



سؤالات



آموزش - توابع

۱۰۰

آموزش - تابع بازگشتی

۱۰۰

مبنای آینه‌ای

۱۰۰

تابع فی

۱۰۰

جی سی پی دی - (One-liner)

۱۰۰

عدد به پیر به پیغمبر اول!!

۱۵۰

همه ارسال‌ها

ارسال‌های نهایی

جدول امتیازات



آموزش - توابع

یک تابع در واقع گروهی از دستورات است که با همدیگر یک هدف واحد را برآورده میکنند. هر برنامه ی C دارای حداقل یک تابع است (همان `main()`) و در عمده ی برنامه ها می توان توابع متعدد دیگری را هم مشاهده کرد.

شما می توانید با استفاده از توابع برنامه تان را با قطعات (یا اصطلاحاً module های) جدا از هم، بخش کنید. این که چگونه این تقسیم بندی را انجام می دهد فقط به دیدگاه شما نسبت به برنامه بستگی دارد ولی از لحاظ منطقی، یک تقسیم بندی باید طوری باشد که هر تابع وظیفه ی انجام یک کار به خصوص را عهده دار باشد.

اعلام یک تابع (declaration)، کامپایلر را راجع به نام تابع، نوع بازگشتی آن و پارامتر های ورودی آن آگاه می کند. **تعریف** یک تابع (definition)، در واقع بدنه ی اصلی همان تابع است.

تعریف یک تابع

```
return_type function_name (parameter list)
{
    body of the function
}
```

تعریف یک تابع در زبان C از یک هدر (Header) و یک بدنه (Body) تشکیل شده است. در زیر مشروح تمامی قسمت های تشکیل دهنده ی یک تابع را می توانید ببینید:

- Return Type:

یک تابع "می تواند" دارای مقدار بازگشتی باشد. `return_type` در واقع نوع داده ای است که تابع باز خواهد گرداند. بعضی از توابع کاری که به آنها محول شده است را بدون بازگرداند مقدار بازگشتی انجام می دهند. در این صورت `return_type` برابر کلمه ی کلیدی (keyword) `void` خواهد بود.

- Function Name:

این قسمت در واقع اسم تابع را نشان می دهد. اسم تابع و لیست پارامتر های ورودی تابع در کنار هم، امضای تابع (نشان تابع) را تشکیل می دهند. (Function Signature)

- Parameters:

هر پارامتر در واقع مانند یک placeholder است. هنگامی که یک تابع فراخوانی می شود، مقداری را به عنوان پارامتر به تابع پاس می دهیم. این مقدار را پارامتر اصلی و یا یک آرگومان می نامیم. لیست پارامتر ها در واقع به نوع، ترتیب و تعداد پارامتر های یک تابع اطلاق می شد. پارامتر ها برای یک تابع اختیاری هستند، در واقع می توانیم تابع بدون پارامتر ورودی داشته باشیم.

- Function Body:

بدنه ی یک تابع در واقع مجموعه ای از دستورات است که کار و هدف اصلی یک تابع را مشخص می کنند.

به عنوان مثال، تابع زیر با گرفتن دو آرگومان `num1` و `num2`، مقدار ماکزیمم این دو را باز می گرداند.

```
1  /* function returning the max between two numbers */
2  int max(int num1, int num2) {
```

```
3
4  /* local variable declaration */
5  int result;
6
7  if (num1 > num2)
8      result = num1;
9  else
10     result = num2;
11
12     return result;
13 }
```

اعلام توابع

همانطور که گفته شد، اعلام یک تابع کامپایلر را راجع به نام تابع و چگونگی فراخوانی تابع آگاه می کند. بدنه ی اصلی تابع را می توان جداگانه تعریف کرد. اعلام یک تابع دارای قسمت های زیر است:

```
return_type function_name (parameter list);
```

برای مثال بالا (تابع max())، اعلام تابع یه صورت زیر است:

```
int max(int num1, int num2);
```

اسامی پارامتر ها در اعلام یک تابع ضروری نیستند، و فقط نوع آن ها حائز اهمیت است. بنابراین اعلام زیر نیز یک اعلام درست است:

```
int max(int, int);
```

اعلام (یا همان امضای) یک تابع هنگامی مورد نیاز است که تابعی را در یک فایل سورس تعریف کرده اید و می خواهید از آن در یک فایل دیگر استفاده کنید. در این صورت باید اعلام تابع را در بالای فایلی که تابع را در آن فراخوانی کرده اید، بنویسید.

فراخوانی یک تابع

هنگامی که در تابع C را ایجاد می کنید، تعریفی هم برای آن ارائه می کنید که در واقع دستور کار آن تابع می باشد. برای استفاده از تابع برای انجام آن دستور کار، باید آن را فراخوانی کنید. هنگامی که یک برنامه تابعی را فراخوانی می کند، کنترل برنامه به تابع فراخوانی شده منتقل خواهد شد. تابع فراخوانی شد دستور کار تعریف شده برایش را اجرا می کند و هنگامی که به آکولاد انتهای تعریف تابع و یا یک دستور return می رسد، کنترل را به برنامه ی اصلی باز می گرداند. برای فراخوانی یک تابع کافیست که پارامتر های لازم را (با ترتیبی که در اعلام تابع مشخص کرده اید) را به تابع پاس بدهید و با استفاده از نام تابع و این پارامتر ها از مقدار بازگشتی (در صورت وجود) استفاده کنید.

به مثال زیر دقت کنید:

```
1 | #include <stdio.h>
2 |
3 | /* function declaration */
4 | int max(int num1, int num2);
5 |
6 | int main () {
7 |
8 |     /* local variable definition */
9 |     int a = 100;
10 |    int b = 200;
```

```

11     int ret;
12
13     /* calling a function to get max value */
14     ret = max(a, b);
15
16     printf( "Max value is : %d\n", ret );
17
18     return 0;
19 }
20
21 /* function returning the max between two numbers */
22 int max(int num1, int num2) {
23
24     /* local variable declaration */
25     int result;
26
27     if (num1 > num2)
28         result = num1;
29     else
30         result = num2;
31
32     return result;
33 }

```

پس از کامپایل و اجرا، این برنامه خروجی زیر را تولید خواهد کرد:

```
Max value is : 200
```

آرگومان های یک تابع

اگر یک تابع آرگومان ورودی دریافت می کند، باید متغیر هایی را اعلام کند که مقادیر آن آرگومان ها را می پذیرند. به این متغیر ها، پارامتر های رسمی تابع می گوییم. پارامتر های رسمی، مانند هر متغیر محلی دیگر درون یک تابع، در حین فراخوانی تابع ایجاد می شوند و در حین خروج نیز نابود خواهند شد. پاس دادن آرگومان ها به دو صورت کلی انجام می گیرد که اصطلاحاً Call by value و Call by reference خوانده می شوند. در مورد Call by reference بعداً بیشتر بحث خواهد شد ولی فعلاً تمامی توابع در C به صورت Call by value صدا زده می شوند، به این ترتیب که به جای پاس دادن مرجع یک متغیر، مقدار آن را به تابع پاس می دهیم. (بنابراین با توجه به این که پارامتر های رسمی در فقط در حین فعالیت تابع وجود خارجی دارند، نمی توانید برای مثال، با استفاده از Call by value در یک تابع (مثلاً swap) مقدار ذخیره شده در دو متغیر a و b را با هم تعویض کنید. چرا؟)

تمرین: سعی کنید که با نوشتن یک تابع swap و با استفاده از روش Call by value مقدار دو متغیر a و b را که درون main تعریف شده اند، با هم تعویض کنید و علت شکست این عمل را برای خودتان توجیه کنید(با Debug کردن برنامه).

ارسال پاسخ برای این سؤال

مهلت تمرین تمام شده است.



محصولات

آموزش برنامه نویسی
آگهی های استخدام
سؤالات برنامه نویسی

منابع

کوئرا بلاگ
ماشین حساب حقوق برنامه نویسان
آمارهای دنیای برنامه نویسی

رویدادها

کدکاپ
اسکیل آپ
نمایشگاه کارآموزشو

با کوئرا

همکاری با ما
تماس با ما
درباره ما



ساخته‌شده با افتخار در ایران | ۱۳۹۴ - ۱۴۰۱

فوانین و مقررات
حمایت از مسابقات

تربیس‌وی

عضویت در خبرنامه

مسابقات

کلاس‌ها

پلتفرم استخدامی

کوئرا جونیور