# JavaScript运行机制

代码：

let a=20;

const b=30;

var c;

function multiply(e,f){

    var g=20;

    return e\*f\*g;

}

c=multiply(20,30);

console.log(c)

可整理成：

function multiply(e,f){

    var g=20;

    return e\*f\*g;

}

var c;

//只执行这三条

let a=20;

const b=30;

c=multiply(20,30);

1、当代码运行时，JavaScript 引擎首先创建一个全局执行上下文来执行全局代码。

a：未初始化

b：未初始化

C：undefined

multiply:函数

全局执行上下文

创建

全局环境

环境记录

Outer:null

This:全局对象

2、执行阶段，变量赋值完成。

执行 全局环境

a：20

b：30

C：undefined

Multiply:函数

环境记录

Outer：null

This:全局对象

全局执行上下文

3、当遇到调用函数 multiply(20, 30) 时，就创建一个新的函数执行上下文来执行该函数代码。

执行全局上下文

函数执行上下文

创建 全局环境

环境记录

Outer：全局词法环境

This:全局对象或undefined

g：undefined

Argument{0:20,1:30,length:2}

4、该执行上下文进入执行阶段，也就是说，该函数内的变量赋值已经完成。

函数执行上下文

执行全局上下文

执行 全局环境

g：20

Argument{0:20,1:30,length:2}

环境记录

Outer：全局词法环境

This:全局对象或undefined

1. 该函数执行完成后，返回值被存储在 c 中。程序结束

