嵌入式C语言之-

函数的语法格式

讲师：叶大鹏

应用案例

- 分别打印输出1~100、201~300、401~500的累加和：

整体介绍

- **函数的语法格式：**

返回值类型　函数名(数据类型 参数1, 数据类型 参数2, ...)

{

　　函数体

}

```
int32_t Sum(int32_t begin, int32_t end)
{
        int32_t sum = 0;
        int32_t i;

        for (i = begin; i <= end; i++)
        {
                sum += i;
        }
        return sum;
}
```

函数的两种存在形式

- **库函数：**

```
int main(void)
{
        int32_t res = 0;

        res = Sum(1, 100);
        printf("sum of 1~100 is %d.\n", res);
        return 0;
}
```

- printf是C语言的标准库函数，除了标准库以外，还有很多三方开源库，比如cjson、单片机厂商提供的HAL库等等；
- 这些库函数在使用时，需要包含对应的头文件，比如使用printf，需要 #include <stdio.h>。

- **自定义函数：**
- 比如我们前面自定义实现的int32_t Sum(int32_t begin, int32_t end) 函数。

函数返回值

1. 函数的值只能通过return语句返回主调函数。return语句的一般形式为：

return 表达式

```
int32_t Sum(int32_t begin, int32_t end)
{
        int32_t sum = 0;
        int32_t i;

        for (i = begin; i <= end; i++)
        {
                sum += i;
        }
        return sum;
}
```

return 表达式的结果类型和函数定义中的返回值类型应保持一致

函数返回值

2. 函数如果没有返回值，需要使用void关键字修饰，表示“空”的意思：

```
void AFunc(void)
{
        printf("This is AFunc.\n");
}
```

return语句

- **return语句有2个作用：**

1. 跳出整个函数；
2. 将返回值带回给调用函数。

return语句

- **return语句可以根据程序的意图出现在函数的任意位置：**

```
bool islower(char c)
{
        if (c >= 'a' && c <= 'z')
        {
                printf("%c is lower.\n", c);
                return true;
        }
        printf("%c is not lower.\n", c);
        return false;
}
```

return语句

- **return语句可以根据程序的意图出现在函数的任意位置：**

```
void sgp30_init(void)
{
    if (g_sgp30_ctx == NULL)
    {
        if ((g_sgp30_ctx = (sgp30_content_t *)malloc(sizeof(sgp30_content_t))) == NULL)
        {
            HK_LOG_E(TAG, "sgp30_thread():malloc()");
            return;
        }
        memset(g_sgp30_ctx, 0, sizeof(sgp30_content_t));
    }

    BaseType_t ret = xTaskCreate(sgp30_thread, "sgp30_thread", 2 * 1024, NULL, 3, NULL);
    if (ret != pdPASS)
    {
        HK_LOG_I(TAG, "create thread fail, ret is %d", ret);
    }
}
```

THANK YOU!