JavaScript

千锋教育
QIAN FENG EDUCATION

做真实的自己，用良心做教育

张骏
H5教学部

函 数(上)

课程概要：

函数的概念以及作用
函数的创建方式
函数的好处
参数的声明
参数的传递
arguments对象
返回值的设置和接收
JS的编译和执行
递归

函数的概念以及作用

- 函数(function)是一个大型程序中的某部分代码（子程序），由一个或多个语句块组成。它负责完成某项特定任务，而且相较于其他代码，具备相对的独立性。
- 一般会有输入参数并有返回值，提供对过程的封装和细节的隐藏。这些代码通常被集成为软件库。
- 在面向对象编程语言中，类别或对象的子程序也被称作方法(method)

函数的创建方式

- 函数有两种常用的创建方式

1.

```
function 函数名(){
        //声明式函数，使用function声明一个函数
}
```

2.

```
var fn = function(){
        //赋值式，将一个函数赋值给一个变量，也称作函数表达式
}
```

函数的好处

- 1. 使用函数可以程序过程进行封装、隐藏细节
- 2. 函数有很好的复用性
- 3. 函数可以被事件触发执行
- 4. 可以隔离作用域，避免作用域污染

参数的声明

- JavaScript中的函数可以带参数也可以不带参数
- 参数的声明写在小括号中，多个参数使用逗号隔开
- 小括号内的参数叫做形(式)参(数),命名规范与变量命名相同
- 函数体内使用的参数叫做实(际)参(数)

参数的传递

- 调用函数时，可在小括号内传入参数
- JavaScript函数的参数是按值传递的
- 例如：

```
var num1 = 10;
var num2 = 8;
function fn(a,b){
        a+=b;
        console.log(a); //18
}
fn(num1,num2);
console.log(num1,num2); //10 8
```

上面代码调用fn函数时传入的num1,num2是传入的这两个变量的值(将值复制一份传入函数内)，函数内对参数的修改不会影响外部变量的值

arguments对象

- 每个函数对象都有一个arguments属性；此属性只能在函数执行体内使用。
- arguments属性中保存这当前函数接受到的所有实际参数，故可以使用arguments属性处理可变数量的参数。

- arguments对象具有如下属性：
  - length：返回实际传入的参数的个数
  - callee：返回当前函数的引用

返回值的设置和接收

- 函数可以设置返回值也可以没有返回值
- 当函数没有返回值时默认返回undefined
- 使用return语句为函数设置返回值
- 函数的返回值可以是任意数据类型，且只能有一个返回值
- 当return执行时，跳出函数体的执行
- return语句也常被用来终止或跳出函数执行
- 调用有返回值的函数时，可以使用变量接收返回值

JS的编译和执行

JS的编译和执行分为两个阶段

1. 编译阶段

对于常见编译型语言（例如：Java）来说，编译步骤分为：词法分析->语法分析->语义检查->代码优化和字节生成。

对于解释型语言（例如JavaScript）来说，通过词法分析和语法分析得到语法树后，就可以开始解释执行了。

（1）词法分析是将字符流(char stream)转换为记号流(token stream)，就像英文句子一个个单词独立翻译，举例：

代码：var result = testNum1 - testNum2;

词法分析后 ：

NAME "result"

EQUALS

NAME "testNum1"

MINUS

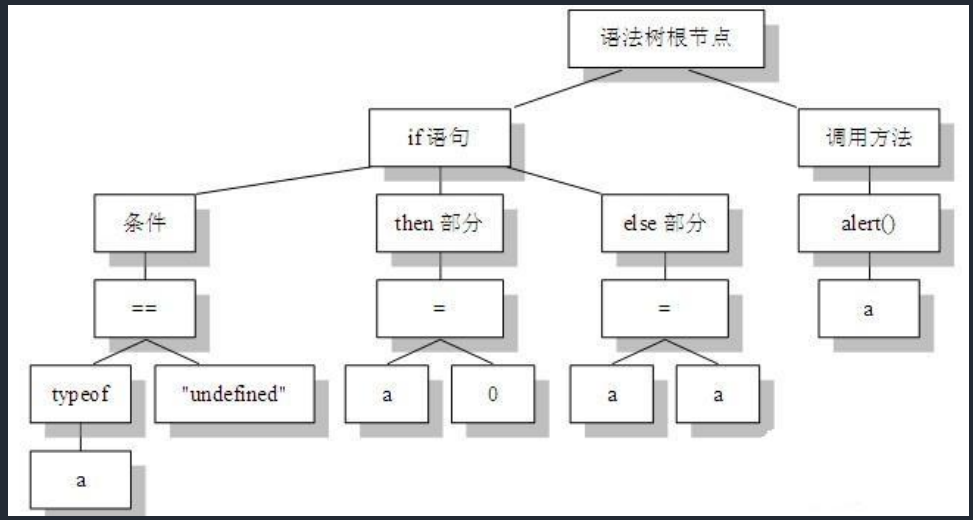
NAME "testNum2"

SEMICOLON

（2）语法分析得到语法树，举例：

条件语句 if(typeof a == "undefined"){ a = 0; } else { a = a; } alert(a);

当JavaScript解释器在构造语法树的时候，如果发现无法构造，就会报语法错误，并结束整个代码块的解析。



JS的编译和执行

（3） “预编译” （并非完全的顺序执行）

“function函数” 是一等公民！编译阶段，会把定义式的函数优先执行，也会把所有var变量创建，默认值为undefined，以提高程序的执行效率！

总结：当JavaScript引擎解析脚本时，它会在预编译期对所有声明的变量和函数进行处理！并且是先预声明变量，再预定义函数！

二、JavaScript执行过程

在解释过程中，JavaScript引擎是严格按着作用域机制（scope）来执行的。JavaScript语法采用的是词法作用域（lexcical scope），也就是说JavaScript的变量和函数作用域是在定义时决定的，而不是执行时决定的，由于词法作用域取决于源代码结构，所以 JavaScript解释器只需要通过静态分析就能确定每个变量、函数的作用域，这种作用域也称为静态作用域（static scope）。补充：但需要注意，with和eval的语义无法仅通过静态技术实现，实际上，只能说JS的作用域机制非常接近lexical scope。

JavaScript中的变量作用域在函数体内有效，无块作用域；

JavaScript引擎在执行每个函数实例时，都会创建一个执行环境（execution context）。执行环境中包含一个调用对象（call object），调用对象是一个scriptObject结构（ “运行期上下文”），用来保存内部变量表varDecls、内嵌函数表funDecls、父级引用列表upvalue等语法分析 结构（注意：varDecls和funDecls等信息是在语法分析阶段就已经得到，并保存在语法树中。函数实例执行时，会将这些信息从语法树复制到 scriptObject上）。scriptObject是与函数相关的一套静态系统，与函数实例的生命周期保持一致，函数执行完毕，该对象销毁。

JavaScript引擎通过作用域链（scope chain）把多个嵌套的作用域串连在一起，并借助这个链条帮助JavaScript解释器检索变量的值。这个作用域链相当于一个索引表，并通过编号来存 储它们的嵌套关系。当JavaScript解释器检索变量的值，会按着这个索引编号进行快速查找，直到找到全局对象（global object）为止，如果没有找到值，则传递一个特殊的undefined值。

递归

- 递归，就是在运行的过程中调用自己。
- 构成递归需具备的条件：
- 1. 子问题须与原始问题为同样的事，且更为简单；
- 2. 不能无限制地调用本身，须有个出口，化简为非递归状况处理。
- 在数学和计算机科学中，递归指由一种（或多种）简单的基本情况定义的一类对象或方法，并规定其他所有情况都能被还原为其基本情况。

THANK YOU

千锋教育
QIANFENG EDUCATION

做真实的自己，用良心做教育