

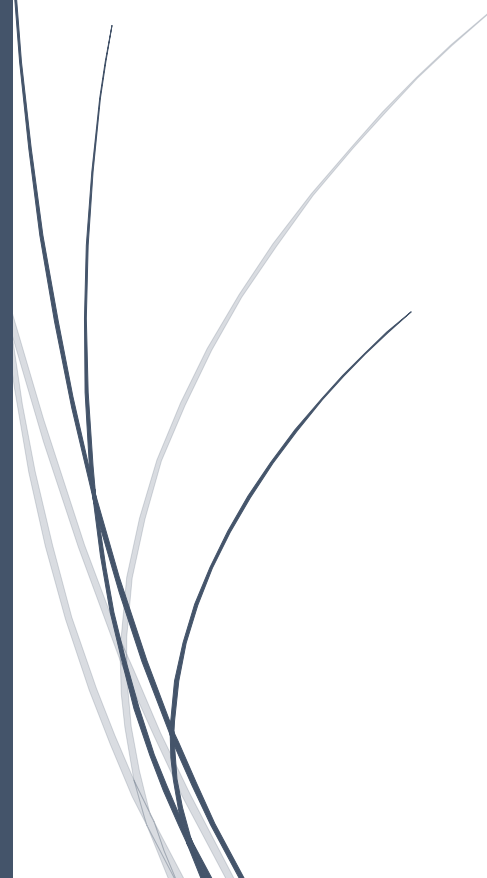
مدرس: آروین خسروی

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 سوم

حلقه ها و دستورات شرطی

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 سوم | حلقه ها و دستورات شرطی در زبان برنامه نویسی C

ساختارهای تکرار:

ساختارهای تکرار، تحت شرایط خاصی، یک یا چند دستور را چندین بار اجرا می کنند. به عنوان مثال، اگر بخواهیم تعدادها عدد را از ورودی بخوانیم و آنها را با هم جمع کنیم. باید عمل خواندن عدد را ۱۰۰ بار تکرار کنیم. ساختارهای تکرار در زبانهای برنامه سازی مختلف به شکلهای گوناگونی مورد استفاده قرار می گیرند. این ساختارها را در زبان C مورد بررسی قرار می دهیم.

حلقه for:

ساختار تکرار for یکی از امکانات ایجاد حلقه است و معمولاً در حالتی که تعداد دفعات تکرار حلقه از قبل مشخص باشد، به کار می رود. در این ساختار، متغیری وجود دارد که تعداد دفعات تکرار حلقه را کنترل میکند. این متغیر را شمارنده یا اندیس حلقه تکرار می نامیم. اندیس حلقه دارای یک مقدار اولیه است و در هر بار اجرای دستورات حلقه، مقداری به آن اضافه می شود. این مقدار را که پس از هر بار اجرای حلقه به شمارنده اضافه می شود، گام حرکت گویند. گام حرکت می تواند عددی صحیح و اعشاری، مثبت یا منفی و یا کاراکتری باشد. یکی دیگر از اجزای حلقه for، شرط حلقه است. شرط حلقه مشخص می کند که دستورات داخل حلقه تا چه زمانی باید اجرا شوند. اگر این شرط دارای ارزش درستی باشد، دستورات داخل حلقه اجرا می شوند وگرنه کنترل برنامه



آموزش زبان برنامه نویسی C

از حلقه تکرار خارج می شود . اندیس حلقه تکرار می تواند عددی منفی ، مثبت ، صحیح و یا اعشاری و کاراکتری باشد . دستور for را به شکل زیر می توان به کار برد :

For { (گام حرکت ; شرط حلقه ; مقدار اولیه اندیس حلقه)

دستورات:

}

مثال ۱-۱:

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

int main(){

    int sum=0, num, i;
    float ave;
    int n = 2;
    for (i = 0; i<n; i++){
        printf("Adad shomare %d ra vared konid: ", n+i);
        scanf("%d",&num);
        sum += num;
        printf("\n");
    }
    ave = sum / n;
    printf("the ave is %f ", ave);

    getch();
    return 0;
}
```

خروجی:

```
Adad shomare 1 ra vared konid: 10
Adad shomare 2 ra vared konid: 20
the ave is 15
```



آموزش زبان برنامه نویسی C

تحلیل :

در برنامه بالا متغییر n را اساس حلقه for قرار دادیم و این حلقه را ۲ بار اجرا کردیم سپس ۲ عدد را از کاربر دریافت کرده و آن را در متغییر sum ذخیره کردیم. توجه کنید که متغییر sum باید در اول مقداری را داشته باشد و اگر نه برنامه با مشکل مواجه میشود. سپس بعد از اتمام حلقه متغییر ave که مخفف شده عبارت میانگین است را با مجموع دو عدد وارد شده که همان sum است بر n که همان تعداد عدد است تقسیم کردیم تا میانگین را بدست آوریم. سپس به خط بعدی رفته و متغییر ave که از نوع float یا عدد اعشاری است چاپ کردیم.

حلقه های تودرتو:

وقتی حلقه تکراری در داخل حلقه تکرار دیگر قرار داشته باشد ، می گوییم که حلقه های تودرتو ایجاد شده اند . قانونی که بر حلقه های تکرار تو در تو حاکم است این است که ، به ازای هر بار اجرای حلقه تکرار خارجی ، حلقه تکرار داخلی به طور کامل اجرا می شود . ضمناً ، انتهای حلقه تکرار داخلی ، زودتر از حلقه تکرار خارجی مشخص می شود . به عنوان مثال ، در دستورات زیر ، حلقه تکرار با اندیس i ، حلقه خارجی و حلقه تکرار بانديس j ، حلقه تکرار داخلی است :

```
For (i = 0; i<10 ; i++){  
    ...  
    For (j = 0 ; j<10 ; j++){  
        ...  
    }  
    ...  
}
```




آموزش زبان برنامه نویسی C

مثال جدول ضرب با حلقه های تو در تو:

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

int main(){
    int i,j;
    for(i = 1 ; i<=4 ; i++){
        for (j = 1 ; j<=4 ; j++){
            printf("%3d", i*j);
        }
        printf("\n");
    }
    getch();
    return 0;
}
```

خروجی:

```
1  2  3  4
2  4  6  8
3  6  9 12
4  8 12 16
```

تحلیل:

اول از همه اندیس های حلقه های خود را انتخاب کرده که همان متغیر های i و j هستند را از نوع عدد صحیح یا همان `int` تعریف کردیم و اساس هر دو حلقه را عدد 4 در نظر گرفتیم و سپس در حلقه دوم یا حلقه داخلی خود مقدار $i*j$ را به صورت `%3d` که همان عدد صحیح مرتب شده است قرار دادیم تا جدول در خروجی منظم و زیبا نشان داده شود.



آموزش زبان برنامه نویسی C

ساختار while :

ساختار while یکی دیگر از ساختار حلقه ها میباشد و روش استفاده از آن به صورت

زیر می باشد :

```
while (شرط) {  
    دستورات 1  
    دستورات 2  
    .  
    دستورات n  
}
```

مثال حلقه while :

```
#include <stdio.h>  
#include <conio.h>  
  
int main(){  
  
    int x = 1;  
    printf("adad zoj 1 ta 20 : ");  
    while (x < 21){  
        if (x%2 == 0){  
            printf("%d ",x);  
        }  
    }  
  
    getch();  
    return 0;  
}
```

خروجی:

```
adad zoj 1 ta 20 : 2 4 6 8 10 12 14 16 18 20
```



آموزش زبان برنامه نویسی C

تحلیل:

در حلقه بالا شرط ما این است تا زمانی که متغیر x کوچکتر از عدد ۲۱ بود حلقه اجرا میشود و در صورت درست نبودن این عبارت حلقه اجرا نمیشود. سپس درون دستورات گفتیم اگر متغیر x ما عددی زوج بود آن را نشان بده. زوج بودن متغیر را از این روش متوجه شدیم که گفتیم اگر باقیمانده تقسیم متغیر x بر ۲ برابر با ۰ بود دستور را انجام بدهد (علامت باقیمانده تقسیم % میباشد).

ساختار do while :

ساختار تکرار `do ... while` مانند ساختار تکرار `while` است؛ با این تفاوت که در ساختار `while`، شرط حلقه در ابتدای حلقه تست می شود، در حالی که در `do ... while` شرط حلقه در انتهای حلقه تست می گردد. بنابراین دستورات موجود در حلقه `do ... while`، در هر حال، حداقل یک بار اجرا می شوند. این ساختار به صورت زیر به کار می رود.

```
do {  
    دستورات 1  
    دستورات 2  
    .  
    دستورات n  
} while (شرط);
```



آموزش زبان برنامه نویسی C

مثال عدد شانس با do while :

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

int main(){

    int x = 37;
    int y = 1;
    int z;
    do {
        printf("Enter 1 num : ");
        scanf("%d",&z);
        if (z != x){
            printf("\n enter 1 more num : ");
            scanf("%d", &z);
        }
        else {
            printf("\n great");
            y = 0;
        }
    } while (y == 1);

    getch();
    return 0;
}
```

خروجی:

```
Enter 1 num : 28
Enter 1 num : 4
enter 1 more num : 37
great
```

تحلیل :



آموزش زبان برنامه نویسی C

در مرحله اول یک متغیر از کاربر دریافت کرده و چک می کنیم اگر متغیر با متغیر x که همان عدد ۳۷ است برابر بود عبارتی را نشان داده و متغیر y را با عدد e قرار داده و شرط حلقه برقرار نباشد و حلقه تمام شود. و اگر عدد وارد شده با ۳۷ برابر نباشد دوباره حلقه اجرا میشود زیرا به متغیر y دست نزدیم و مقدار آن را تغییر نداده به همین دلیل شرط حلقه درست است و دستورات دوباره اجرا میشوند.

از کدام حلقه استفاده کنیم؟

زمانی که تعداد دفعات اجرای برنامه مشخص باشد بهتر است از حلقه `for` استفاده کنیم و در شرایطی که هر بار باید یک شرطی بخصوص چک شود بهتر است از حلقه `while` و حلقه `do ... while` برای راحتی کار استفاده کنیم. در غیر این صورت زبان `C` زبانی بی حد و مرز است و در هر شرایطی میتوان از هر حلقه ای استفاده کرد ولی مشخصا کاری آسان نیست.

ساختارهای شرطی:

همانطور که دیدید، ساختارهای تکرار، برای تکرار اجرای دستورات مورد استفاده قرار می گیرند. اما اگر بخواهیم، تحت شرایطی، تعدادی از دستورات اجرا شوند و یا تعدادی دیگر از دستورات اجرا نشود، باید از ساختارهای تصمیم استفاده کنیم. این ساختارها شرطی را تست کرده در صورت درست بودن شرط، مجموعه ای از دستورات را انجام می دهند. در زبان `C` چندین ساختار تصمیم وجود دارد که آنها را در این بخش بررسی میکنیم.

دستور شرطی `if`:



آموزش زبان برنامه نویسی C

ساختار if که نام دیگرش دستور انتقال کنترل شرطی است ، شرطی را تست میکند و در صورتی که شرط دارای ارزش درستی باشد ، مجموعه ای از دستورات را اجرا میکند . این دستور به صورت زیر به کار میرود.

```
if (شرط){  
دستورات  
}  
else {  
دستورات  
}
```

در هر یک از روشهای کاربرد ، چنانچه شرط مورد بررسی درست باشد ، دستور یا دستورات بعد از if وگرنه دستور یا دستورات بعد از else اجرا می شوند . اگر بیش از یک دستور بعد از if یا else بیابند ، آن دستورات باید در بین { و } قرار گیرند . دستور if می تواند فاقد قسمت else باشد . در این صورت ، چنانچه شرط مورد بررسی ، درست باشد ، دستورات بعد از if اجرا می شوند وگرنه بدون اجرای این دستورات ، کنترل اجرای برنامه از if خارج می شود .



آموزش زبان برنامه نویسی C

مثال عدد بزرگتر :

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

int main(){

    int i;
    int sum = 0;
    int inx = 0;
    for(i = 1 ; i<=5 ; i++){
        printf("num %d : ",i);
        scanf("%d",&inx);
        if (inx > sum){
            sum = inx;
        }
    }
    printf("big num is %d : ",sum);
    getch();
    return 0;
}
```

خروجی :

```
num 1 : 10
num 2 : 67
num 3 : 23
num 4 : 98
num 5 : 0
big num is : 98
```

تحلیل :

در مثال بالا در مرحله اول یک اندیس برای حلقه خود انتخاب کردیم و اسم آن را i نامیدیم سپس در هر مرحله به مدت ۵ بار یک عدد از کاربر دریافت کردیم و گفتیم اگر عدد وارد شده از قبلی بزرگتر بود ، آن را در متغییر sum ذخیره کن و اگر این شرط برقرار نبود کاری



آموزش زبان برنامه نویسی C

انجام نده و شرط را تمام کن ، سپس در مرحله آخر و بیرون از حلقه عدد بزرگتر یا همان متغیر sum را نمایش بده.

دستور شرطی else if :

اگر بخواهیم از دستور if برای تست شرطهای متعددی استفاده کنیم باید آنها را به طور تو در تو به کار ببریم . کاربرد if به صورت تو در تو ، نه تنها موجب طولانی شدن برنامه می شود ، بلکه از خوانایی برنامه نیز می کاهد . ساختار else if می تواند به جای if های تو در تو به کار گرفته شود و میزان خوانایی برنامه را بالا ببرد . برای روشن شدن موضوع ، به مثال زیر که یک مسئله را حل می کنند توجه کنید .

مثالی که یک عدد از کاربر دریافت کرده و چک کند این عدد چند رقمی است:

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

int main(){

    int x;
    printf("Enter the num: \n");
    scanf("%d",&x);
    if (x>0 && x<10){
        printf("ek raghami");
    }
    else if (x>10 && x<100){
        printf("do raghami");
    }
    else if (x>100 && x<1000){
        printf("se raghami");
    }

    getch();
}
```




آموزش زبان برنامه نویسی C

```
return 0;  
}
```

خروجی :

```
Enter the num: 345  
se raghami
```

تحلیل :

در مرحله اول یک عدد صحیح از کاربر دریافت کرده و بازه آن عدد را بدست میآوریم و تعداد رقم آن را در خروجی نشان میدهیم ، توجه کنید خوبی `else if` این است که اگر `if` قبلی درست باشد ، `else if` بعدی چک نمیشود و به سرعت برنامه اضافه میکند. همانطور که میدانید دستور `&&` به معنای `>` و `<` در برنامه نویسی میباشد و در صورتی شرط مقدار `true` یا درست را برمیگرداند که هر دو شرط نوشته شده درست باشد. اگر به جای `&&` عملگر `||` را میگذاشتیم اگر یکی از شرط ها درست بود مقدار `true` برگردانده میشود به دلیل اینکه این عملگر به معنای `>` یا `<` در برنامه نویسی زبان C و اکثر زبان ها میباشد.

انتقال کنترل غیر شرطی:

دستور `if` شرطی را بررسی کرده ، بر اساس نتیجه شرط ، دستورات عملیاتی را انجام می دهد . در C دستورات عملیاتی وجود دارند که بدون تست شرطی می توانند کنترل اجرای برنامه را از نقطه ای به نقطه دیگر منتقل کنند . این دستورات را انتقال کنترل غیر شرطی گویند



آموزش زبان برنامه نویسی C

دستور break :

این دستور موجب خروج از حلقه های تکرار می شود . نحوه کاربرد این دستور به صورت زیر است :

```
break;
```

اگر چند حلقه تو در تو وجود داشته باشد ، این دستور موجب خروج از داخلی ترین حلقه تکرار می شود . کاربرد دیگر این دستور ، خاتمه دادن به ساختار switch است که در ادامه بحث خواهد شد .

دستور continue :

این دستور در حلقه تکرار موجب انتقال کنترل به ابتدای حلقه می شود . پس از انتقال کنترل به ابتدای حلقه ، شرط حلقه مورد بررسی قرار می گیرد ، چنانچه شرط درست باشد ، اجرای دستورات حلقه ادامه می یابد وگرنه حلقه تکرار خاتمه می یابد .

مثال ۱-۲ :

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

int main(){

    int d = 1;
    char ch;
    while (d){
```



آموزش زبان برنامه نویسی C

```
printf("\n enter the char :");
scanf("%c",&ch);
if (ch == 'e'){
    d = 0;
    continue;
}
}
printf("\n ok bye");
getch();
return 0;
}
```

خروجی:

```
enter the char : q
enter the char : l
enter the char : m
enter the char : e
ok bye
```

تحلیل:

در مرحله اول متغیر d را تعریف کردیم و مقدار آن را برابر با عدد 1 قرار دادیم زیرا در حلقه while عدد 1 نشان گر مقدار true است و حلقه تا زمانی که مقدار d برابر با عدد 1 باشد کار میکند و برنامه بسته نمیشود. ما در هر مرحله 1 کارکتر از کاربر دریافت کرده و چک میکنیم که اگر کارکتر برابر با e بود ، مقدار متغیر d را به 0 تغییر بدهد زیرا اگر شرط حلقه while برابر با عدد 0 باشد این حلقه کار نمیکند و مقدار false را برمیگرداند. سپس اگر کاربر کارکتر e را وارد نکرد ، شرط ما کاری را انجام نمیدهد و حلقه از اول یک کارکتر از کاربر دریافت میکند.

دستور switch :



آموزش زبان برنامه نویسی C

ساختار switch یکی از ساختارهای جالب و مهم در زبان C است. از این ساختار برای تصمیم گیریهای چندگانه براساس مقادیر مختلف یک عبارت، استفاده می شود. به طور کلی، در تمام تصمیم گیریهایی که بیش از سه انتخاب وجود داشته باشد، بهتر است از ساختار switch استفاده شود. به عنوان مثال، فرض کنید، متغیری به نام دارید که این متغیر مقادیر ۱، ۷، ۹ و ۱۵ را می پذیرد و می خواهید براساس مقادیر مختلف x، تصمیم گیریهای متعددی انجام دهید. اگر x برابر با یک بود، «مجموعه دستورات ۱ <»، اگر x برابر با ۷ بود، «مجموعه دستورات ۲ <»، اگر x برابر با ۹ بود، «مجموعه دستورات ۳ <» و اگر x برابر با ۱۵ بود، «مجموعه دستورات ۴ <» اجرا شوند و اگر x با هیچکدام از اعداد برابر نبود مجموعه دستورات دیگری را انجام دهد.

ساختار اصلی switch :

```
Switch(متغیر){
    شرط :
        دستورات
}
```

مثال ۱-۳ :

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

int main(){
    int x;
    printf("enter the num : ");
    scanf("%d", &x);
    switch(x){
        case 1:
            printf("\n heloo");
            break;
        case 2:
```



آموزش زبان برنامه نویسی C

```

printf("\n javapro");
break;
case 3:
printf("\n Arvin");
break;
case 4:
printf("\n zare");
break;
default:
break;
}
getch();
return 0;
}

```

خروجی:

```

enter the num : 2
javapro

```

تحلیل :

در کد بالا یک متغیر از نوع اعداد صحیح دریافت کرده و چک میکنیم اگر این متغیر برابر با یکی از اعداد وارد شده باشد رشته مورد نظر چاپ شود در غیر این صورت که کد های این شرط در قسمت default نوشته میشود با یک دستور break شرط را خاتمه داده و به ادامه برنامه میپردازیم

تمرینات:

۱. برنامه ای که ۴ عدد را از ورودی دریافت کند و آن عددی که از همه کوچکتر بود را در خروجی نشان دهد
۲. برنامه ای که علامت \$ را به صورت یک جدول ۵ در ۵ چاپ کند
۳. برنامه کی سری فیبوناتچی را تا شماره الگو ۱۰ نشان دهد (به کمک حلقه ها)
۴. برنامه ای که ۲۰ عدد را از کاربر دریافت کرده و انها را جمع کرده و در خروجی نشان دهد
۵. برنامه که اعداد ۱ تا ۳۰ را در خروجی نمایش دهد
۶. برنامه ای که شکل زیر را در خروجی نشان دهد



آموزش زبان برنامه نویسی C

*

**

۷. برنامه ای بنویسید که ۵ عدد از ورودی دریافت کند و بگوید هر کدام از آنها چند رقمی است (به کمک حلقه ها و عدد باید حداکثر ۴ رقمی باشد)



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.