

如果现在我们的编译器里面写了这些代码，在编译阶段会怎么样呢

计算机的内存是分为栈结构和堆结构的
V8引擎在执行代码的时候会先创建一个执行上下文栈（execution context stack），也叫函数调用栈。以后代码想要运行，都要在这个栈里面才能运行
因为我们可能要执行全局代码，所以我们会创建一个东西叫做全局执行上下文（global execution context）全局代码执行的时候才会创建全局执行上下文

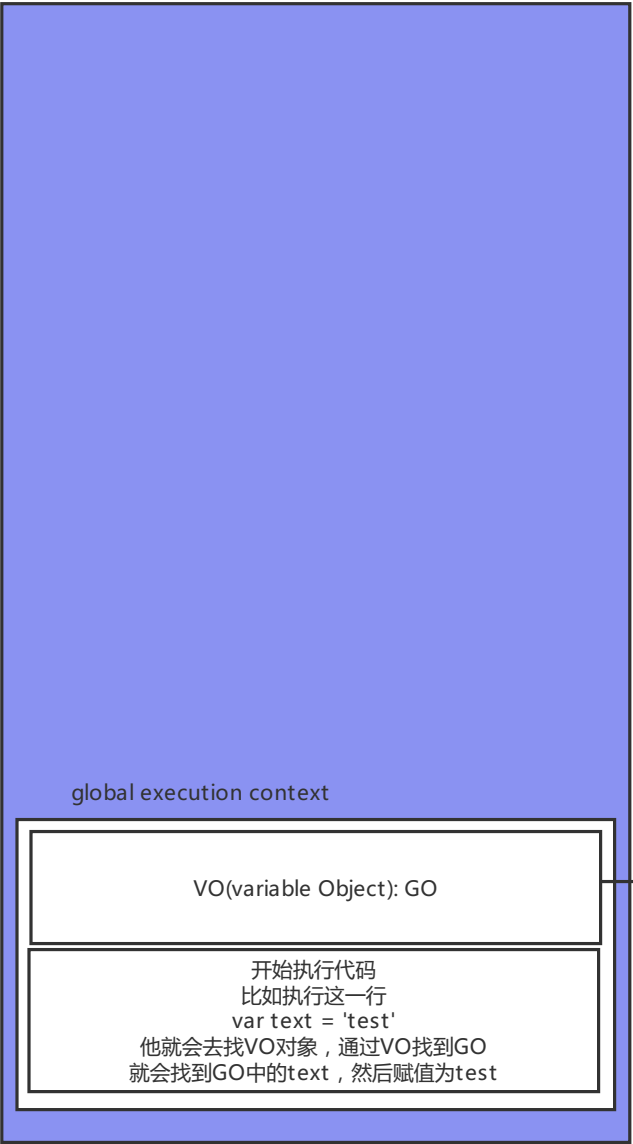
在这个过程中会生成一个对象叫做GlobalObject
这个对象里面有很多东西，比如 globalObject = {
String : '类', Date : '类', console : '类', setTimeout:'函数', setInterval:'函数', window: globalObject
}
其中window指向的就是GlobalObject

var text= 'test';
var num1 = 1;
var num2 = 2;

在这个过程中会生成一个对象叫做GlobalObject
这个对象里面有很多东西，比如 globalObject = {
String : '类',
Date : '类',
console : '类',
setTimeout:'函数',
setInterval:'函数',
window: globalObject 指向的就是globalObject这个对象
text: undefined,
num1: undefined,
num2: undefined
}
其中window指向的就是GlobalObject

在解析的时候会解析我们的代码，如果有上面定义的变量，就会放到globalObject这个对象中并且赋值为undefined

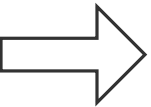
ECStack(execution context stack)



vo对应的是GO

globalObject = {
String : '类',
Date : '类',
console : '类',
setTimeout:'函数',
setInterval:'函数',
window: globalObject 指向的就是
globalObject这个对象
text: undefined,
num1: undefined,
num2: undefined
}

当上面的所有准备完成以后，开始执行代码，挨个执行



var text= 'test';
var num1 = 1;
var num2 = 2;