

千锋HTML5学院

第二阶段javascript课程课件



1

函数的概念，及作用

2

函数的定义

3

如何执行函数(调用)

4

事件驱动的概念

5

函数的参数(实参，形参)

6

作用域

函数的概念和作用

□ 函数的概念

函数就是把完成特定功能的一段代码抽象出来，使之成为程序中的一个独立实体，起个名字（函数名）。可以在同一个程序或其他程序中多次重复使用（通过函数名调用）。

□ 函数的作用

- 使程序变得更简短而清晰
- 有利于程序维护
- 可以提高程序开发的效率
- 提高了代码的重用性(复用性)

不使用函数，**JS**代码必须执行，且执行时间，执行哪部分代码都是不可控的。
使用函数后，浏览器不会直接执行它，只有在调用这个函数时，它才会被浏览器所执行。

函数的分类

□ 内置函数（系统函数，官方函数）

是官方提供好的函数，直接使用，如：alert()。

■eval()：用于计算字符串表达式的值

■isNaN()：用于验证参数是不是数字，是not a number的意思。

■setInterval(“函数名()”,1000);

//按照指定的周期（以毫秒计）来调用函数或计算表达式。
此处的1000为毫秒，即是1秒。

□ 自定义函数（用户自定义的函数）

用户根据实际需求，需要自己封装一个函数。

函数的定义

□function关键字定义一个函数

```
function 函数名(参数1名字, 参数2名字, ……) //函数可以没有形参
{
    语句
    [return 返回值;] //函数可以没有返回值
}
```

□示例:

```
function sum(one, two) {
    var s = one+two;
    return s;
}
```

函数名

参数

返回值

参数是已知条件，
返回值是函数的执行结果

函数的调用

□通过在程序中使用函数名称，可以执行函数中包含的语句，这称为调用函数，同时传入实参。

函数名(参数1值, 参数2值, ……);

□示例:

`alert(“亲”);`这句话是调用内置函数

`sum(2, 8);`这句话是在调用上页ppt中的自定义函数，计算2+8的结果。

注意:

整个程序的开始执行，是由事件驱动的

示例

- 判断一个年份是不是闰年
- 判断一个数是不是素数
- 年月日分别为自定义函数的参数，判断是否为正确的日期
- 验证一下文本框里的内容是否为数字（结合HTML元素的）
- 点击不同的数字按钮，显示不同的图片（类似于图片轮播中去掉自动播放）

把函数写在单独的JS的文件中，
可以体现函数的优势

事件驱动

□事件驱动:

所有的函数，没有调用不会执行，那么函数调用的源头在何处，就是事件，我们已经学习了 `onclick`，还有 `onload`、`onfocus`、`onblur` 等等。

`onload`: 页面加载完成 (页面完全打开)

`onfocus`: 获得焦点 (简单理解为，光标进入)

`onblur`: 失去焦点 (简单理解为，光标离开)

示例：自动清空文本框的内容。

形参和实参

□形参和实参：

函数的参数跟变量是一样使用。

形参就是在函数定义时，函数名后面的参数，不能用**var**修饰。

实参就是调用时，函数名后面的参数

□参数可以传递多个，与函数定义时的参数个数无关

如：

```
function test(paramX,paramY){  
  
}  
test("a","b","c");
```

使用**arguments**对象可以判断参数的个数，**arguments**是个数组。

变量的作用域

□作用域：就是起作用的范围。或者说有效范围。

□局部变量

局部变量就是定义在函数内部的变量，这个变量只能在函数内部使用，即作用域范围只是函数内部，另外，形参也是局部变量。

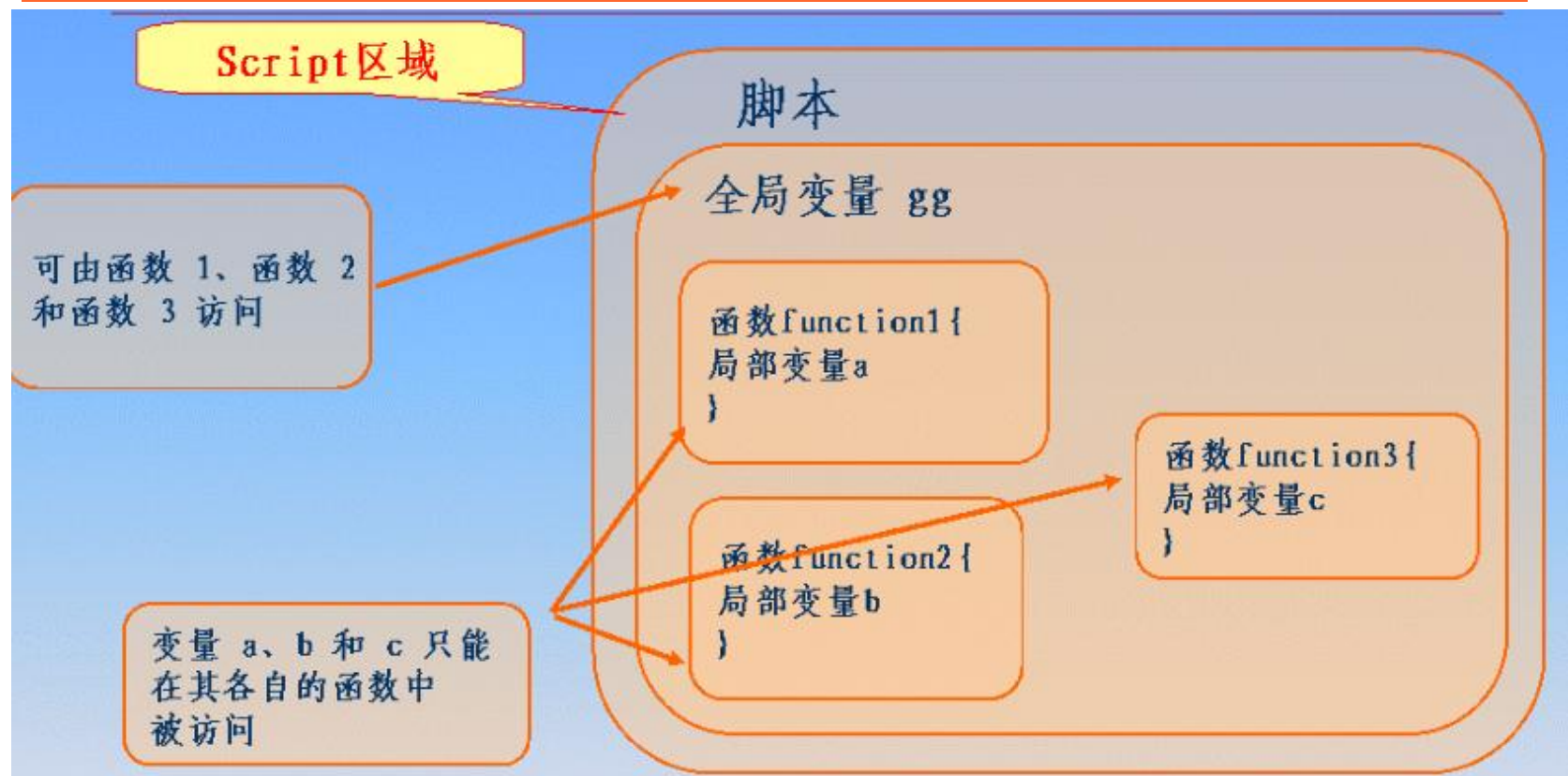
注意：

局部变量和全局变量最好不要重名

□全局变量

全局变量就是定义在函数外部的变量，这个变量在任何函数中都有效，即作用域范围是当前文件的任何地方。

变量的作用域



□注意：在函数内部定义变量时，没有使用var，也是属于全局变量。

函数的嵌套调用

□概念：

函数之间允许相互调用，也允许嵌套调用

函数的递归调用

□概念：

函数还可以自己调用自己，称为递归调用

□重要性：

递归的分量，递归属于函数中比较难理解的知识，在应用开发中，虽然使用不是很频繁，但是很体现你的功底，而且，从事IT行业开发，最好要会递归，如果说现在可以不要求灵活运用的话，以后到公司中一定要会，如果面试中有人问你递归，说明，他对你要求挺高。

函数的递归调用

□ 示例 (阶乘) :

5 的 阶乘 为 : 120

分析:

$$\begin{aligned} 5! &= 5 * 4 * 3 * 2 * 1 \\ 4! &= 4 * 3 * 2 * 1 \\ 3! &= 3 * 2 * 1 \\ 2! &= 2 * 1 \\ 1! &= 1 \\ 0! &= 1 \end{aligned}$$

$$n! = n * (n-1) ! ;$$

函数的递归调用

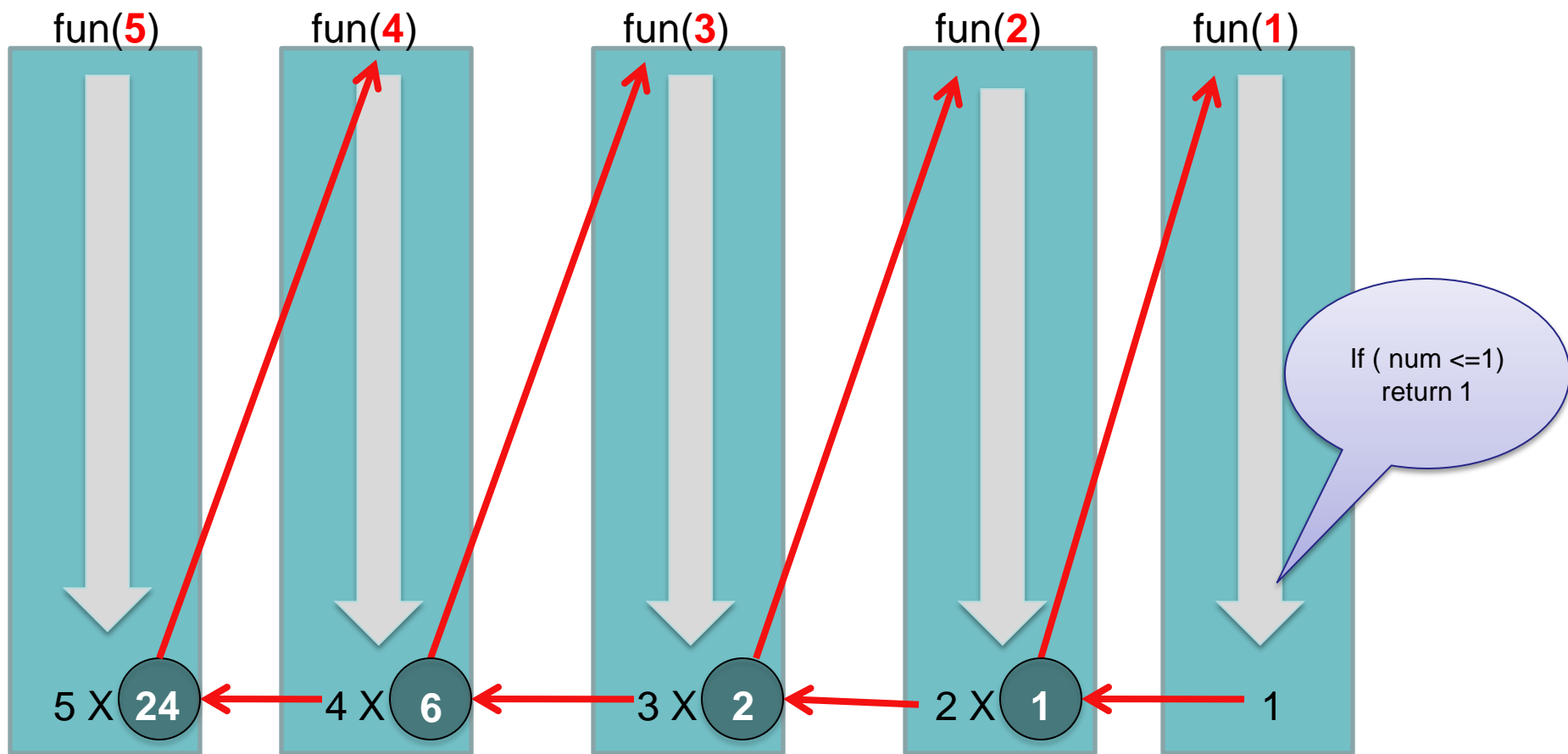
□ 强调：

递归函数的函数体内一定需要有一个分支是不调用自身的，否则，会无休止地调用下去，引起栈溢出。

函数的递归调用

```
function fun(num){  
    if(num<=1){  
        return 1;  
    }else{  
        return num*fun(num-1);  
    }  
}
```

□ 函数阶乘代码执行过程



With的用法

□with:

可以省略对象名前缀，即点前面的代码，以后会讲对象

```
with(document){  
    write("abc");  
    write("def");  
    getElementById("div_bar");  
}
```

作业

必做：

1. 把今天的课堂案例敲一遍
2. 年月日分别为自定义函数的参数，判断是否为正确的日期
3. 验证一下文本框里的内容是否为数字（结合HTML元素的）

可选：

- 1、可以尝试做检验文本框内容的长度
提示：**length**是获得文本框值的长度
document.getElementById("id的值").value.length
2. 点击不同的数字，显示不同的图片（类似于图片轮播中去掉自动播放）（结合HTML元素的）
3. 输出斐波那契额数列的前20项(递归)
1 1 2 3 5 8 13 21

