X 语言程序设计



本节内容

作用域

内存布局

动态内存分配

变量的作用域

X 语言的变量作用域分为：

代码块作用域（代码块指 {} 之间的一段代码）

函数作用域

文件作用域

局部变量

局部变量也叫 auto 自动变量（auto 可写可不写），一般情况下代码块 {} 内部定义的变量都是自动变量，它有如下特点：

在一个函数内定义，只在函数范围内有效

在复合语句中定义，只在复合语句中有效

随着函数调用的结束或复合语句的结束，局部变量生命周期也结束

如果没有赋初值，内容随机

静态（στατιχ）局部变量

στατιχ 局部变量作用域在代码块或定义的函数内有效

στατιχ 局部变量的生命周期和程序运行周期一样

στατιχ 局部变量的值只初始化一次

如果没有赋初值，会自动赋值，数值型为 0 ，字符型为空字符

全局变量

在函数外定义，可被本文件和其它文件中的函数所共用

全局变量的生命周期和程序运行周期一样

不同文件的全局变量不可重名

静态全局变量

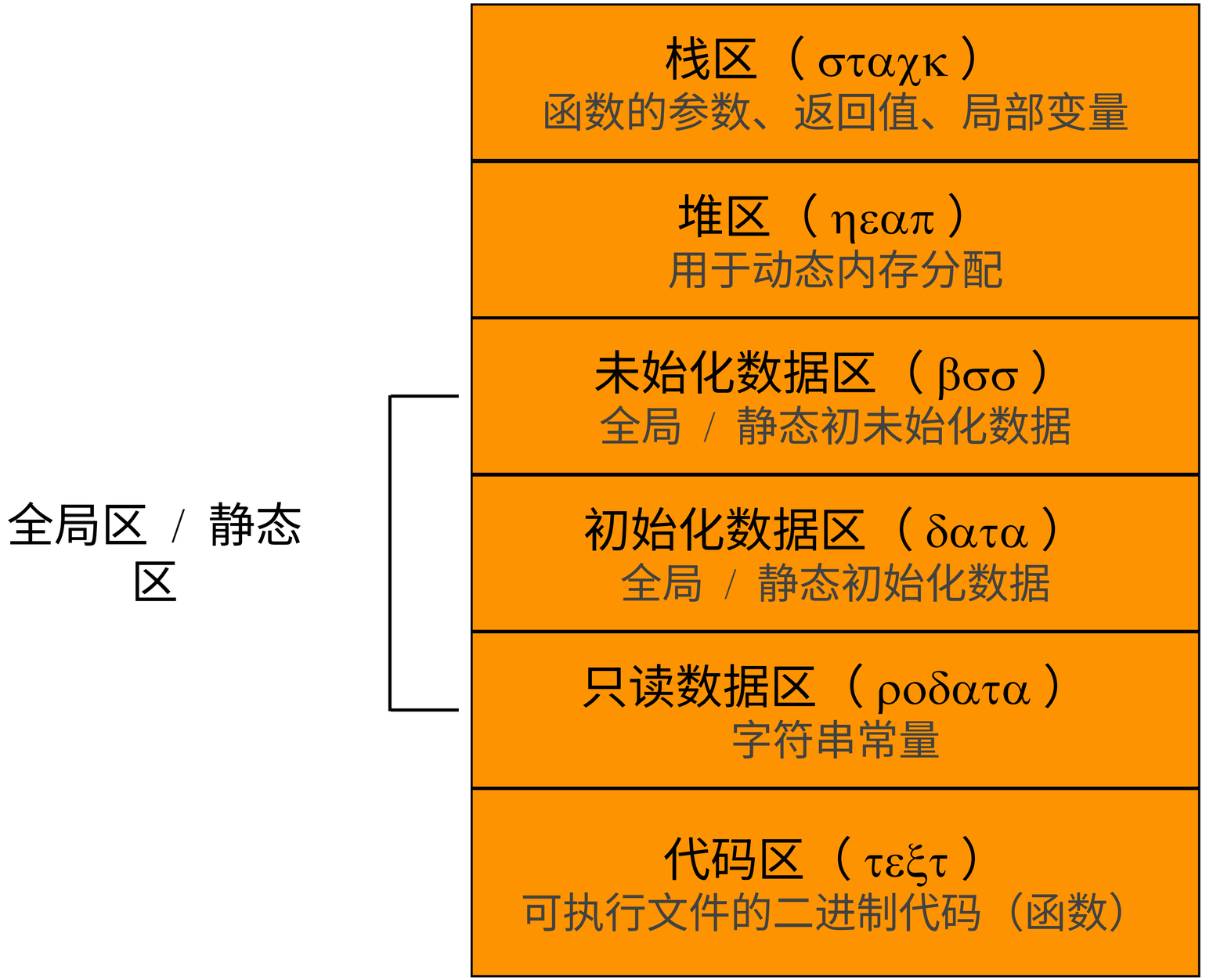
在函数外定义，作用范围被限定在所定义的文件中

不同文件静态全局变量可以重名，作用域不冲突

生命周期与程序运行周期一样

只初始化一次

内存布局


栈区（σταχκ）
函数的参数、返回值、局部变量
堆区（ηεαπ）
用于动态内存分配
未始化数据区（βσσ）
全局 / 静态初未始化数据
初始化数据区（δατα）
全局 / 静态初始化数据
只读数据区（ροδατα）
字符串常量
代码区（τεξτ）
可执行文件的二进制代码（函数）
全局区 / 静态区

堆内存分配与释放

在堆内存中分配一块长度为 σιζε 字节的连续区域，用来存放指定类型的数据

```
#include<stdlib.h>
void* malloc(size_t size)
```

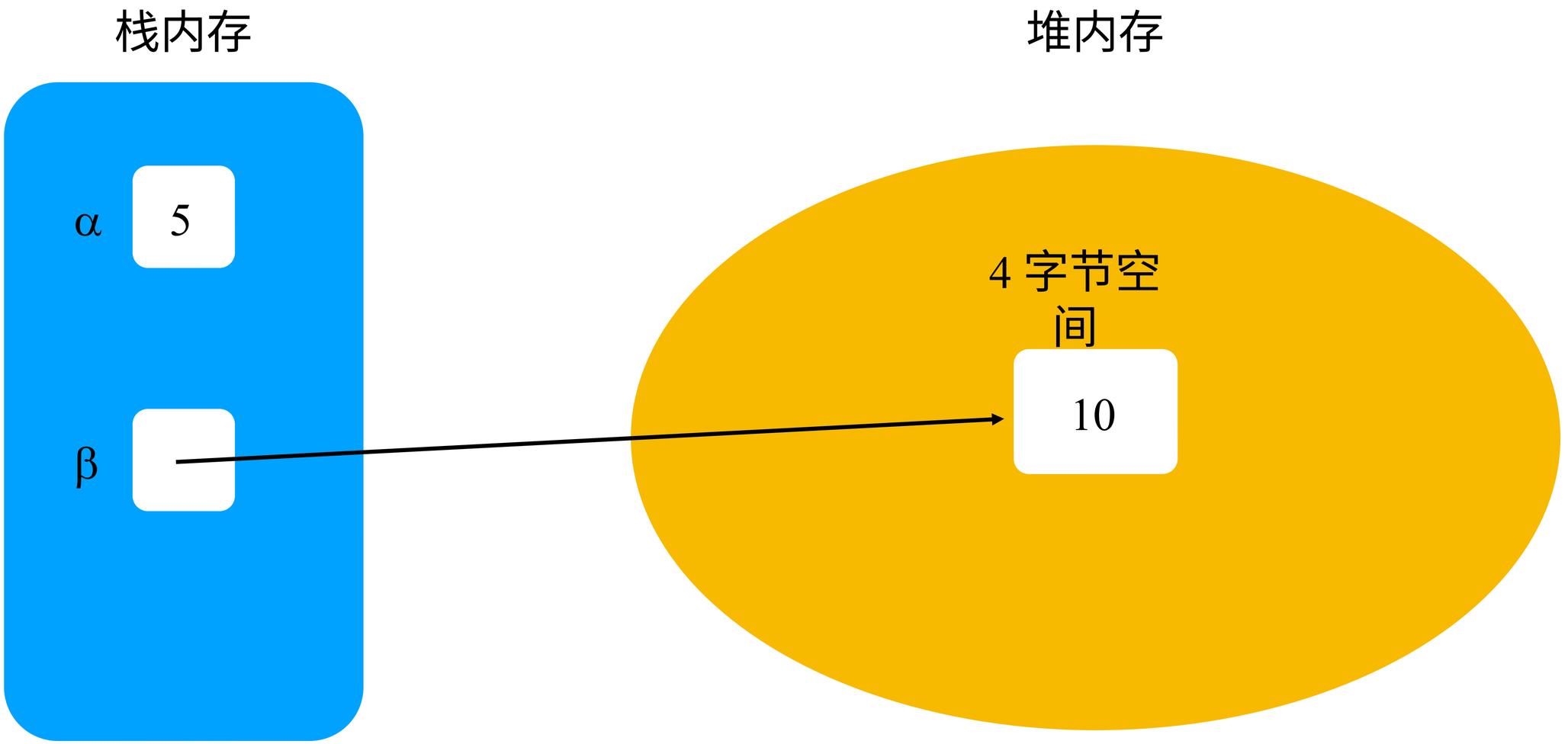
μεμορψ
αλλοχατε

返回值：ωοιδ*：万能指针，可以被转换成任意类型的指针

成功：分配空间的起始地址

失败：
NYΛΛ

```
//在栈内存中声明一个int类型的变量，占4个字节
int a = 5;
//在堆内存中开辟一片4个字节的空间，栈内存中的指针变量b保存堆内存空间的首地址
int* b = (int*)malloc(sizeof(int));
```



堆内存分配与释放

释放 πτρ 所指向的一块内存空间，对同一内存空间多次释放会出错。

```
#include<stdlib.h>

void free(void* ptr)
```

内存泄漏

内存只申请不释放，导致程序使用的内在空间持续增长

