

بسم الله الرحمن الرحيم

موضوع مقاله: آموزش برنامه نویسی به زبان LITE C

مترجم: محمد غضنفری (MAMAD G.H.A)
Mamad5959@yahoo.com



بخش یازدهم: شاخه های If - Else

اگر (If) من درس بخوانم با سواد می شوم در غیر اینصورت (Else) من یک بی سوادم.

این فقط یک مثال ساده از شاخه های If - Else بود. اکنون شما می توانید در کد زیر این مثال را ببینید.

```
if (reading_Study > 1500)
    have_literacy( );
else
    miss_literacy( );
```

هنگامی که شما می خواهید بازیتان در برخی شرایط به شاخه های متفاوتی حرکت کند به دستورالعمل If نیاز دارید.

استفاده از اعداد تصادفی که نتیجه یک عملکرد ریاضی می باشد که پایه ایست در دستورالعمل If اکنون به کد زیر دقت کنید تابع do_something یک دستورالعمل اجرا کننده است.

```
if (some condition is true)
    do_something(); // execute this instruction (a single instruction!)
```

یا

```
if (some condition is true)
{
    // execute one or several instructions that are placed inside the curly brackets
}
```

یک مجموعه بیشتری از If در زیر است.

```
if (some condition is true)
{
    // execute one or several instructions that are placed inside the curly
    brackets
}
else
{
    // execute one or several instructions that are placed inside these
    curly brackets
}
```

دستورالعملی که درون قسمت Else گذاشته شده است. هنگامی اجرا می شود که در برخی مواقع دستور IF نادرست باشد. در اینجا یک نمونه قابل استفاده موجود است:

```
if (my_age > 65)
    income = 2000;
else // age less than or equal to 65?
    income = 3000;
```

این دستور تا حدی برای شما آشکار است. بالاخره یا income ۲۰۰۰ است و یا ۳۰۰۰. زیرا این دستور تنها یکی از دو شاخه ها را می تواند اجرا کند. و هر دو نمی تواند اجرا شوند. این ترسیم برخی پایان هایی است که می تواند اتفاق بیافتد:

- دستورالعمل شاخه های IF درون دو پرانتز می باشند.
 - دستورات درون پرانتز بعد از IF با وضعیت مقایسه می شود در برخی حالات ممکن است این دستور درست و ر برخی مواقع غلط باشد.
 - اگر بعد از بررسی شرایط دستور درون دو پرانتز درست بود آن را اجرا می کند.
 - و اگر بعد از بررسی شرایط دستور درون دو پرانتز نسبت به وضعیت غلط بود LITE C دستورات درون پرانتز را اجرا نمی کند و از آن رد می شود و به دستورات درون دو پرانتز دومی یعنی دستورات درون پرانتز ELSE می رسد.
- شما می توانید این فرایند شاخه شدن دستورات را ببینید. پس فایل script12 را باز کنید.

```

////////////////////////////////////
var weight = 300;
STRING* fluffy_str = "Fluffy!";
STRING* skinny_str = "Skinny!";
////////////////////////////////////
TEXT* weight_txt =
{
    pos_x = 300;
    pos_y = 230;
    flags = VISIBLE;
}
////////////////////////////////////
function main()
{
    video_mode = 7; // create a program window of 800x600 pixels
    screen_color.blue = 1; // make the background dark

    while (1)
    {
        if (weight > 200)
            (weight_txt.pstring)[0] = fluffy_str;
        else
            (weight_txt.pstring)[0] = skinny_str;
        wait (1);
    }
}

```

این کد زیاد پیچیده به نظر نمی‌یاد. خوب ما ابتدا یک متغیر با نام weight که مقدار ۳۰۰ هم به آن دادیم تعریف کردیم. سپس دو رشته "Fluffy!" و "Skinny!" را تعریف نمودیم.

همچنین ما کدهایی را برای ساختار متن مثل نوع فونت و موقعیت من بر روی صفحه نوشتیم.

سپس تابع مهم main را به همراه دو متغیر پیش فرض video_mode و screen_color.blue تعریف نمودیم.

سپس در این قسمت از کد ما دستورات مان را درون حلقه while (1) گذاشتیم.

```

while (1)
{
    if (weight > 200)
        (weight_txt.pstring)[0] = fluffy_str;
    else
        (weight_txt.pstring)[0] = skinny_str;
    wait (1);
}

```

اگر مقالات قبلی ما را خواند باشید حتما می‌توانید بفهمید که "wait (1);" چیست.

و اما ساختار حلقه while خیلی ساده است. که حلقه دستورات درون دو پرانتز را اجرا کرده و هنگامی که به انتهای دستور می رسد دوباره همه دستورات از ابتدا اجرا می گردند.

اکنون طبق حلقه اگر if مقدار متغیر weight بیشتر از ۲۰۰ باشد ما به weight_txt می گوئیم که متن واقع در رشته fluffy_str را در صفحه نمایش می دهیم. اما اگر این طور نباشد else و متغیر weight کمتر از عدد ۲۰۰ و یا خود عدد ۲۰۰ باشد ما به weight_txt می گوئیم که متن واقع در رشته skinny_str را در صفحه نمایش می دهیم. که اگر دقت کرد باشید از روش نقطه در داخل پرانتز های if و else استفاده می شود.

ممکن است تاکنون شما توانسته باشید حدس بزنید که pstring چیست. این یک خصوصیت متن هست که باعث می شود جایگاه متن تغییر کند. که از روش نقطه استفاده می کند. این رشته ها می توانند برای متنهايمان استفاده شوند. که برای هر متن یک رشته باید تعریف گردد. (بخش ششم را به یاد می آورید.)

اکنون اسکریپت را اجرا کنید. تا تصویری همچون شکل زیر را ببینید.



اکنون بر روی کلید **Tab** از روی صفحه کلید کلیک کنید و سپس کد زیر را بنویسید.

```
weight = 100;
```

سپس کلید **Enter** را از روی صفحه کلید فشار دهید. متن از "Fluffy!" به "Skinny!" تغییر می کند. زیرا شاخه ELSE فعال شده است. همانطور که گفتیم این شاخه هنگامی فعال می شود که مقدار متغیر کمتر یا مساوی با عدد ۲۰۰ شوند که شما همین الان مقدار متغیر را برابر عدد ۱۰۰ کردید که کمتر از عدد ۲۰۰ است و در نتیجه شاخه IF غیر فعال شده و شاخه ELSE فعال گردیده است.



اکنون بیا باید برنامه را طوری طراحی کنیم که وقتی متغیر `weight` برابر با ۰ گردید برنامه خود به خود قطع شود. اکنون باید کد زیر را به برنامه اضافه کنید.

```
if (weight == 0)
    sys_exit(NULL);
```

اکنون باید کدهای برنامه به شکل زیر باشد. کدهای سبز رنگ کدهایی هستند که به برنامه اضافه شدند.

```
////////////////////////////////////
#include <acknex.h>
#include <default.c>
////////////////////////////////////
var weight = 300;

STRING* fluffy_str = "Fluffy!";
STRING* skinny_str = "Skinny!";
////////////////////////////////////
TEXT* weight_txt =
{
    pos_x = 300;
    pos_y = 230;
    flags = VISIBLE;
}
////////////////////////////////////
function main()
{
    video_mode = 7; // create a program window of 800x600 pixels
    screen_color.blue = 1; // and the background dark
    while (1)
    {
        if (weight == 0)
            sys_exit(NULL);
        if (weight > 200)
            (weight_txt.pstring)[0] = fluffy_str;
        else
            (weight_txt.pstring)[0] = skinny_str;
        wait (1);
    }
}
```

کدهای جدید زیاد پیچیده به نظر نمی آیند شاید برای شما این سوال مطرح گردد که چرا ما از دو `==` در نسبت دادن ۰ به متغیر `weight` استفاده کردیم. چون ما می خواهیم متغیر `weight` را برابر با ۰ نشان دهیم و به همین علت از دو علامت مساوی استفاده کردیم. هنگامی ما باید از یک علامت مساوی استفاده کنیم که بخواهیم عدد ۰ را برای این متغیر تعریف کنیم اما اکنون ما نمی خواهیم عدد ۰ را تعریف کنیم بلکه می خواهیم بگوییم که این متغیر برابر با ۰ است.

این کدها به موتور می گویند که اگر متغیر `weight` برابر با ۰ گردید برنامه قطع گردد. در مورد متغیر `"sys_exit"` هم در بخش پنجم توضیحاتی داده ایم.

اکنون برنامه را اجرا کرده و با استفاده از کلید **Tab** از روی صفحه کلید بخش فرمان را باز کرده و دستور زیر را وارد کنید.

`weight = 0;`

اکنون برنامه باید قطع گردد.

نکته مهم: این نکته بسیار مهم و حائز اهمیت است که برخی ها در حلقه به جای گذاشتن دو علامت مساوی و برابر کردن متغیر از یک علامت مساوی استفاده می کنند و در حقیقت عدد ۰ را برای متغیر تعریف می کنند که خیلی این اشتباه رایج است و شما اید مراقب باشید که به این امر عادت نکنید و همیشه هنگام برنامه نویسی دقت کرده و بدانید که چه می نویسید. و یا این که برخی ها خیلی چیزها را جا به جا می نویسند و مثلاً به جای اینکه در قسمت چپ نام متغیر را بنویسند و در قسمت راست علامت عدد را بنویسند با این فکر که هیچ فرقی نمی کند این دو را جا به جا می نویسند مثلاً برای یک عبارت شرطی ممکن است اینگونه بنویسند:

```
if (0 == weight) // correct
    sys_exit(NULL);
```

این عبارت به این معنی است که ما مقدار `weight` را برابر با متغیر 0 کردیم و در کل همه چیز با اشتباه روبرو می شود در صورتی که **همیشه باید نام متغیر در سمت چپ و عبارتی که می خواهیم به آن نسبت یا با آن برابر بدانیم در سمت راست و در میان علامت مورد نظر گذاشته می شود.**

همچنین شما می توانید از برخی علامات برای بزرگ و کوچک نشان دادن استفاده کنید.

```
if (energy < 50) // simple expression inside the parenthesis
{
    health = 10;
}
```

شما می توانید حلقه تان را پیچیده تر بکنید همچون مثال زیر:
 if ((armor + energy * 0.25) > (health * (1 - class / 3))) // more complicated expression

```
{
    health += 5;
}
```

شما باید با علامات ریاضی که در برنامه نویسی مورد استفاده قرار می گیرند کاملاً تسلط داشته باشید. برای کسی که می خواهد یک برنامه نویس حرفه ای شود خیلی بد است که کار با این علامات را بلد نباشد.

نمونه کاربرد	عمل گر	موقعیت
If (a==b)	==	نمونه شماره ۱ = نمونه شماره ۲
If (a>b)	>	نمونه شماره ۱ بزرگتر از نمونه شماره ۲ است
If (a>=b)	>=	نمونه شماره ۱ بزرگتر و مساوی یا نمونه شماره ۲ است.
If (a<=b)	<	نمونه شماره ۱ کوچکتر از نمونه شماره ۲ است
If (a<=b)	<=	نمونه شماره ۱ کوچکتر و مساوی یا نمونه شماره ۲ است.
If (a!=b)	!=	نمونه شماره ۱ با نمونه شماره ۲ نمی تواند برابر باشد

شما برخی اوقات به چک کردن علامات نیاز دارید تصور کنید که می خواهید یک کد برای آتش اسلحه بنویسید.

- سلامتی بازیکن باید بالاتر از ۰ باشد. (چون اگر بازیکن مرده باشد دیگر نمی تواند شلیک کند).
- مقدار مهمات اسلحه باید بیشتر از ۰ باشد. (تا بازیکن بتواند شلیک کند).

برای این کار چندین روش وجود دارید که یکی از این روشها استفاده از شاخه های if می باشد.
 به عنوان مثال:

```
if (player_health > 0)
{
    if (ammo > 0)
    {
        fire_bullets();
    }
}
```

در ابتدا میزان سلامتی بازیکن با ۰ مقایسه می شود. و طبق علامت باید میزان سلامتی بزرگتر باشد. اگر بازیکن زنده بود $player_health > 0$ باید میزان مهمات آن مورد بررسی قرار بگیرد. که بیشتر از صفر باشد. اگر تعداد مهمات بازیکن برابر با صفر بود ($ammo \leq 0$) برنامه همه چیز را نادیده گرفته و برنامه از اول دوباره اجرا می شود اما اگر میزان مهمات بالاتر از ۰ بود تابع `fire_bullets` اجرا می شود.

خوب دوستان به انتهای این قسمت رسیدیم. سعی کنید با تمرینات بیشتر و تغییر دادن کارها و اعداد و علامات تمرینات بیشتری را در این قسمت مهم و حساس انجام دهید. در مبحث بعد ما به استفاده از صفحه کلید برای بازی مان می رسیم که برای ساخت بازی این یکی از مباحث مهم و حساس است و شما باید خودتان را با تمرینات بسیار آماده کنید.

مترجم: محمد غضنفری
(MAMAD G.H.A)
Mamad5959@yahoo.com



منبع:
مجله رسمی موتور AKCNEX
Acknex User Magazine (aum)
<http://aum.conitec.net>

و برخی اطلاعاتی که از قبل داشتم.

هرگونه کپی برداری از این اثر تنها با ذکر منبع و نام ناشر و مترجم مجاز می باشد.

ناشر:
<http://futureworld.ir>