

# 嵌入式C语言之- 参数的数值传递本质

讲师：叶大鹏

助力你成为优秀的电子工程师！



# 函数的参数

1. 如果函数有参数，在调用它时，必须传递正确数量和类型的参数：

```
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;
}
```

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

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

# 函数的参数

- 在调用函数时，传递给参数的可以是以下几种形式：

1. 常数值：Sum(1, 100);
2. 变量：Sum(a, b);
3. 函数的返回值：Sum(Sum(1, 10), 100);
4. 表达式：Sum(1 + 2, 100);

# 函数的参数

2. 如果函数没有参数，在定义它时，使用void进行修饰：

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

```
int main(void)
{
    AFunc();
    return 0;
}
```

# 参数的数值传递

```
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;
}
```

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

    int32_t a = 201;
    int32_t b = 300;
    res = Sum(a, b);
    return 0;
}
```

	 &begin	0x200003F4
	 &end	0x200003F8

	 &a	0x20000404
	 &b	0x20000400

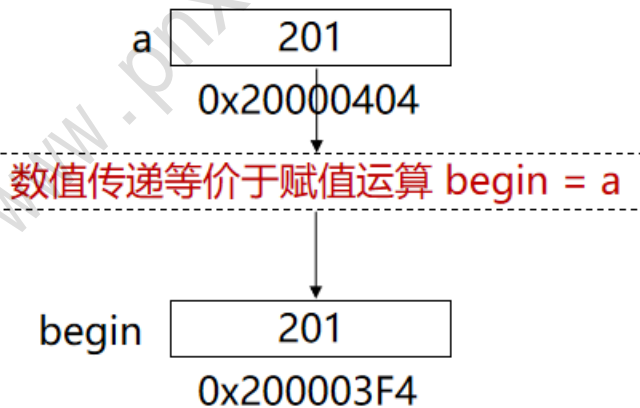
# 参数的数值传递

```
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;
}
```

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

    int32_t a = 201;
    int32_t b = 300;
    res = Sum(a, b);
    return 0;
}
```



# 这个程序能正确执行吗？

## ➤ 设计实现两个整数互换功能的函数

```
void Swap(int32_t x, int32_t y)
{
    int32_t temp = x;
    x = y;
    y = temp;
}
```

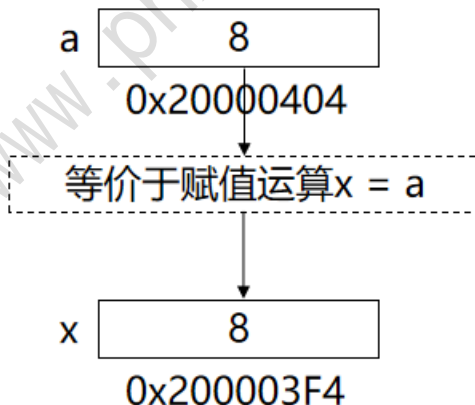
```
int main(void)
{
    int32_t a = 8;
    int32_t b = 9;
    Swap(a, b);
    printf("after swap, a = %d, b = %d \n", a, b);
    return 0;
}
```

# 这个程序能正确执行吗？

## ➤ 设计实现两个整数互换功能的函数

```
void Swap(int32_t x, int32_t y)
{
    int32_t temp = x;
    x = y;
    y = temp;
}
```

```
int main(void)
{
    int32_t a = 8;
    int32_t b = 9;
    Swap(a, b);
    printf("after swap, a = %d, b = %d \n", a, b);
    return 0;
}
```





# 如何实现两个整数互换功能？

- C语言有2种参数传递方式：

1. 数值传递；
2. 指针传递（引用传递）。

**THANK YOU!**