

## خلاصه‌ی جزوه‌ی درسی برنامه‌نویسی به زبان پایتون

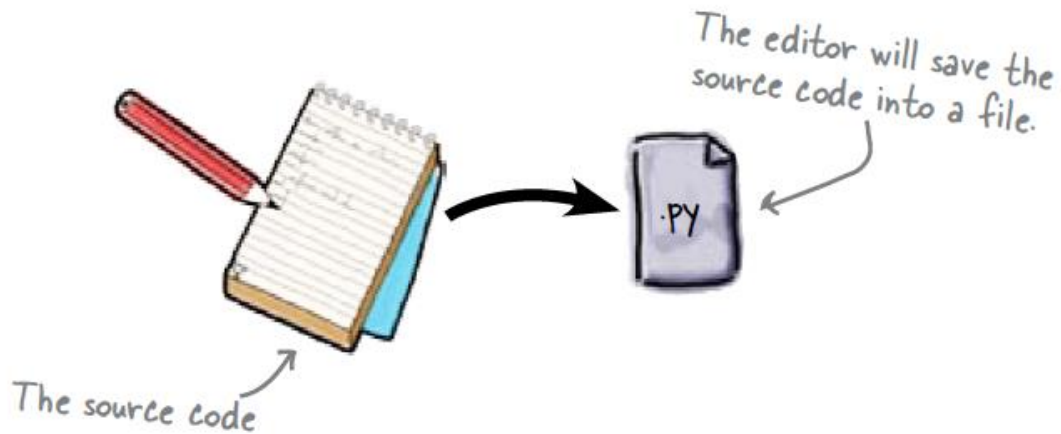
در این جزوه سعی شده است مباحث مهمی که در کلاس درس مطرح می‌شود به صورت خلاصه بیان شود. برخی از موارد به صورت سؤال رها شده است ولی با توجه به اینکه این موارد در کلاس مطرح شده است، از سؤال به راحتی می‌توانید به پاسخ دست یابید ... ممکن است برخی از استفاده‌های دیگر موارد توضیح داده شده اشاره نشده باشد. توجه کنید که ما انتظار داریم که شما تمامی مواردی که در این جزوه اشاره شده است و سؤالات آن را یاد بگیرید و موارد اضافه امتیازی است ...

۱. چرا برنامه‌نویسی؟

۲. چرا برنامه‌نویسی به زبان پایتون؟

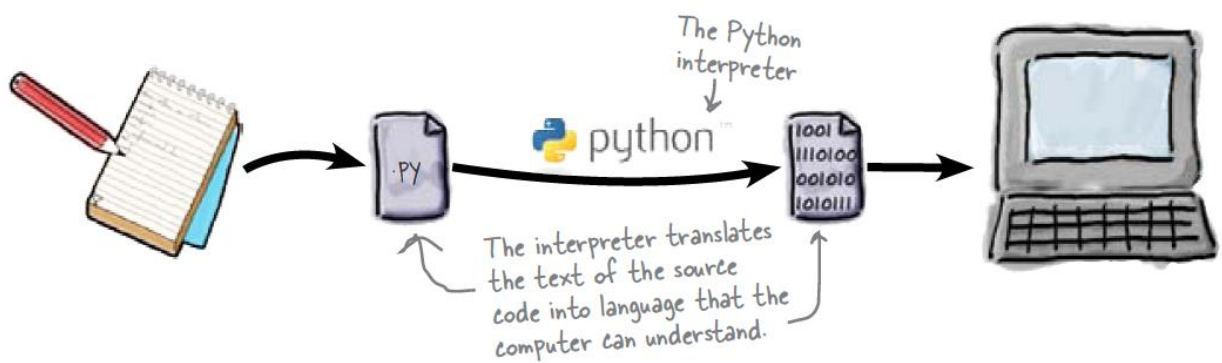
۳. ویرایشگر یا ویراستار:

برای برنامه‌نویسی به هر زبانی یک ویرایشگر برای برنامه‌نویسی لازم است. ویرایشگر یا همان editor برای نوشتن برنامه و همچنین ذخیره‌ی آن در کامپیوتر مورد استفاده قرار می‌گیرد. ویرایشگر مورد استفاده در درس برنامه‌نویسی به زبان پایتون از ویرایشگر notepad++ استفاده می‌کنیم.



۴. مفسر (تبدیل کننده‌ی برنامه به زبان ماشین):

برنامه‌ی نوشته شده توسط شما در ویرایشگر برای کامپیوتر قابل فهم نیست و باید تبدیل به زبان کامپیوتر شود. زبان کامپیوتر زبان باینری یا همان 01 (صفر و یک) است. برای تبدیل برنامه‌ی نوشته شده در ویرایشگر به زبان کامپیوتر نیازمند یک مفسر هستیم. در واقع زبان برنامه‌نویسی انتخابی توسط ما برای ما یک مفسر در اختیار می‌گذارد که می‌توانیم از آن برای تبدیل برنامه‌ی خود به زبان ماشین یا همان زبان صفر و یک استفاده کنیم.



## ۵. تعریف متغیر:

متغیر در زبان برنامه‌نویسی مانند یک کشو می‌ماند، همانطور که شما در کشور وسایل خود را نگهداری می‌کنید، در برنامه‌نویسی با استفاده از متغیر می‌توانید اعداد، حروف، کلمات و سایر مواردی که لازم دارید را ذخیره کنید. با استفاده از متغیرها می‌توانید عملیات ریاضی و تعداد دفعات تکرار و بسیار موارد دیگری را در برنامه‌نویسی انجام دهید. در واقع متغیر جزو یکی از ارکان اصلی برنامه‌نویسی است.

در پایتون برخلاف برخی زبان‌های برنامه‌نویسی لازم نیست نوع متغیر را تعریف کنیم. تنها کافی است متغیر را مقداردهی کنید. همان‌طور که در جدول زیر قابل مشاهده است. برای مثال وقتی می‌نویسیم  $a = 10$  یعنی که متغیر  $a$  از نوع عدد صحیح تعریف شده است و مقدار آن برابر ۱۰ قرار گرفته است.

ردیف	نوع داده‌ای	نوع داده‌ای	نحوه‌ی تعریف	تابع تبدیل
۱	integer	عدد صحیح	$a = 10$ $b = 12$	int()
۲	Float	عدد اعشاری	$a = 12.4$ $b = 23.6$	float()
۳	string	رشته	$S = \text{"salam"}$	str()
۴	list	لیست	$a = [1,2,3,4]$ $b = [\text{"salam"}, \text{"hi"}, \text{"bye"}]$	

	$c =$ $[1,2,"salam",13.5]$  $d = \text{range}(1,5)$			
--	--	--	--	--

با تعریف یک متغیر یک خانه از حافظه به آن متغیر اختصاص داده خواهد شد و با تغییر متغیر عدد یا حرف یا کلمه‌ی ذخیره‌شده در آن خانه از حافظه نیز تغییر می‌کند. برای مثال برنامه‌ی زیر را در نظر بگیرید.

$$A = 10$$

$$B = 12$$

$$A = B + 4$$

$$B = A + 14$$

توضیح برنامه‌ی بالا به این صورت است، ابتدا متغیر  $A$  از نوع عدد صحیح تعریف شده است و مقدار آن برابر ۱۰ قرار گرفته است. یعنی یک خانه از حافظه به متغیر  $A$  اختصاص داده شده است و عدد ۱۰ در آن خانه ذخیره شده است. در خط دوم متغیر  $B$  از نوع عدد صحیح تعریف شده است و مقدار آن برابر ۱۲ در نظر گرفته شده است. همانند متغیر  $A$  متغیر  $B$  نیز یک خانه‌ی جداگانه از حافظه را به خود اختصاص داده است و عدد ۱۲ در آن خانه ذخیره شده است. در خط سوم از برنامه نوشته شده است  $A = B + 4$  وقتی با این خط روبرو می‌شویم به این معنی است که باید ابتدا طرف راست مساوی را حساب کنیم و حاصل آن را در متغیر  $A$  قرار دهیم. در این حالت حاصل سمت راست مساوی برابر  $12 + 4$  یا همان ۱۶ است. پس در نهایت این خط به این معنی است که متغیر  $A$  را برابر ۱۶ کرده‌ایم و عدد ۱۰ از خانه‌ی حافظه‌ی  $A$  حذف می‌شود و عدد ۱۶ به جای آن قرار خواهد گرفت. خط چهارم برنامه نوشته شده است  $B = A + 14$  این خط نیز مانند خط سوم باید ابتدا طرف راست را محاسبه کنیم و بعد حاصل را در متغیر  $B$  قرار دهیم. در خانه‌ی  $A$  عدد ۱۶ بود و جمع آن با ۱۴ برابر ۳۰ خواهد شد و در نهایت در متغیر  $B$  عدد ۳۰ را خواهیم داشت. اگر بعد از این دستورات متغیر  $A$  را پرینت کنیم با ۱۶ روبرو خواهیم شد و اگر  $B$  را چاپ کنیم ۳۰ خواهد بود.

## ۶. تعریف تابع:

تابع مانند یک جعبه‌ای است که ورودی دریافت می‌کند و یکسری عملیات بر روی ورودی‌ها انجام می‌دهند و در نهایت خروجی مشخصی دارد. ورودی و خروجی تابع در تابع تعریف می‌شود.



در پایتون یکسری توابع از پیش تعریف شده در اختیار داریم. برای مثال تابع `print` یا تابع `input` از ابتدا در پایتون تعریف شده است. امکان تعریف توابع جدید نیز در پایتون وجود دارد که در جلسات آینده به آن اشاره خواهد شد.

برای مثال برای درست کردن آب دوغ خیار، باید آب، ماست، خیار و سایر موارد لازم را استفاده کنیم تا به آب دوغ خیار برسیم. پس ورودی‌های تابع این غذای لذیذ تابستانی آب، ماست، خیار، کشمش، گردو و سبزی خواهد بود. بعد از انجام عملیات آشپزی بسیار دشوار به غذای لذیذ آب دوغ خیار خواهیم رسید که خروجی تابع است. در واقع در این مثال تابع ما عملیات درست کردن آب دوغ خیار بود.

## ۷. تعریف تابع print:

تابع print برای نمایش یک عبارت، عدد یا متغیر در صفحه‌ی خروجی استفاده می‌شود.

```
print ("salam")
```

```
print (12)
```

```
a = 16
```

```
print (a)
```

همان‌طور که در مثال‌های بالا مشخص است، در اولین خط عبارت salam در خروجی چاپ می‌شود. در خط دوم عدد ۱۲ چاپ می‌شود. و در خط چهارم برنامه آنچه در متغیر a ذخیره شده است را چاپ می‌کند که در واقع عدد ۱۶ است.

## ۸. تعریف تابع input():

تابع input برای گرفتن عدد، حرف یا کلمه از کاربر مورد استفاده قرار می‌گیرد. برای مثال اگر بخواهیم یک عدد از کاربر دریافت کنیم باید همانند مثال زیر عمل کنیم.

```
A = int(input('insert number: '))
```

در واقع در این خط از برنامه از تابع int() نیز استفاده کردیم. تابع input() به صورت پیش فرض خروجی رشته (string) برمی‌گرداند. برای اینکه بخواهیم آنچه کاربر وارد کرده است را به عدد تبدیل کنیم باید از این تابع استفاده کنیم.

## ۹. تعریف تابع int():

برای تبدیل ورودی به عدد صحیح مورد استفاده قرار می‌گیرد.

## ۱۰. تعریف تابع float():

برای تبدیل ورودی به عدد اعشاری مورد استفاده قرار می‌گیرد.

## ۱۱. تعریف تابع str():

برای تبدیل ورودی به رشته‌ی حروف (کلمه) مورد استفاده قرار می‌گیرد.

## ۱۲. تعریف الگوریتم:

مفهوم الگوریتم را معمولاً با تشبیه به دستور آشپزی توضیح می‌دهند. مثلاً اگر بخواهیم آبگوشت درست کنیم (عمل موردنظر) با فرض اینکه مواد خام را داریم (حالت اولیه) مراحل مشخصی را باید طبق دستور آشپزی طی کنیم (دستورالعمل‌ها) تا به آبگوشت آماده (حالت پایانی) برسیم. البته الگوریتم‌ها معمولاً پیچیده‌تر از این هستند.

مثال دیگری که از الگوریتم و برای مشخص کردن میزان فهم کامپیوتر استفاده می‌شود مثال باز کردن درب شیشه مربا است. فرض کنید شما بخواهید درب شیشه‌ی مربا را باز کنید. برای این کار چه مرحله‌ی را انجام می‌دهید؟ شاید بگویید که خب درب شیشه‌ی مربا را باز می‌کنیم! ولی خیر این‌طوری نیست!! شما برای کامپیوتر ابتدا باید دست چپ و راست را تعریف کنید. پس از آن باید شیشه‌ی مربا را تعریف کنید و بعد باید مشخص کنید که شیشه‌ی مربا از دو جزء شیشه و درب تشکیل شده است. بعد باید دست چپ را بالا ببرید و به سمت شیشه حرکت دهید و با دست چپ شیشه را محکم گرفته و بعد دست راست را بالا ببرید و درب مربا را گرفته و بعد به جهت باز شدن درب بگردانید و تا وقتی که باز شود. این مثال نشان می‌دهد شما برای نوشتن برنامه باید تک‌تک عناصر یا المان‌ها را برای کامپیوتر ابتدا تعریف کنید و بعد گام‌به‌گام راه‌حل خود برای رسیدن به نتیجه‌ی برنامه را توضیح دهید.

مثال الگوریتم جمع کردن دو عدد:

۱. شروع
۲. عدد اول را دریافت کن  $a =$
۳. عدد دوم را دریافت کن  $b =$
۴. عدد اول و دوم را جمع کن  $a + b =$
۵. حاصل را چاپ کن  $\text{print}(a+b)$
۶. پایان

مثال الگوریتم محاسبه‌ی میانگین دو عدد؟

الگوریتم محاسبه‌ی میانگین  $n$  عدد؟

### ۱۳. برنامه‌ی حدس عدد (بخش اول دستور شرطی):

در این بخش می‌خواهیم یک متغیر تعریف کنیم و عددی را در آن قرار دهیم. برای مثال متغیر  $A = 5$  را تعریف کرده و از کاربر بخواهیم آن عدد را حدس بزند و اگر کاربر درست حدس زد برنده‌شده است و اگر اشتباه حدس بزند بازنده شده است. برای اینکه بفهمیم عددی که کاربر وارد کرده است برابر با عدد متغیری که ما تعریف کرده‌ایم هست یا خیر باید از دستور `if` یا شرط استفاده کنیم. دستوری شرط به صورت زیر قابل استفاده خواهد بود.

```
A = 5
```

```
B = int(input("insert number = "))
```

```
if (A == B):
```

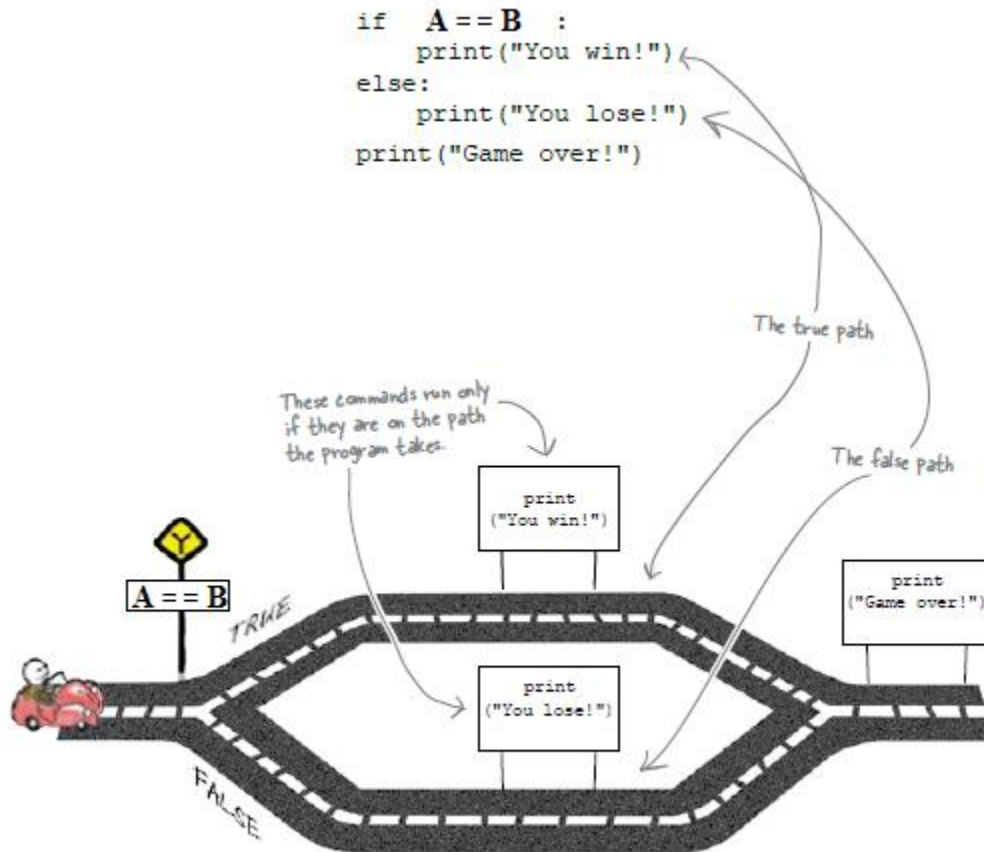
```
    print("you win")
```

```
else:
```

```
    print("you lose")
```

```
print("Game over")
```

برنامه‌ای که در بالا نوشته شده است همان برنامه‌ی حدس عدد اولیه‌ی هست که مورد نظر ما بود. دستور `if` در برنامه‌نویسی مانند یک دوراهی است. شکل برنامه‌ی بالا در زیر قابل مشاهده است.



## ۱۴. برنامه‌ی حدس عدد (بخش دوم کتابخانه):

در این بخش می‌خواهیم عددی که کاربر باید حدس بزند را به‌صورت تصادفی تعریف کنیم. در زبان‌های برنامه‌نویسی به‌صورت پیش‌فرض کتابخانه‌هایی توسط توسعه‌دهندگان زبان برنامه‌نویسی ایجاد شده است که می‌توانیم از این کتابخانه‌ها برای نوشتن برنامه‌ی خود استفاده کنیم. یکی از این کتابخانه‌ها `random` هست. با استفاده از دستور `import` می‌توانیم این کتابخانه و توابع داخل آن را به برنامه‌ی خود اضافه کنیم. این کتابخانه دارای یک تابع به نام `randint` هست که عدد صحیح تصادفی بین دو بازه‌ای که در تابع تعریف می‌کنیم را بازمی‌گرداند. در برنامه‌ی با استفاده کتابخانه‌ی `random` و تابع `randint` توانستیم کاری کنیم که عددی که کاربر باید حدس بزند به‌صورت تصادفی تعریف شود.

```
import random

A = random.randint(1,10)

B = int(input("insert number = "))

if ( A == B):

    print("you win")

else:

    print("you lose")

print("Game over")
```

## ۱۵. برنامه‌ی حدس عدد (حلقه‌ی تکرار):

در این بخش می‌خواهیم کاربر امکان حدس بیش از یک‌بار عدد را داشته باشد. برای این مورد باید از حلقه‌ی `while` استفاده کنیم. حلقه این امکان را می‌دهد که بتوانیم یک قسمت از برنامه را تکرار کنیم.

قبل از اینکه به برنامه‌ی حدس عدد بپردازیم یک مثال از حلقه را بررسی خواهیم کرد.

```
answer = "no"
while answer == "no":
    answer = input("Are we there? ")
    print("We're there!")
```

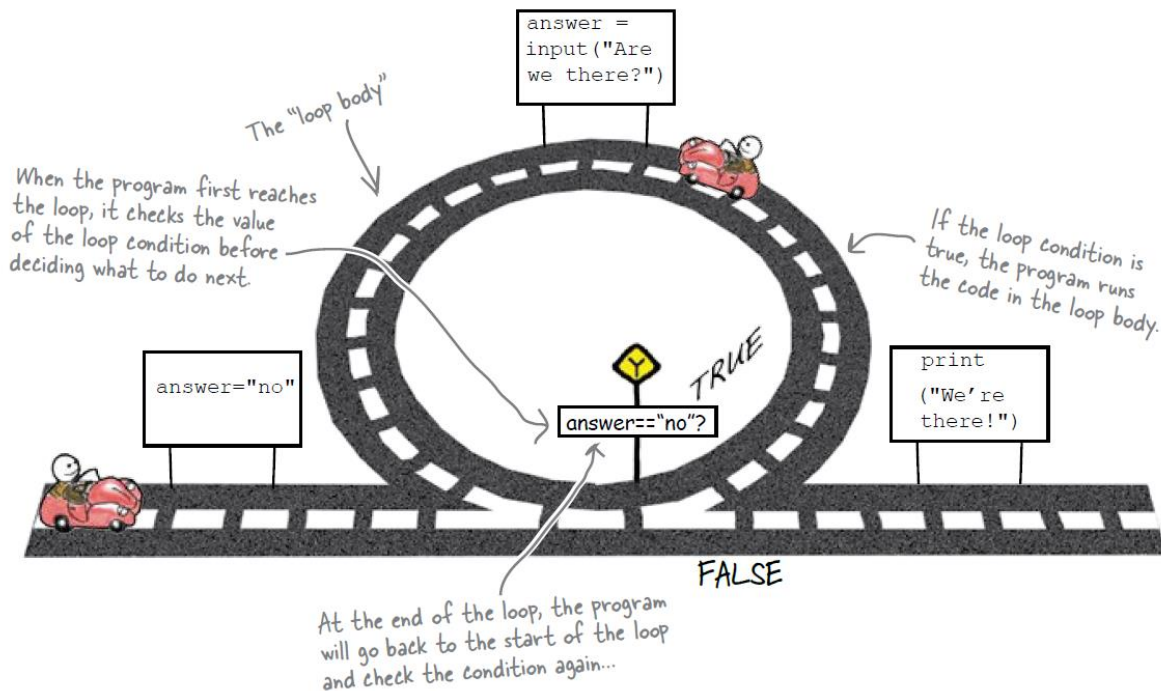
We want to make sure the loop runs the first time.

The loop condition

The loop body is the indented code following the "while" line.

The loop body is just one line of code in this example, but the loop body can be many lines of code. It might even include branches and other loops.

برنامه‌ای که در عکس بالا آمده است با استفاده از حلقه‌ی `while` نوشته شده است. همان‌طور که بارها در کلاس اشاره شد. حلقه مانند یک میدان می‌ماند که برای ورود به میدان یک شرط داریم و اگر شرط برقرار باشد وارد میدان می‌شویم و تا زمانی که شرط برقرار باشد از میدان خارج نخواهیم شد. این مثال تا وقتی که کاربر پاسخ `no` را بنویسد، ادامه پیدا خواهد کرد تا به مقصد برسد و بعد از آن هر ورودی دیگری به جز `no` را وارد کند از میدان یا همان حلقه خارج خواهد شد.





مثال چاپ اعداد ۱ تا ۱۰:

```
A = 10
while(A > 0):
    print(A)
    A = A - 1
Print("Game Over")
```

در برنامه‌ای که کد آن در بالا نوشته شده است. اعداد ۱ تا ۱۰ در خروجی چاپ می‌شود.

مثال جمع اعداد ۱ تا n:

برای جمع اعداد ۱ تا n باید از یک متغیر کمکی برای جمع اعداد استفاده کنیم. همچنین باید عدد n را از کاربر دریافت کنیم.

```
N = int(input())
S = 0
while(N > 0):
    S = S + N
    N = N - 1
print(S)
print("Game Over")
```

حال می‌رسیم به برنامه‌ی حدس عدد. برای امکان تکرار حدس باید به‌صورت زیر برنامه را بنویسیم. حالت‌های دیگر این برنامه نیز در کلاس مطرح شد ...

```
import random
A = random.randint(1,10)
B = -1
While(A != B):
    B = int(input("insert number = "))
    if ( A == B):
        print("you win")
    else:
        print("you lose")
print("Game over")
```

## ۱۶. حلقه‌ی تکرار for:

حلقه‌ی for همانند حلقه‌ی while هست با این تفاوت که کاربردهای متفاوتی دارد و نوع نوشتار آن متفاوت است. یکی از کاربردهای حلقه‌ی for در کار با لیست هست.

مثال چاپ اعداد ۱ تا ۱۰ با حلقه‌ی for:

```
for i in range(1,10):  
    print(i)
```

مثال جمع اعداد ۱ تا ۱۰ با حلقه‌ی for:

```
S = 0  
for i in range(1,10):  
    S = S + i  
print(S)
```