

神语言

The God Language

[1. 1+2=3](#)

[2. 函数](#)

[3. table](#)

[4. 宏](#)

[5. 元编程](#)

[6. 闭包](#)

[7. 协程](#)

[8. 类型转换](#)

[9. 常量](#)

[10. 控制结构](#)

1. 1+2=3

第一个 god 程序：

```
a=1  
b=2  
print(a+b)
```

god没有main函数，语句后面的分号是可选的。

2. 函数

举例说明：

```
var f=function(a){  
    return a*2  
}  
  
print(f(3))
```

带 var 的是局部变量，不带 var 的是全局变量。

可变参数：

```
b=function(...){  
    arg=[...]  
    print(#arg)  
}  
  
b(9)  
b(9,8)
```

#表示取 table 的长度。

多返回值：

```
c=function(a){  
    return a*2,a*3  
}  
  
print(c(4))
```

3. table

god 的 table 使用方括号而不是花括号，下标从 1 开始：

```
var a=[7,8]
print(a[1])
print(a[2])
```

table 索引用法：

```
var b=['abc','123',test=111]
print(b[1])
print(b.test)
print(b['test'])
```

4. 宏

普通宏：

```
define var local
define elseif else if
define && and
define || or
define ! not
```

超级宏：

```
define$ for ( _mword ) {
{
    for $0 do
}
```

带参数的宏和变参宏可用超级宏实现。

5. 元编程

和 lua 一样：

```
assert(loadstring('print(11)'))()
```

6. 闭包

和 lua 一样：

```
var a=2
(function(){
  print(a)
})()
```


7. 协程

和 lua 一样。

8. 类型转换

和 lua 一样。

9. 常量

和 lua 一样。

10. 控制结构

控制结构的圆括号和花括号均不可省略：

```
if(i==1)
{
    print(i)
}
elif(i!=2)
{
    print(i)
}
else
{
    print(i)
}
```

elif 和 else if 是等价的。

god 没有 switch，只能用 if...elif...代替。

while 循环：

```
while(i<10)
{
    print(i)
    i++
}
```

注意++运算符有限制。

for 循环和 lua 一样：

```
for(i=1,10)
{
    print(i)
}
```

god 没有 continue，只能用 goto 代替。

至于 break 和大多数语言一样，不再赘述。