

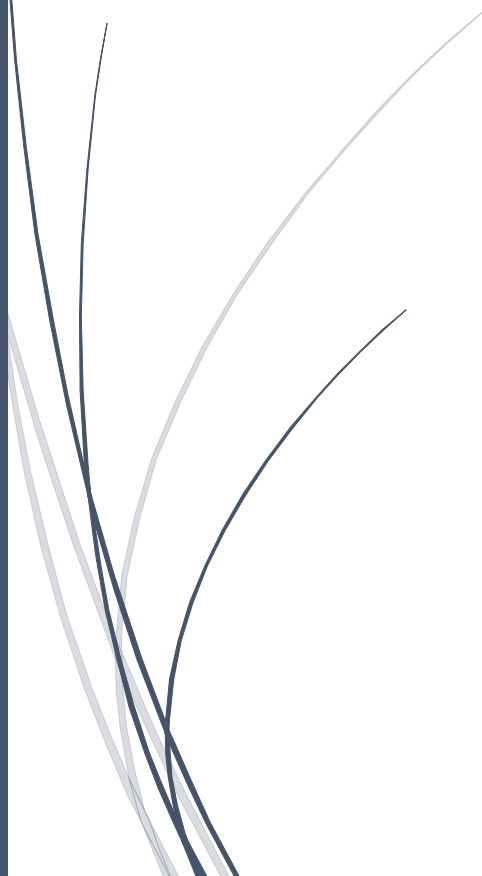
مدرس: آروین خسروی

دوره رایگان آموزش زبان C

[صفر تا پیشرفته]

مالکیت آموزش : سایت آموزشی جاواپرو

www.javapro.ir





آموزش زبان برنامه نویسی C

به نام خدا

تقدیم به هموطنان عزیزم

زبان C را با لذت یاد بگیر!



آموزش زبان برنامه نویسی C

آموزش زبان برنامه نویسی C

جلسه دوم

مقدمات، متغیرها و عملگرها در زبان برنامه نویسی C

C را ساده، آسان و شیرین بنوشید!!!



این جلسه آموزشی رایگان است، فروش و ویرایش آن ممنوع و حرام می باشد. اما این کتاب را می توانید همین جور که هست در سایت و شبکه اجتماعی خود به اشتراک بگذارید.



آموزش زبان برنامه نویسی C

ارتباط با ما:

سایت: www.javapro.ir

ایمیل: RAHMAN.ZARIE92@GMAIL.COM

کانال تلگرام:

[@javalike](https://t.me/javalike)

گروه پرسش و پاسخ برنامه نویسی :

[@javapro_ir](https://t.me/javapro_ir)



آموزش زبان برنامه نویسی C

گام به گام این سرفصل ها و مباحث را به صورت رایگان آموزش داده و در سایت به اشتراک می‌گذاریم. اگر تمایل داری تو هم در اشتراک رایگان این آموزش سهم داشته باشی و علاقمند هستی که خیلی سریع به اشتراک گذاشته شود از سایت آموزشی جاواپرو حمایت مالی کن! شاید این موضوع برای کم اهمیت باشد اما برای ما خیلی اتفاق بزرگی است!

[جهت حمایت مالی از جاواپرو اینجا کلیک کنید.](#)



آموزش زبان برنامه نویسی C

جلسه دوم | مقدمات، متغیرها و عملگرها در زبان برنامه نویسی C

مقدمات زبان برنامه نویسی C :

زبان C در سال ۱۹۷۲ توسط دنیس ریچی طراحی شد. این زبان تکامل یافته زبان BCPL است که طراح آن هارتین ریچاردز است.

زبان C از زبان های برنامه نویسی سطح میانی است ، زبان برنامه نویسی سطح میانی به این معنا است که به صورت مستقیم به حافظه دسترسی دارد و میتواند همانند زبان های سطح بالا ، خوانا باشد.

زبان های سطح بالا	زبان های سطح میانی	زبان های سطح پایین
پاسکال	جاوا	اسمبلی
کوبول	C	ماکرو اسمبلر
بیسک	C++	

جدول ۱-۱ سطح زبان های برنامه نویسی

C زبان برنامه نویسی سیستم است. برنامه های سیستم برنامه هایی هستند مانند کامپایلر ، مفسر و ادیتور متن.

این زبان کلمات کلیدی کمی دارد (۳۰ کلمه کلیدی) ولی تعداد کلمات نشانه از ضعف زبان نیست.

برخی از ویژگی های زبان برنامه نویسی C :

- تمامی دستورات این زبان به ; ختم میشود

- تمامی دستورات میتوانند چندین سطر ادامه داشته باشند



آموزش زبان برنامه نویسی C

- به دلیل وجود ; چندین دستور میتوانند در یک خط نوشته شوند

- برای توضیحات میتوان متن را درون /**/ نوشت

متغیرها :

متغیر نامی برای کلمات حافظه است که داده ها درون آن قرار میگیرند و ممکن است در طول اجرای برنامه تغییر کنند.

برای مراجعه به متغیرها از نام آنها استفاده میکنیم. نام متغیرها نمیتوانند با اعداد شروع شوند و نمیشود در میان حروف یک متغیر از فاصله یا space استفاده کرد.

نام متغیر میتواند در هر مقدار طولی تعریف شد , اما کامپایلر فقط ۳۱ حرف اول را در نظر میگیرد.

تعریف متغیر :

همانطور که گفته شد متغیرها محل ذخیره سازی اطلاعات در حافظه هستند , و اطلاعات نیاز به داشتن نوع دارند. برای مثال چند نوع اطلاعات را در جدول ۲-۱ میتوانید ببینید.

نوع داده	روش تعریف
یک کارکتر	char
یک رشته	String
یک عدد حداکثر ۱۶ رقمی	int
یک عدد اعشاری حداکثر ۱۶ رقمی	float
مقدار true یا false	bool

جدول ۲-۲ انواع داده

تعریف متغیر در زبان C یک روش کلی دارد و برای تعریف آن به صورت زیر عمل میکنیم:



آموزش زبان برنامه نویسی C

نوع داده ; مقدار متغیر

مثال ۱-۱:

```
#include <stdio.h>
#include <conio.h>

int main(){

    int x = 1234;

    getch()
    return 0;
}
```

در مثال بالا در مرحله اول نوع متغیر را int به معنای یک عدد صحیح ۱۶ رقمی تعریف کردیم و با گذاشتن یک فاصله اسم آن را x قرار دادیم.

در مرحله بعدی یک = گذاشته و سپس داده آن را عدد ۱۲۳۴ تعریف کردیم.

در مرحله آخر آن را نیز با یک ; به پایان رساندیم تا به کامپایلر بگوییم این سطر از کد تمام شده است.

مثال ۱-۲:

```
#include <stdio.h>
#include <conio.h>

int main(){

    float y = 3.6789;

    getch()
    return 0;
}
```




آموزش زبان برنامه نویسی C

}

در مثال بالا در مرحله اول نوع متغییر را float به معنای یک عدد اعشاری ۱۶ رقمی تعریف کردیم و با گذاشتن یک فاصله اسم آن را x قرار دادیم.

در مرحله بعدی یک = گذاشته و سپس داده آن را عدد ۳/۶۷۸۹ تعریف کردیم.

برای نوشتن نوشتن اعشار بعد از خود عدد یک نقطه گذاشته و بدون هیچ فاصله ای عدد اعشار را به آن اضافه میکنیم.

در مرحله آخر آن را نیز با یک ; به پایان رساندیم تا به کامپایلر بگوییم این سطر از کد تمام شده است.

• مثال ۱-۳:

```
#include <stdio.h>
#include <conio.h>

int main(){

    char k = 'a';
    char l = 'javapro';

    getch()
    return 0;
}
```

در مثال بالا در مرحله اول یک متغیر از نوع char یا یک کارکتر تعریف کردیم.

سپس نام آن را k قرار داده و داده آن را درون یک کوتیشن قرار داده ایم و در آخر ; را به آن اضافه کردیم.



آموزش زبان برنامه نویسی C

در خط دوم در مرحله اول یک متغیر از نوع char یا یک کارکتر تعریف کردیم. سپس نام آن را L قرار داده و داده آن را درون یک کوتیشن قرار داده ایم و در آخر ; را به آن اضافه کردیم.

* نکته *

در سطر دوم متغیر L برابر با داده z است. زیرا نوع char فقط یک کارکتر را پشتیبانی میکند و اگر بیشتر از یک کارکتر نوشته شود, کامپایلر حرف یا کارکتر اول را در نظر میگیرد.

مثال ۱-۴:

```
#include <stdio.h>
#include <conio.h>

int main(){

    int x = 14;
    int y;
    y = x;

    getch()
    return 0;
}
```

در مثال بالا اول یک متغیر با نام x تعریف کردیم و مقدار آن را ۱۴ قرار دادیم.

سپس یک متغیر با نام y ایجاد کردیم و مقداری به آن ندادیم. به دلیل اینکه به آن مقداری نداده ایم, متغیر y در حافظه به صورت NULL تعریف میشود. سپس در سطر سوم مقدار متغیر y را با مقدار متغیر x برابر قرار داده ایم



آموزش زبان برنامه نویسی C

تعریف ثوابت :

ثوابت متغیر هایی هستند که در طول برنامه تغییر نمیکنند و همیشه یک مقدار را دارند. ثوابت به دو دسته #define و دسته const تقسیم میشوند . توضیحاتی تکمیلی را میتوانید در جدول ۱-۳ ببینید.

نوع داده	روش تعریف
مقادیر اعشاری تا ۱۶ رقم	#define
بستگی به نیاز برنامه	const

جدول ۱-۳ انواع ثوابت

ثابت #define فقط مقادیر اعشاری تا ۱۶ رقم را پشتیبانی میکند و به صورت زیر تعریف میشود:

```
#include <stdio.h>
#include <conio.h>

#define PI 3.14

int main(){

    getch()
    return 0;
}
```

دقت کنید برای قرار دادن یک مقدار در این متغیر نباید از علامت = استفاده کنیم



آموزش زبان برنامه نویسی C

ثابت `const` تمامی مقادیری که در بخش های قبل توضیح داده شد را پشتیبانی میکند و به صورت زیر نوشته میشود:

```
#include <stdio.h>
#include <conio.h>

const int x = 700;
const float y = 3.6789;
const char L = 'j';

int main(){

    getch()
    return 0;
}
```

برخلاف ثابت `#define` ثابت `const` نیاز به علامت ؛ در اخر سطر است و بدون گذاشتن این علامت در بخش کامپایل برنامه با مشکل مواجه میشوید.

عملگرها :

عملگرها علامت های خاصی هستند که اغلب عملیات های ریاضی را انجام میدهند ، برخی از آنها در جدول ۴-۱ آمده است.

کار انجام شده	علامت
جمع کردن	+
تفریق	-
ضرب	*
تقسیم	/

جدول ۴-۱ انواع عملگرها



آموزش زبان برنامه نویسی C

عملگر های محاسباتی :

عملگر هایی هستند که عملیات ریاضی را انجام میدهند و برخی از آنها را میتوانید در جدول ۲-۴ ببینید.

کار انجام شده	علامت
جمع کردن	+
تفریق	-
ضرب	*
تقسیم	/
یکی به مقدار عدد اضافه می کند	++
یکی از مقدار عدد کم می کند	--

جدول ۲-۴ عملگر های محاسباتی

برای استفاده از این عملگر ها نوع داده ما حتما باید از نوع float یا باید از نوع int باشد، در غیر این صورت با ارور مواجه میشویم.

مثال ۱-۴ :

```
#include <stdio.h>
#include <conio.h>

int main(){

    int x = 13;
    x++;

    getch()
    return 0;
}
```



آموزش زبان برنامه نویسی C

در مثال بالا در مرحله اول یک متغیر با نام x تعریف کردیم و مقدار آن را ۱۳ قرار دادیم ، سپس با استفاده از عملگر ++ یک عدد به مقدار آن اضافه کردیم.

مثال ۲-۴ :

```
#include <stdio.h>
#include <conio.h>

int main(){

    int x = 13;
    x--;

    getch()
    return 0;
}
```

در مثال بالا در مرحله اول یک متغیر با نام x تعریف کردیم و مقدار آن را ۱۳ قرار دادیم ، سپس با استفاده از عملگر -- یک عدد از مقدار آن کم کردیم.

عملگر های رابطه ای :

عملگر های رابطه ای ارتباط بین علوند ها را مشخص میکنند. برای مثال بزرگتر یا کوچکتر یا برابری و مخالف بودن. در جدول ۵-۱ میتوانید اقلب این عملگر ها را ببینید.

نام	علامت
بزرگتر	<
کوچکتر	>



آموزش زبان برنامه نویسی C

مساوی	==
بزرگتر مساوی	>=
کوچکتر مساوی	<=

جدول ۱-۵ عملگر های رابطه ای

مثال ۱-۵ :

```
#include <stdio.h>
#include <conio.h>

int main(){

    int x = 13;
    m = x >= 5;

    getch()
    return 0;
}
```

در مثال بالا در مرحله اول متغییری را تعریف کردیم و نام آن را x تعریف کردیم و مقدار آن را با عدد ۱۳ برابر قرار دادیم. در سطر بعدی متغیر m را تعریف کردیم و مقدار آن را نوشتیم. در قطعه کد بالا به ارور مواجه میشویم زیرا ما متغیر x را با عدد ۵ مقایسه کردیم و این کار یک عدد یا مقدار نیست که متغیر m آن را داشته باشد. از این عملگر ها معمولا در دستورات شرطی استفاده میشود که آن را در فصل های بعد توضیح داده ایم.

عملگر های منطقی :



آموزش زبان برنامه نویسی C

عملگرهای منطقی بر روی عبارت های منطقی کار میکنند. عبارات منطقی عباراتی هستند که یک مقدار درست یا نادرست را در پی دارند. برخی از این عملگرها در جدول ۱-۶ آمده است.

نام	علامت
نقیض	!
و	&&
یا	

جدول ۱-۶ برخی از عملگرهای منطقی

مثال ۱-۶:

```
int x = 15;
int y = 50;
x || y == 50;
```

در مثال بالا دو متغیر x و y را تعریف کردیم و مقادیر آنها را ۱۵ و ۵۰ قرار داده ایم.

در سطر آخر گفته ایم اگر متغیر x برابر با ۵۰ بود یا متغیر y برابر با ۵۰ بود کاری را انجام دهد. همانطور که در قبل گفته شد این عملگرها برای دستورات شرطی بکار میروند که در فصل های آینده میخوانید. توجه شود که استفاده در تعداد عملگر محدودیتی ندارد و میتوان از یک عملگر در یک سطر چندین بار استفاده کرد. مانند مثال زیر:

```
x==4 || x==5 || x==9 || x==10;
```




آموزش زبان برنامه نویسی C

عملگرهای & و * :

همانطور که در گذشته گفته شد، متغییر دارای نام و مقداری در حافظه است و یک متغییر در حافظه دارای آدرس و مقدار مشخصی است. با استفاده از عملگر & میتوان آدرس یک متغییر را به دست آورد و با استفاده از عملگر * میتوان دسترسی مستقیم به مقدار یک آدرس داشت.

مثال ۷-۱ :

```
#include <stdio.h>
#include <conio.h>

int main(){

    float p = &x;
    *p = 5;

    getch()
    return 0;
}
```

در مثال بالا یک متغییر به نام p تعریف کردیم و مقدار آن را برابر با آدرس متغییر x قرار دادیم. سپس مقدار متغییر p که همان آدرس x است را برابر با ۵ قرار داده ایم. یعنی الان متغییر x برابر است با ۵.

عملگر؟ :

این عملگر یک عبارت منطقی را میسازد و در صورت درستی یا نادرستی عبارت کاری را انجام میدهد. برای درک بهتر به مثال زیر دقت کنید.

```
#include <stdio.h>
#include <conio.h>
```



آموزش زبان برنامه نویسی C

```
int main(){
    int x = 10;
    int y = x>20 ? x*2:x*3;

    getch()
    return 0;
}
```

در سطر اول متغییری با نام x تعریف کردیم و مقدار آن را برابر ۱۰ قرار داده ایم.

در سطر دوم متغییر y را تعریف کردیم و گفتیم اگر متغییر x بزرگتر از ۲۰ بود ، متغییر y برابر است با $x*2$ و اگر این عبارت درست نبود ، متغییر y برابر است با $x*3$ که در مثال بالا متغییر y برابر است با عدد ۳۰ زیرا $x>20$ نیست و کوچکتر از آن است و عبارت دوم در نظر گرفته میشود.

تمرینات:

۱. ویژگی های زبان C را بگویید
۲. ثوابت را تعریف کرده و فرق میان `define#` و `const` را بگویید.
۳. دستوراتی بنویسید که متغییر x را با عدد ۵ برابر قرار دهد.
۴. برای اضافه کردن یک عدد به مقدار یک متغییر از کدام عملگر استفاده میشود و نوع این عملگر چیست؟
۵. برنامه ای بنویسید که متغییر x را با عدد ۱۲/۹۸۷ برابر قرار دهد و بگوید اگر متغییر x از عدد ۱۲/۵ کوچکتر و مساوی بود متغییر y را با عدد ۱۲ و در غیر این صورت با عدد ۲۴ برابر قرار دهد
۶. برنامه ای بنویسید که مقدار x را با عدد ۷۶ برابر قرار دهد و از آن یک مقدار کم کند و برابر x قرار دهد.
۷. برای مسیله زیر یک برنامه بنویسید:

دانش آموزی ۵ نمره خود را درون ۵ متغیر قرار داده است. میخواهد ببیند کدام یک از این متغیر ها برابر با عدد

۷۷ است. (هر ۵ متغیر همزمان مقایسه شوند)



آموزش زبان برنامه نویسی C

سایت آموزشی رایگان جاواپرو

www.JAVAPRO.ir

آموزش زبان c را با تجربه شخصی و به زبان خودهونی یاد بگیرید!!!!

بازدید از کانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.

دخل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاواپرو به دور از اخلاق حرفه ای ست و حرام می باشد.