

به نام یزدان پاک

درسنامه

کار و فناوری

پودمان برنامه نویسی

بر اساس صفحه بندی کتاب هفتم

نام دبیر: علیرضا اربطی

آموزش و پرورش ناحیه ۲ زنجان

دبیرستان دوره اول: شهید بهشتی ۲

سال تحصیلی ۱۴۰۱-۱۴۰۲

سلام دانش آموزان عزیز

فرض کنید یک نفر هست که اصلا چایی درست کردن را بلد نیست...

لطفاً طریقه درست کردن چایی را با زدن شماره مرحله مرحله بنویسید، طوری که اگر او از روی دستور العمل شما جای درست کند یک چایی خوب و خوش رنگ و قابل نوشیدن باشد.

لطفاً شماره ۱ را شروع و آخرین شماره را پایان بنویسید.

بچه های خوب چیزی رو که شما نوشتید اسمش هست "الگوریتم"

میتوانیم مثال های بیشتری از کارهایی که بصورت روزانه انجام میدهیم به این روش بیان کنیم، که همگی اینها در واقع یک روش حل مسئله یا الگوریتم خواهد بود.

الگوریتم یعنی بیان کردن مراحل انجام یک کار و یا حل گام به گام یک مساله به صورت دقیق و با جزئیات، طوری که شروع و پایان هم داشته باشد.

الگوریتم باید هم خلاصه باشد و هم تمام نکات مهم داخل آن باشد.

الگوریتم یک واژه یونانی است و اصل کلمه از نام دانشمند ایرانی «الخوارزمی» گرفته شده است. چون برای اولین بار ایشان این روش حل مساله را بکار برده است.

الگوریتم در واقع مبنا و ریشه اصلی برنامه نویسی است که به روشهای مختلف بیان میشود. در پایه نهم شما با این موضوع بیشتر آشنا خواهید شد.

حالا ببینیم برنامه نویسی چیست؟

برنامه نویسی در واقع دستورالعمل هایی جزء به جزء است که ما به رایانه میدهیم و می گوییم که چه کاری را انجام بدهد.

برای اینکه بتوانیم این دستورات را به رایانه بدهیم ، باید زبان رایانه را بلد باشیم، زبان رایانه زبان باینری (صفر و یک) است. برای اینکه با این زبان بتوانیم با رایانه صحبت کنیم باید **زبان های**



برنامه نویسی را بیاموزیم و به کار ببریم.

زبان های برنامه نویسی در واقع زبان ما را

به زبان کامپیوتر نزدیکتر میکنند.

زبان های برنامه نویسی انواع مختلفی دارد که شاید شما با برخی از آن ها آشنایی داشته



باشید. برخی از آنها عبارتند از:

۱- زبان برنامه نویسی جاوا

۲- زبان برنامه نویسی C

۳- زبان برنامه نویسی C++

۴- زبان برنامه نویسی C#

۵- زبان برنامه نویسی Objective-C

۶- زبان برنامه نویسی PHP

۷- زبان برنامه نویسی ASP.NET

۸- زبان HTML

۹- زبان برنامه نویسی Python

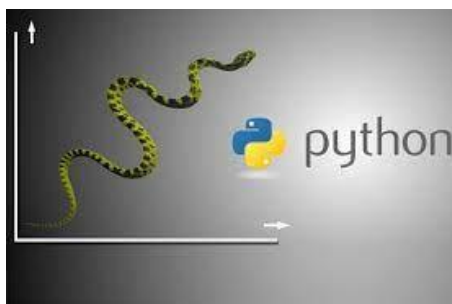
۱۰- زبان برنامه نویسی Ruby

۱۱- زبان JavaScript

۱۲- زبان SQL



که هر یک از این زبانها با ویژگی‌های مختلفی که دارند برای اهداف خاصی به کار برده میشوند.



چرا پایتون؟

پایتون، مار محبوب دنیای برنامه نویسی

لازم نیست از اسم پایتون بترسید چون این یکی نیشتان نمی‌زند. در دنیای برنامه نویسی،

۱- پایتون یک زبان چندمنظوره (یعنی می‌توانید با پایتون برای تمام سیستم عامل‌ها نرم افزار بنویسید)



۲- متن باز (می‌توانید آن را تغییر دهید و نسخه‌ی بهتری ارائه دهید)

۳- یک زبان سطح بالا و در عین حال ساده است.

۴- دارای کتابخانه‌های کاربردی و فراوان است. (در صفحات بعد در مورد کتابخانه توضیح داده می‌شود).

به این دلایل برای خیلی از تازه واردهای دنیای برنامه نویسی پایتون زبان جذابی است. چون با یادگیری این زبان خیلی زود می‌توانند کدنویسی را شروع کنند. شاید پیرسید: زبان سطح بالا یعنی چه؟ هرچه یک زبان برنامه نویسی به زبان ماشین نزدیک‌تر باشد، در این صورت یک زبان سطح پایین محسوب می‌شود، اما زبان‌های برنامه‌نویسی سطح بالا به زبان انسان نزدیک‌تر و در نتیجه قابل فهم‌تر هستند. بنابراین یادگیری زبان‌های برنامه‌نویسی سطح پایین به مراتب سخت‌تر است.

نحوه نصب برنامه پایتون (python)

در کتاب درسی بطور کامل توضیح داده شده است. ولی جالب است بدانید که یک نسخه پایتون برای کار در گوشی (اندروید) بنام **pydroid 3** نیز وجود دارد که می‌توانید آنرا از اینترنت دانلود و نصب نمایید.

The image is a screenshot of the Google Play Store page for the app "Pydroid 3 - IDE for Python 3". The app icon shows "Py3" in white on a blue background. The app name is "Pydroid 3 - IDE for Python 3". It is categorized under "Education" and has a rating of 4.167 stars. The page also indicates that the app contains ads and offers in-app purchases, and is compatible with all devices. There is an "Install" button and an "Add to wishlist" button.

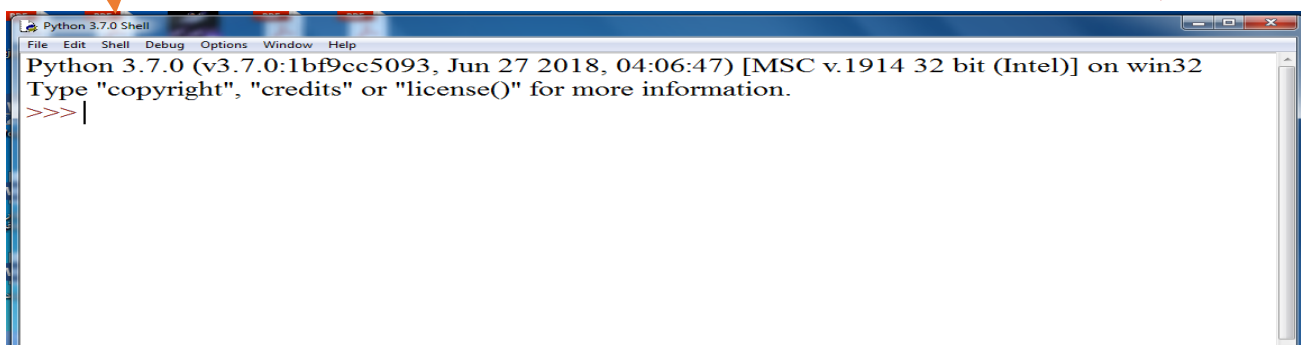
برای **کدنویسی پروژه‌های کوچک** با پایتون، می‌توان از محیط **شل (Shell)** استفاده کرد. اما اگر بخواهیم روی پروژه‌های بزرگ‌تر کار کنیم، استفاده از یک **ویرایشگر کد اختصاصی** یا محیط توسعه یکپارچه (**IDE | Integrated Development Environment**) ضرورت دارد.

IDLE یک محیط توسعه یکپارچه است که به همراه پایتون به کاربران ارائه می‌شود. ولی IDE های دیگری نیز برای نوشتن و اجرای کدهای پایتون وجود دارد که برخی از آنها عبارتند از:

Atom , Pycharm , Sublime Text , Visual Studio Cod , Jupyter Notebook, AWS Cloud9

استفاده از یک محیط برنامه نویسی مناسب برای اجرای کدهای پایتون بسیار اهمیت دارد. با وجود اینکه استفاده از ابزارهای تخصصی در پروژه‌های ساده ضرورتی ندارد، اما پس از عبور از برنامه‌های ساده، وجود یک **IDE** الزامی است. **IDE** های متعددی برای **برنامه نویسان** ارائه شده است که می‌توان گفت برخی از این **IDE** ها، برای یک زبان برنامه نویسی خاص مناسب‌تر از بقیه آن‌ها هستند. تعدادی از **IDE** هایی که به صورت رایگان ارائه شده‌اند، دارای محدودیت‌هایی هستند که در چنین شرایطی می‌توان با **پرداخت هزینه** به امکانات کامل این محیط‌های برنامه نویسی دسترسی داشت. یکی از ویژگی‌های محیط‌های برنامه نویسی پایتون، امکان اجرا و ویرایش کدهاست. حداقل نیازمندی برای یک محیط مناسب برنامه نویسی پایتون این است که قابلیت‌هایی مثل (هایلایت کردن سینتکس) (**Syntax Highlighting**) ، **Code Folding** ، (امکان مخفی‌سازی یا نمایش بخشی از کد)، «تطبیق پرانتز (**Bracket Matching**)» و سایر موارد در آن تعبیه شده باشد. ممکن است محیط برنامه نویسی پایتون ویژگی‌های پیشرفته‌تری همچون پیشنهاد کد، اشکال‌زدایی و قابلیت اضافه کردن مخزن‌های آنلاین به پروژه را نیز شامل شود. با بررسی محیط‌های برنامه نویسی مختلف و همچنین میزان استفاده کاربران از آن‌ها، می‌توان **Visual Studio Code ، Pycharm و AWS Cloud9** را به عنوان کاربردی‌ترین محیط‌های برنامه نویسی پایتون انتخاب کرد.

کار با پایتون ؛ با باز کردن فایل نصب شده ابتدا محیط **shell window** ظاهر میشود.



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

این محیط برای اجرا و مشاهده دستورات به صورت خط به خط است. نمونه نوشتن و اجرای دستور را در شکل زیر ملاحظه می کنید.

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 5
5
>>> 6
6
>>> print('سلام')
سلام
>>> |
```

دستور `print()` برای چاپ خروجی استفاده میشود.

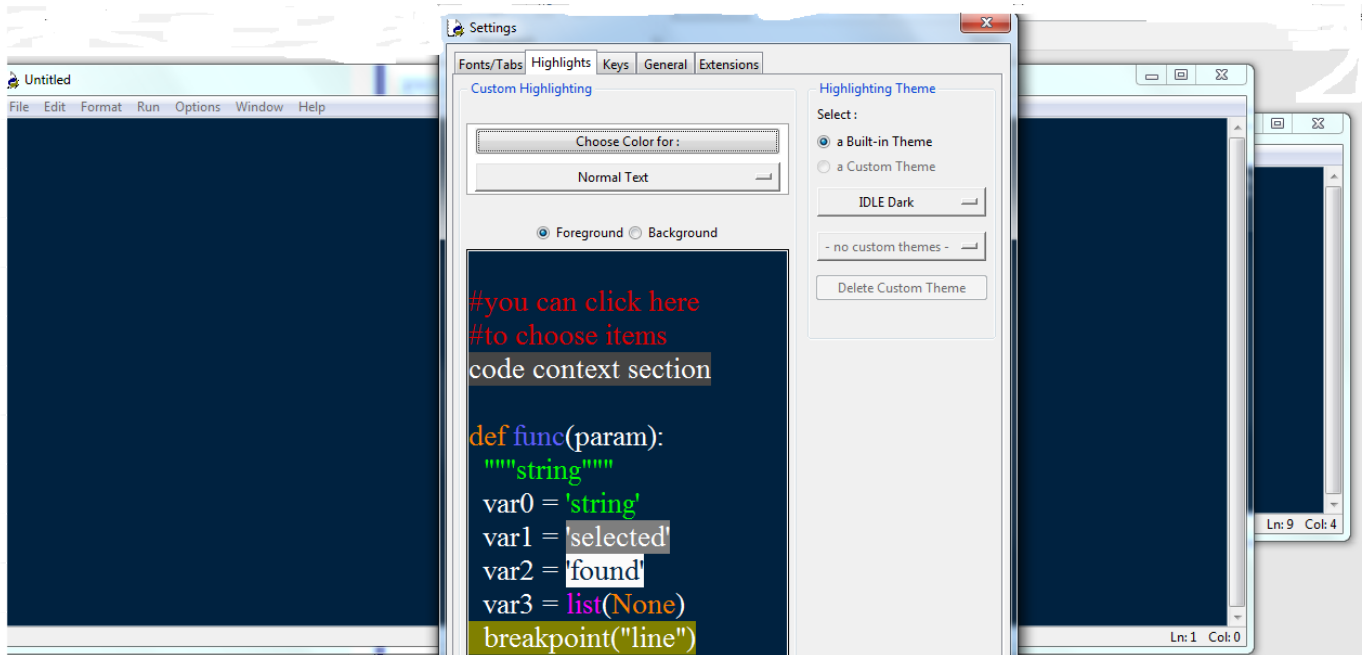
برای رفتن به محیط ویرایشگر متن (IDE) از منوی `file` گزینه `New file` را انتخاب می کنیم. در این محیط میتوانیم دستورات را نوشته، ویرایش، ذخیره و اجرا کنیم.

لازم به ذکر است که از گزینه `Options` میتوانیم مواردی مانند `Font` نوشته ها، `Tem` صفحه و ...



را تغییر دهیم. مثال: مسیر تغییر رنگ صفحه

`Options` → `Configure IDLE` → `Highlights` → `IDLE Dark` → `Apply/Ok`

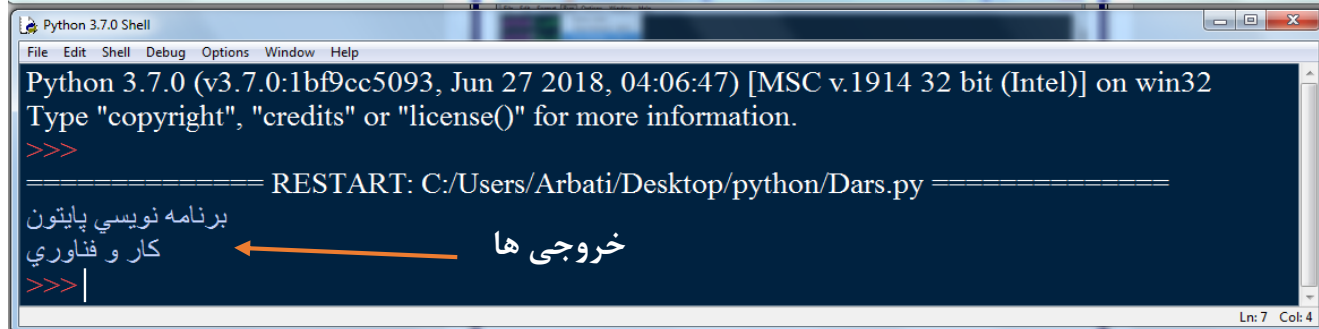
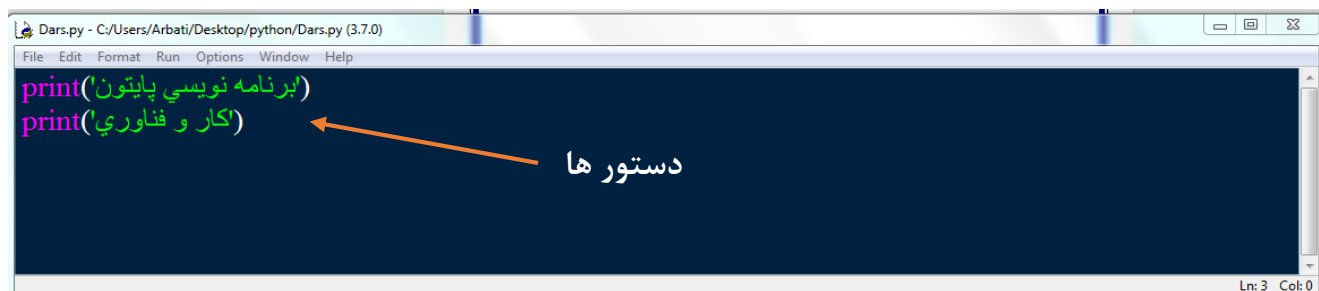
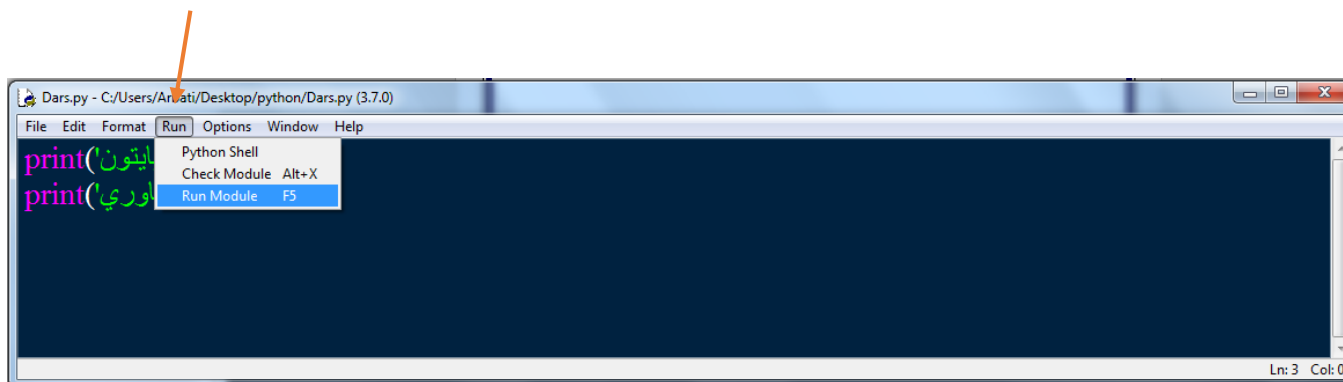


نوشتن و اجرای دستور در محیط IDLE

پس از نوشتن دستور از گزینه Run و از مسیر زیر آن را Save و اجرا میکنیم.

F5 یا زدن کلید Run → Run Module F5

نکته: پسوند فایل های ذخیره شده با پایتون .py است.

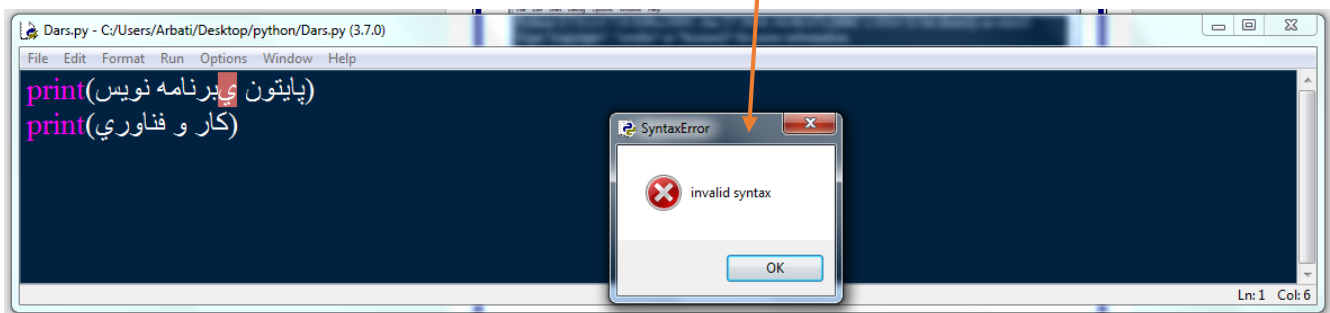


توجه: برای باز کردن و ویرایش کردن فایل های ذخیره شده از منوی File گزینه Open و سپس از مسیر ذخیره شده فایل مورد نظر را انتخاب می کنیم.

در نوشتن دستورات متنی حتما باید متن یا نوشته داخل پرانتز و همچنین در دو طرف آن ' کوتیشن ' یا " دبل کوتیشن " قرار گیرد .

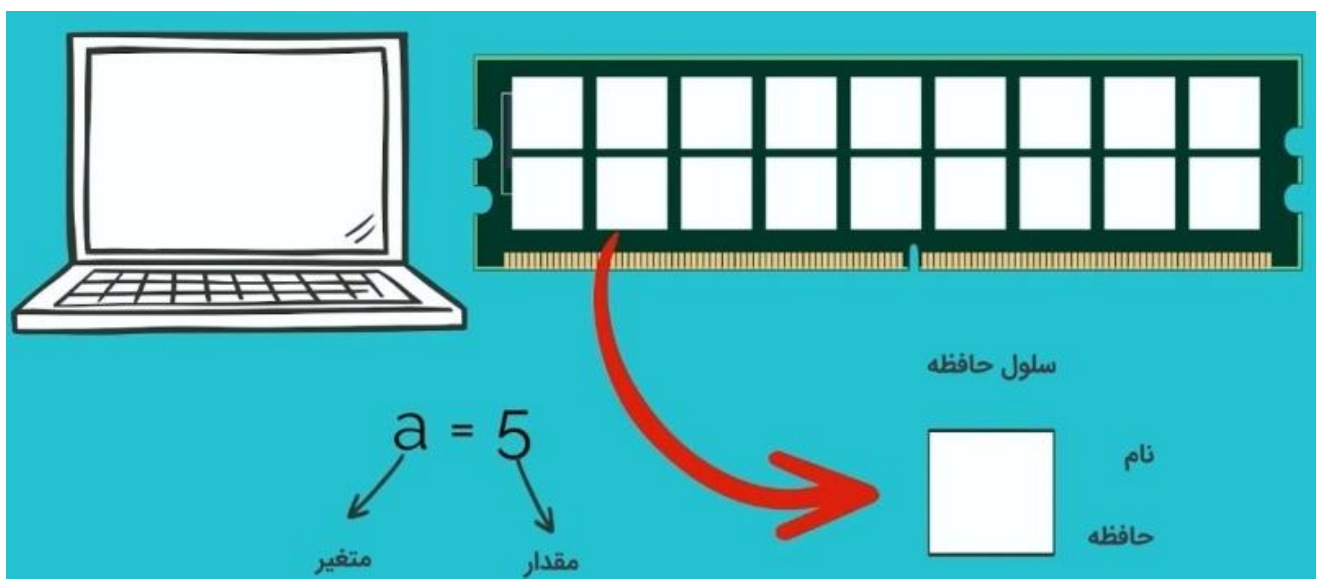
```
*Dars.py - C:/Users/Arbati/Desktop/python/Dars.py (3.7.0)*
File Edit Format Run Options Window Help
print('ایرنامه نویسی پایتون')
print('کار و فناوری')
```

وگرنه خطا خواهد داد و اجرا نخواهد شد.



متغیرها

حافظه رایانه از تعدادی خانه حافظه تشکیل یافته است. متغیر (variable) در پایتون را میتوانیم همانند یک محفظه در نظر بگیریم که برای ذخیره موقت اطلاعات استفاده میشود.



ساخت متغیر در پایتون

ساخت یک متغیر جدید با علامت تساوی (=) انجام می شود. به مثال زیر دقت کنید:

```
>>> n = 300
```

در مثال بالا، مقدار عددی 300 را در متغیری به نام n ذخیره کردیم. حالا می‌توانید از این متغیر در برنامه استفاده کنید:

```
>>> print(n)
300
```

در زمان مقداردهی متغیرها، پایتون به طور اتوماتیک نوع متغیر را مشخص میکند. مثلا در کد بالا، مفسر پایتون میتواند تشخیص دهد که یک عدد را در متغیر n ذخیره کردیم. برای تشخیص نوع متغیر می‌توانید از تابع type استفاده کنید. مثلا در کد زیر، نوع متغیر عددی int است:

```
>>> n = 300
>>> type(n)
<class 'int'>
```

با اگر متغیر با نوع رشته ذخیره کنید، مفسر پایتون نوع متغیر را str مشخص میکند:

```
>>> m = 'amir'
>>> type(m)
<class 'str'>
```

تغییر مقدار متغیر در پایتون

در پایتون می‌توانید نوع یک متغیر را تغییر دهید. مثلا اگر مقدار ذخیره شده در یک متغیر عدد باشد، می‌توانید آن را به انواع دیگر مانند رشته و ... تغییر دهید.

همانطور که در مثال زیر می بینید، مقدار متغیر n ابتدا عدد بود اما توانستیم آن را به نوع رشته (متن) تغییر دهیم.

به کد زیر دقت کنید:

```
>>> n = 300
>>> n = 'amir'
>>> n
amir
```

- موارد زیر **قوانین نامگذاری متغیرها** در پایتون هستند که باید آنها را رعایت کنیم ،
- ۱- نام ها به حروف کوچک و بزرگ حساس هستند. برای مثال دو نام **Age** و **age** با هم متفاوتند.
 - ۲- نام ها می توانند شامل حروف کوچک و بزرگ، همچنین اعداد و زیر خط باشند.
[**A-z** , **0-9** , **_**]
 - ۳- نام ها نمی توانند شامل کاراکترهای خاص مانند **!@#\$%^&*** باشند.
 - ۴- نام ها نمی توانند با عدد شروع شده و یا دارای فاصله باشند.
 - ۵- استفاده از کلمات کلیدی زبان پایتون در نامگذاری متغیر مجاز نیست مانند **from** و **for** و **int** :
... و
 - ۶- نام متغیر ها نمی تواند به **_** (زیر خط) ختم شود.
 - ۷- نام متغیرها حتما باید با حروف الفبای انگلیسی و یا زیر خط شروع شود.

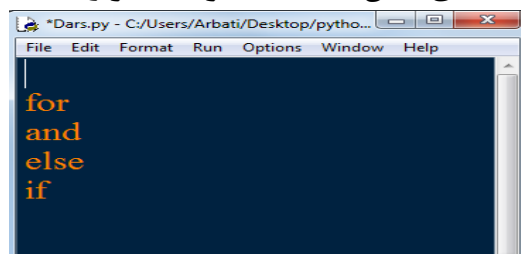
کلمات کلیدی پایتون

ejiga.com

تمامی کلمات کلیدی در پایتون

AND	FALSE	TRUE
AS	CLASS	DEF
ASSERT	IS	FROM
BREAK	FINALLY	NONLOCAL
NOT		RETURN
OR		TRY
PASS		WHILE
RAISE		WITH
DEL	IN	NONE
ELIF	IMPORT	CONTINUE
ELSE	IF	FOR
EXCEPT	GLOBAL	LAMBDA
	YIELD	

نکته: هنگام نوشتن دستور با کلمات کلیدی ، این کلمات معمولا تغییر رنگ داده و معمولا نارنجی



The screenshot shows a window titled '*Dars.py - C:/Users/Arbati/Desktop/pytho...' with a menu bar (File, Edit, Format, Run, Options, Window, Help). The code in the editor is:


```
for
and
else
if
```

 The words 'for', 'and', 'else', and 'if' are highlighted in orange, demonstrating the visual feedback provided by the Python interpreter for keywords.

رنگ میشوند. ←

پاسخ کار غیر کلاسی صفحه ۶۱ کتاب کار و فناوری پایه هفتم

```
File Edit Format Run Options Window Help
print("کار غیر کلاسی صفحه ۶۱ کتاب کار و فناوری هفتم")
print("-----")
print('          سوال ۱          ')
print('تعداد صفحات خوانده شده=',(100-10))
print('تعداد صفحه خوانده شده در هر ساعت=',(100-10)/6)
print("-----")
print('          سوال ۲          ')
print('یک ساعت و بیست دقیقه و سی ثانیه چند ثانیه؟',(60*60)+(20*60)+30)
print("-----")
print('          سوال ۳          ')
print('مساحت دایره=',(5*5)*3.14)
print('محیط دایره=',(10*3.14))
print("-----")
print('          سوال ۴          ')
print('قیمت کیف=',(5*3000)*2)
```

دریافت مقدار از ورودی

برای این کار از دستور `input()` استفاده می کنیم.

```
File Edit Format Run Options Window Help
a=input('طول را وارد کنید')
```

مثال: ←

توجه: مقادیر دریافتی توسط دستور `input()` بصورت رشته (متن) در داخل متغیر قرار میگیرد و برای دریافت مقدار عدد صحیح از دستور `int()` و برای عدد اعشاری از `float()` قبل از `input()` استفاده می کنیم.

```
*Untitled*
File Edit Format Run Options Window Help
a=float(input('طول را وارد کنید'))          #اعشاري
a=int(input('طول را وارد کنید'))           #صحيح
```

مثال:

پاسخ فعالیت گروهی صفحه ۶۱ کتاب کاروفناوری پایه هفتم

n=1400
N=1401
Print(n)

n=n+1400
Print(n)

n=1400
n=n+1
Print(n)

n=1400
N=1401
Print(N)

1400

2800

1401

1401 پاسخ:

توجه: گزینه اول و آخر در کتاب درسی، مشابه هم هستند ولی در اینجا دستور **print()** آنها تغییر کرده است.

در اینجا یادآوری چند نکته حائز اهمیت است:

- ۱- پایتون به حروف بزرگ و کوچک حساس است
- ۲- در پایتون مقدار آخرین متغیر مد نظر است یعنی اگر مثلاً به متغیر n ، ۱۰۰ تا عدد هم بدهیم، پایتون برای اجرای دستور آخرین عدد متغیر را در نظر می‌گیرد.

عملگرهای ریاضی

برای انجام محاسبات ریاضی در پایتون از عملگرهای زیر استفاده می‌شود:

مثال	شرح	عملگر
a+b	جمع	+
a-b	تفریق	-
a*b	ضرب	*
a/b	تقسیم	/
a//b	تقسیم صحیح	//
a%b	باقیمانده تقسیم	%
a**b	توان	**

اولویت	عملگر
۱	**
۲	%, //, /, *
۳	-, +

نکته: پرانتز اولویت انجام عملگرها را

تغییر می‌دهد.

مثال: مقایسه دستور int , float

```
62.py - C:/Users/Arbati/Desktop/python/62.py (3.7.0)
File Edit Format Run Options Window Help
a=float(input('طول را وارد کنید'))
print(a)

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Arbati/Desktop/python/62.py =====
طول را وارد کنید
5.0
>>>
```

همانطور که ملاحظه میکنید پس از اجرای دستور float عدد 5 با یک رقم اعشار یعنی 5.0 نمایش داده میشود، در صورتی که در دستور int عدد 5 بصورت عدد صحیح بدون اعشار نمایش داده میشود.

```
62.py - C:/Users/Arbati/Desktop/python/62.py (3.7.0)
File Edit Format Run Options Window Help
a=int(input('طول را وارد کنید'))
print(a)

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Arbati/Desktop/python/62.py =====
طول را وارد کنید
5
>>>
```

پاسخ کار غیر کلاسی صفحه ۶۲ کتاب کاروفناوری پایه هفتم

```
r=float(input('شعاع قاعده را وارد کنید'))
h=float(input('ارتفاع منبع را وارد کنید'))
print('حجم منبع آب=',(r*r*3.14)*h)
```

پاسخ سوال ۱:

```
F=float(input('مقدار نیرو را وارد کنید'))
W=float(input('مقدار کار انجام شده را وارد کنید'))
print('مقدار جابجایی یا d =',W/F)
```

پاسخ سوال ۲:

```
x=float(input('جرم وزنه به کیلوگرم وارد شود'))
print('طول فنر یا',(0.8*x)+10)
```

پاسخ سوال ۳:

عملگرهای مقایسه ای

برنامه نویسان به نماد های جدول زیر عملگرهای مقایسه ای میگویند. نتایج این عملگرها **True**(درست) و یا **False**(غلط) است. با عملکرد برخی از این موارد در صفحات بعد آشنا خواهید شد.

مثال	شرح	علامت
a == b	مساوی	==
a != b	نامساوی	!=
a < b	کوچک تر	<
a > b	بزرگ تر	>
a <= b	کوچک تر مساوی	<=
a >= b	بزرگ تر مساوی	>=

شرط

در برنامه نویسی گاهی نیاز است قسمتی از کدها در صورت برقراری شرطی اجرا شود. برای بررسی یک شرط از دستور **if** استفاده می کنیم.

بلاک کد: مجموعه کدهایی که در صورت برقراری شرط اجرا می شوند ، یک بلاک کد را تشکیل می دهند ، که معمولا به یک میزان تورفتگی (indentation) دارند. (یک Tab یا چهار Space)

```
*64.py - C:/Users/Arbati/Desktop/python/64.py (3.7.0)*
File Edit Format Run Options Window Help
print('سلام')
temp=8
if temp<10:
    print('هوا سرد است')
    print('لباس گرم بپوشید')
print('خدا نگهدار')
```

مثال: خروجی قطعه کد روبرو را حدس بزنید.

در این مثال درجه هوا است. در صورتی که درجه هوا از ۱۰ کمتر باشد، شرط اجرا می شود. و اگر بالاتر از ۱۰ باشد از دستور **if** رد شده و دستور آخر یا همان خدا نگهدار را اجرا میکند که در بلاک کد نیست.

```

64.py - C:/Users/Arbati/Desktop/python/64.py (3.7.0)
print('سلام')
temp=8
if temp<10:
    print('هوا سرد است')
    print('لباس گرم بپوشید')
    print('خدا نگهدار')

Python 3.7.0 Shell
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1916 64-bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Arbati/Desktop/python/64.py =====
سلام
هوا سرد است
لباس گرم بپوشید
خدا نگهدار
>>>
    
```

دستور

اجرای دستور

```

64.py - C:/Users/Arbati/Desktop/python/64.py (3.7.0)
print('سلام')
temp=12
if temp<10:
    print('هوا سرد است')
    print('لباس گرم بپوشید')
    print('خدا نگهدار')

Python 3.7.0 Shell
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1916 64-bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Arbati/Desktop/python/64.py =====
سلام
خدا نگهدار
>>>
    
```

پاسخ کار کلاسی صفحه ۶۴ کتاب کاروفناوری پایه هفتم (مشخص کردن زوج یا فرد بودن

عدد ورودی کاربر)

```

File Edit Format Run Options Window Help
p=int(input('عدد را وارد کنید'))

if p%2==0:
    print('عدد زوج است')
else:
    print('عدد فرد است')
    
```

در مواقعی که شرط فقط دارای دو

حالت باشد، از دستور **if-else**

استفاده میکنیم. در این حالت اگر

شرط دستور **if** برقرار باشد، بلاک کد

مربوط به **if** اجرا و در غیر اینصورت

بلاک کد مربوط به **else** اجرا می شود. (اگر = **if** , وگرنه = **else**)

تهیه و تنظیم: علیرضا اربتی - دبیر کار و فناوری ناحیه ۲ زنجان

پاسخ کار کلاسی پایین صفحه ۶۴ کتاب کار و فناوری هفتم

```
File Edit Format Run Options Window Help
print('سلام')
temp=int(input('مقدار دمای هوارا وارد کن'))
if temp<10:
    print('هوا سرد است')
    print('لباس گرم بپوشید')
else:
    print('خدا نگهدار')
```

```
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun
bit (Intel)] on win32
Type "copyright", "credits" or "license
>>>
===== RESTART: C:\Users\Arbatil\De
سلام
مقدار دمای هوارا وارد کن=8
هوا سرد است
لباس گرم بپوشید
>>>
```

وقتی دما کمتر از ۱۰ وارد می شود، دستور if اجرا می شود: ←

```
File Edit Format Run Options Window Help
print('سلام')
temp=int(input('مقدار دمای هوارا وارد کن'))
if temp<10:
    print('هوا سرد است')
    print('لباس گرم بپوشید')
else:
    print('خدا نگهدار')
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun
it (Intel)] on win32
Type "copyright", "credits" or "license
>>>
===== RESTART: C:\Users\Arbatil\D
سلام
مقدار دمای هوارا وارد کن=12
خدا نگهدار
>>>
```

وقتی دما بالاتر از ۱۰ وارد می شود، دستور else اجرا می شود: ←

پاسخ کار غیر کلاسی صفحه ۶۵ کتاب کار و فناوری هفتم

```
p=int(input('عدد اولی را وارد کن'))
q=int(input('عدد دومی را وارد کن'))
if p>q:
    print(p)
else:
    print(q)
```

«دریافت دو عدد از ورودی»
راه حل (۱)

```
p=int(input('عدد اولی را وارد کن'))
q=int(input('عدد دومی را وارد کن'))
print(max(p,q))
```

راه حل (۲) بزرگترین (Max)

«دریافت سه عدد از ورودی»

```
A=int(input('عدد اولی را وارد کن'))
B=int(input('عدد دومی را وارد کن'))
C=int(input('عدد سومی را وارد کن'))
if A>=B and A>=C:
    print(A)
elif B>=A and B>=C:
    print(B)
else:
    print(C)
```

راه حل (۱)

```
A=int(input('عدد اولی را وارد کن'))
B=int(input('عدد دومی را وارد کن'))
C=int(input('عدد سومی را وارد کن'))
print(max(A,B,C))
```

راه حل (۲)

بزرگترین (Max)

نکته: اگر گزینه های شرط زیاد باشد میتوانیم از دستور elif استفاده کنیم. دستور elif وقتی اجرا می شود که اولین دستور if اشتباه باشد و اگر دستور elif نیز اشتباه باشد، در آخر دستور else اجرا می شود. (راه حل اول)

```
f=input('لطفا نام کاربری را وارد کنید')
e=input('لطفا گذرواژه را وارد کنید')
F=('admin')
E=('12345')
if f==F and e==E:
    print('خوش آمدید')
else:
    print('دسترسی غیر مجاز است')
```

← پروژه (۱) - صفحه ۶۵

```
k=int(input('ضلع اول را وارد کن'))
l=int(input('ضلع دوم را وارد کن'))
m=int(input('ضلع سوم را وارد کن'))
if k+l>m and k+m>l and l+m>k:
    print('میتوان با این سه ضلع مثلث را رسم کرد')
else:
    print('نمیتوان با این سه ضلع مثلث را رسم کرد')
```

کتابخانه های پایتون

در واقع کتابخانه ها، قطعه کدهای آماده (مجموعه دستورات با یک موضوع) هستند که استفاده از آن ها از نوشتن کدهای پیچیده و وقت گیر جلوگیری می کند. یکی از بزرگترین دلایل محبوبیت پایتون، این است که بیش از صد و سی هزار کتابخانه پایتونی در دسترس برای انتخاب وجود دارد که هر یک با هدف و کاربرد خاصی ایجاد گردیده اند. در زیر به چند نمونه از آنها اشاره شده است:

Scrapy (تجزیه و تحلیل داده ها)

Pygame (طراحی بازی)

Pillow (ویرایش تصاویر)

Turtle (گرافیکی)

و ...

کتاب خانه لاکپشت (turtle)

یکی از کتابخانه های پایتون، کتابخانه **Turtle** (لاک پشت) است که در اینمبحث مورد بررسی قرار خواهد گرفت. با کتابخانه **Turtle** می توانیم اشکال و الگوهای مختلفی را ترسیم کنیم. برای آنکه بتوانیم از کدهای کتابخانه ترتل استفاده کنیم باید آنرا به محیط پایتون فراخوانی کنیم.

```
import turtle
laki=turtle.Turtle()
# و یا
import turtle as laki
```

روش فراخوانی کتابخانه ترتل ←
با این کار در واقع ما به لاکپشت یک اسم میدهیم
که در اینجا ما به دو روش اینکار را انجام دادیم و
`turtle.Turtle()` را `laki` قرار دادیم.

توجه: میتوانیم بجای لاکي از هر حرف یا کلمه دیگری نیز استفاده کنیم. مثلا:
`t=turtle.Turtle()` و یا `import turtle as t` که در اینصورت نام لاکپشت ما `t` خواهد بود.

نکته: امکان ساخت لاکپشت های دیگری نیز وجود دارد. مانند: `laki1 , laki2` و یا `t1 , t2`

کار کلاسی صفحه ۶۷ کتاب کاروفناوری هفتم (طراحی کارت شناسایی)

```
import turtle # فراخوانی کتابخانه لاکپشت
laki=turtle.Turtle() # نام گذاری لاکپشت
# و یا
import turtle as laki # نام گذاری لاکپشت

laki.shape('turtle') # شکل لاکپشت
laki.penup() # بالا بردن قلم
laki.color('red') # رنگ لاکي
laki.pencolor('blue') # رنگ قلم
laki.pensize(10) # ضخامت قلم
laki.width(5) # ضخامت قلم
laki.pendown() # پایین آوردن قلم
laki.forward(50) # دستور ۵۰ گام به جلو
laki.fd(100) # دستور ۱۰۰ گام به جلو
laki.left(90) # چرخش ۹۰ درجه به چپ
```

برخی دستورات و کدهای
اولیه مورد نیاز برای ترسیم
کارت شناسایی ←
لازم به ذکر است که برخی
کدها را میتوانیم به صورت -
های مختلف بنویسیم، که در
کدهای روبرو مشاهده
می شوند.

مانند: ضخامت قلم و یا حرکت به جلو

```

file Edit Format Run Options window help
import turtle # فراخوانی کتابخانه لاکپشت
laki=turtle.Turtle() # نام گذاری لاکپشت
# و یا
import turtle as laki # نام گذاری لاکپشت

laki.shape('turtle') # شکل لاکپشت
laki.penup() # بالا بردن قلم
laki.color('red') # رنگ لاکي
laki.pencolor('blue') # رنگ قلم
laki.pensize(5) # ضخامت قلم
laki.pendown() # پایین آوردن قلم
laki.fd(200) # دستور حرکت ۲۰۰ گام به جلو
laki.left(90) # چرخش ۹۰ درجه به چپ
laki.fd(100) # دستور حرکت ۱۰۰ گام به جلو
laki.left(90)
laki.fd(200)
laki.left(90)
laki.fd(100)
    
```

دستورات ترسیم
کاد کارت شناسایی
←



← نحوه اجرای دستورات بالا

```

import turtle as t # فراخوانی لاکپشت با نام دیگر
t.penup()
t.goto(35,75) # رفتن به نقطه
t.pencolor('red')
t.write('نام و نام خانوادگی: علیرضا اربطی', font=('b titr', 10))
t.penup()
    
```

ادامه دستورات
← ترسیم
کارت شناسایی

```
t.goto(115,55)
t.pendown()
t.pencolor('green')
t.write('کد پرسنلی: ۸۴۱۱۴',font=('b titr',8))
t.penup()
t.goto(125,40)
t.pendown()
t.pencolor('orange')
t.write('کد ملی: ۹۵۴۲۷',font=('b titr',8))
t.penup()
t.goto(70,25)
t.pendown()
t.pencolor('navy')
t.write('دبیر کارو فناوری ناحیه ۲ زنجان',font=('b titr',8))
t.ht() # پنهان کردن لاکپشت
```

ادامه دستورات

ترسیم ←

کارت شناسایی

نام و نام خانوادگی: علیرضا اربطی

کد پرسنلی: ۸۴۱۱۴

کد ملی: ۹۵۴۲۷

دبیر کارو فناوری ناحیه ۲ زنجان

نحوه اجرای دستورات بالا ←

پروژه (۱) - صفحه ۶۸

این پروژه را به روش های

متفاوت میتوان نوشت.

در اینجا به دو روش

نوشته شده است:

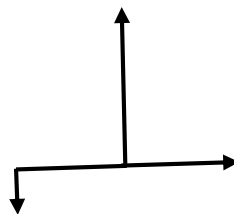
```
for i in range(4):
    laki.fd(10) # روش اول
    laki.left(90)
for j in range(4):
    laki.fd(10*2)
    laki.left(90)
for k in range(4):
    laki.fd(10*3)
    laki.left(90)
for g in range(4):
    laki.fd(10*4)
    laki.left(90)
for d in range(4):
    laki.fd(10*5)
    laki.left(90)
```

```
import turtle
laki=turtle.Turtle()
screen=turtle.Screen()
for i in range(6): # روش دوم
    for J in range(4):
        laki.forward(i*10)
        laki.left(90)
laki.hideturtle()
```

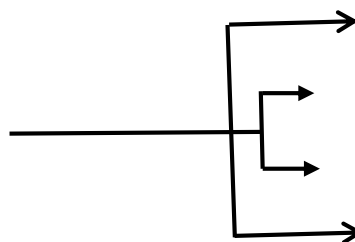
```
import turtle
laki= turtle.Turtle()
laki1= turtle.Turtle()
laki.goto(0,0) ← (۱)
laki.left(180)
laki.fd(40)
laki.left(90)
laki.fd(15)
laki1.goto(0,0)
laki1.left(90)
laki1.fd(80)
```



```
import turtle
laki= turtle.Turtle()
laki1= turtle.Turtle()
laki2= turtle.Turtle()
laki.goto(0,0)
laki.left(180) ← (۲)
laki.fd(40)
laki.left(90)
laki.fd(15)
laki1.goto(0,0)
laki1.left(90)
laki1.fd(80)
laki2.goto(0,0)
laki2.fd(60)
```



```
import turtle
laki= turtle.Turtle()
laki1= turtle.Turtle()
laki2= turtle.Turtle()
laki3= turtle.Turtle()
laki.goto(0,0)
laki.fd(80)
laki.left(90)
laki.fd(50)
laki.right(90) ← (۳)
laki.fd(40)
laki1.goto(80,0)
laki1.right(90)
laki1.fd(50)
laki1.left(90)
laki1.fd(40)
laki2.goto(90,0)
laki2.left(90)
laki2.fd(25)
laki2.right(90)
laki2.fd(20)
laki3.goto(90,0)
laki3.right(90)
laki3.fd(25)
laki3.left(90)
laki3.fd(20)
```



حلقه تکرار

برای جلوگیری از تکرار کد ها در ترسیم ها از دستورات (حلقه) **for** استفاده میکنیم.

مثال: ترسیم مربع به دو روش

```
import turtle
t=turtle.Turtle()
t.shape('turtle')
t.color('red')
for i in range(4):
    t.fd(100)
    t.left(90)
```

روش دوم با حلقه تکرار

```
import turtle
t=turtle.Turtle()
t.shape('turtle')
t.color('red')
t.fd(100)
t.lt(90)
t.fd(100)
t.lt(90)
t.fd(100)
t.lt(90)
t.fd(100)
t.lt(90)
t.lt(90)
```

روش اول بدون حلقه تکرار

پاسخ کار کلاسی صفحه ۶۹ کتاب کار و فناوری هفتم

```
t=turtle.Turtle()#رسم شش ضلعي منتظم
t.shape('turtle')
t.color('green')
t.penup()
t.goto(-100,-100)
t.pendown()
for i in range(6):
    t.fd(100)
    t.left(60)
```

کد ترسیم شش ضلعي منتظم

نکته: برای رسم چند ضلعي منتظم کافی است

عدد ۳۶۰ را به تعداد اضلاع تقسیم کنیم تا زاویه

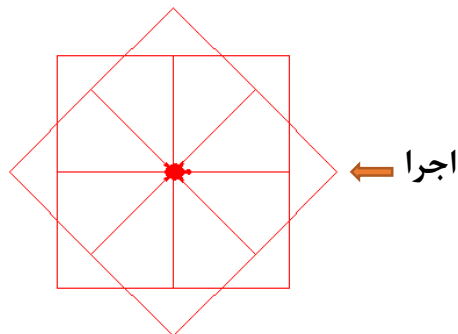
چرخش بدست آید.

```
t=turtle.Turtle()#رسم هشت ضلعي منتظم
t.shape('turtle')
t.color('green')
t.penup()
t.goto(-100,-100)
t.pendown()
for i in range(8):
    t.fd(100)
    t.left(45)
```

کد ترسیم هشت ضلعي منتظم


```
import turtle
t=turtle.Turtle()
t.shape('turtle')
t.color('red')
t.penup()
t.goto(-100,-100)
t.pendown()
for j in range(8):
    t.left(45)
    for i in range(4):
        t.fd(100)
        t.left(90)
```

دستور (کد) ←



در اینجا مشاهده کردید که دو حلقه تکرار وجود دارد. در برخی مواقع لازم است از تکرار چند باره **حلقه for** جلوگیری کنیم که اینکار توسط دستوری بنام حلقه تو در تو انجام میشود.

پروژه صفحه ۷۰

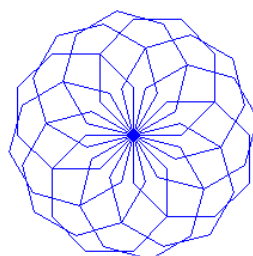
```
import turtle
t=turtle.Turtle()
t.shape('turtle')
t.color('red')
t.pencolor('blue')
t.penup()
t.goto(-100,-100)
t.pendown()
screen=turtle.Screen()
screen.title(" دوران چند ضلعي منتظم ")
z=int(screen.textinput("Z", "تعداد ضلع را وارد کنید"))
T=int(screen.textinput("T", "تعداد تکرار را وارد کنید"))
for i in range(z):
    t.fd(40)
    t.left(360/z)
for j in range(T):
    t.left(360/T)
    for i in range(z):
        t.fd(40)
        t.left(360/z)
t.ht()
```

لازم به ذکر است که برای اجرای این پروژه

یک کد اختصاصی نوشته نشده بلکه این کد

برای ترسیم کلیه چند ضلعی های منتظم با

هر چند بار تکرار نوشته شده است.



با ضلع ۸ و تکرار ۱۲