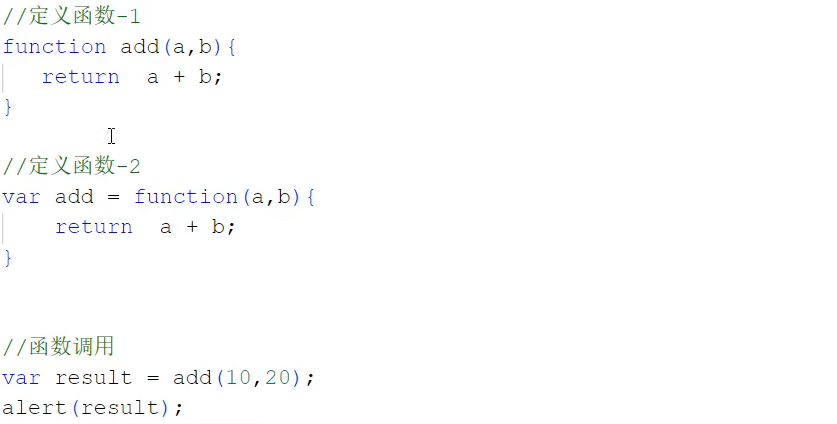
* 函数是被设计为执行特定任务的代码块
* 定义：JavaScript函数通过function关键字进行定义，语法：

Function functionName（参数1，参数2....）{

//要执行的代码

}

* 注意：形式参数不需要类型。因为JavaScript是弱类型语言
* 返回值也不需要定义类型，可以在函数内部直接使用return返回即可
* 调用：函数名称（实际参数列表）



JS对象：

Array：

* JavaScript中Array对象用于定义数组。
* 定义：



* 访问：

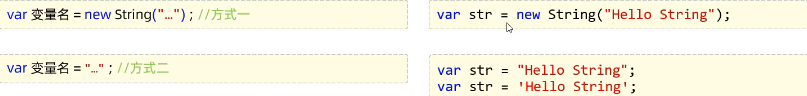


特点： 长度可变，类型可变



String：

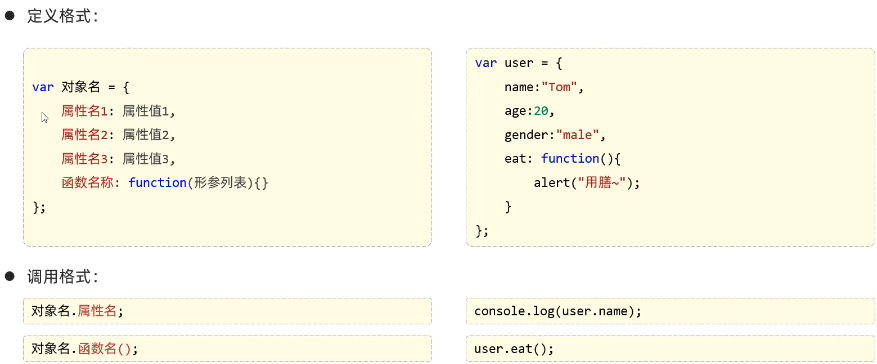
* String字符串对象创建方法有两种：





JavaScript自定义对象：

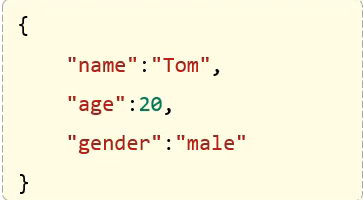
定义格式如下：



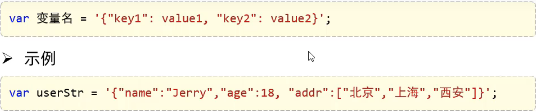
JS-对象-JSON-介绍

JSON是通过JavaScript对象标记法书写的文本

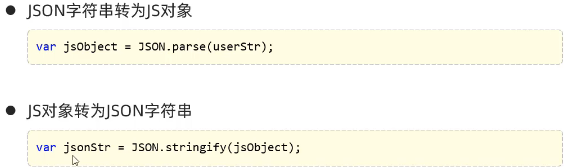
多用于数据的载体，在网络中进行数据传输



语法定义：







JS-对象-BOM-介绍

Window

介绍：浏览器窗口对象。

获取：直接使用window，其中window可以省略

属性：history：对History对象的只读引用。

Location：用于窗口或框架的Location对象。

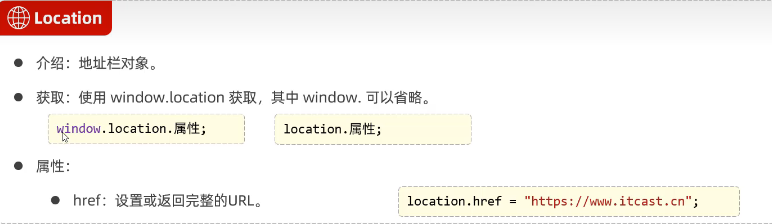
Navigator：对Navigator对象的只读引用。

方法：alert（）：显示带有一段信息和一个确认按钮的警告框

Confirm（）：显示带有一段消息以及确认按钮和取消按钮的对话框

Setlnterval（）：按照指定的周期（以毫秒计）来调用函数或计算表达式

SetTimeout（）：在指定的毫秒数后调用函数或计算表达式



JS-对象-DOM-介绍

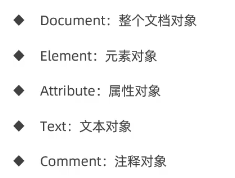
将标记语言的各个组成部分封装为对应的对象：

JavaScript通过DOM，就能够对HTML进行操作：

* 改变HTML元素的内容
* 改变HTML元素的样式
* 对HTML DOM事件作出反应
* 添加和删除HTML元素

DOM是W3C的标准，定义了访问HTML和XML文档的标准，分为3个不同的部分：

1. Core DOM-所有文档类型的标准模型



1. XML DOM-XML文档的标准模型
2. HTML DOM-HTML文档的标准模型



HTML中的Elemnet对象可以通过Document对象获取，而Document对象是通过window对象获取的。

Document对象中提供了以下获取Element元素对象的函数：

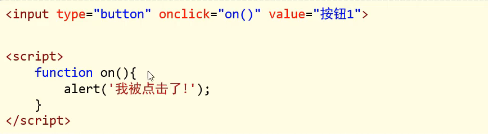


JS事件监听

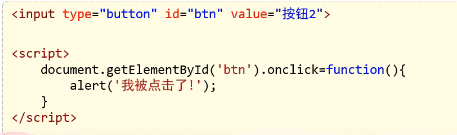
JavaScript可以在事件被侦测到时执行代码

事件绑定

方法一：通过HTML标签中的事件属性进行绑定



方法二：通过DOM元素属性绑定



常见事件：

