

به نام خدا

```
40
41 $(function){cards();};
42 $(window).on('resize', function(){cards();});
43 function cards(){
44   var width = $(window).width();
45   if(width < 750){
46     cardssmallscreen();
47   }else{
48     cardsbigscreen();
49   }
50 }
51 function cardssmallscreen(){
52   var cards = $('<div class="card">').length;
```

درسنامه

پودمان برنامه نویسی

(نرم افزار پایتون)

کتاب کار و فناوری پایه های هفتم، هشتم و نهم متوسطه اول

تهیه کننده: گروه آموزشی کار و فناوری استان فارس

(زهرا کفیتی، لیلا ذکاوت بخش و صدیقه یزدانی)

سال تحصیلی ۱۴۰۱-۱۴۰۲

برنامه نویسی

برخی از شایستگی هایی که در این پودمان بدست می آورید:

- ✚ کار گروهی، مسئولیت پذیری، مدیریت منابع، فناوری اطلاعات و اخلاق حرفه ای؛
- ✚ کسب مهارت حل مسئله به روش شکستن مسئله به بخش های کوچک تر؛
- ✚ پرورش تفکر الگوریتمی و حل مسئله به صورت گام به گام؛
- ✚ توانایی کار با زبان برنامه نویسی پایتون؛
- ✚ توانایی کار با کتاب خانه های پایتون؛
- ✚ توانایی استفاده کاربری از زبان برنامه نویسی پایتون برای حل مسائل ریاضی، علوم، زندگی روزمره.



چرا برنامه نویسی:

برنامه نویسی هنر نوشتن دستورالعمل هاست. ما با این کار به رایانه می گوییم که چه کاری را انجام دهد. هر برنامه مجموعه ای از دستورالعمل هاست. برای نوشتن این دستورالعمل ها از یک زبان برنامه نویسی استفاده می کنیم. پایتون یکی از بهترین زبان ها برای شروع یادگیری مفاهیم برنامه نویسی در همه گروه های سنی است.

مزایای برنامه نویسی:

- پرورش تفکر الگوریتمی، ارتقای حل مسئله و خلاقیت
- تغییر نگرش نسبت به خطا، افزایش اعتماد بنفس، احساس توانمندی در یادگیری و استدلال منطقی
- تاثیر مثبت بر قدرت درک مفاهیم ریاضی و ارتقای مهارت های اجتماعی

کاربرد و نقش رایانه ها:

- صنعت، پزشکی، آموزش، بازی های رایانه ای (سرگرمی)، حمل و نقل، خانه های هوشمند، ادارات و غیره
- سرعت پردازش بسیار بیشتر از سرعت پردازش انسان
- قدرت ذخیره سازی بیشتر و دقت بالا در محاسبات

زبان برنامه نویسی:

زبان رایانه ها که به آن زبان ماشین هم گفته می شود، زبان صفر و یک ها است و تعامل با این زبان برای انسان کار ساده ای نیست. همانطور که انسان ها برای انتقال پیام از ابزاری به نام زبان استفاده می کنند و پیام خود را در قالب کلماتی که با ساختاری قانونمند، در کنار هم چیده شده اند، به دیگری منتقل می کنند، برای تعامل با رایانه ها نیز از ابزار زبان استفاده می کنند. این نوع خاص از زبان را زبان برنامه نویسی می گویند.

نام بسیاری از زبان های برنامه نویسی را احتمالاً شنیده اید. C, C++, Pascal, Delphi, Basic, Fortran, Java, Python و ... هر یک از این زبان ها ویژگی های منحصر بفرد خود را دارد و برای هدفی خاص مورد استفاده قرار می گیرد.



چرا برنامه نویسی پایتون:



- در مقایسه با دیگر زبان های برنامه نویسی، گرامر بسیار ساده ای برای حروف چینی و اجرای برنامه ها دارد.
- فراگیری آن ساده و در عین سادگی کارایی فوق العاده ای دارد.
- کدهای آن خوانایی بالایی دارد و تعداد خط کد کمتری نسبت به زبان هایی مانند، C و Java نوشت.
- منبع کد آزاد است و استفاده از آن کاملاً رایگان است.
- کتاب های مختلف رایگان دارد که موجب قدرت فوق العاده و پر طرف دار شدن آن شده است.
- در بین مبتدیان و هم در میان حرفه ای ها طرفداران زیادی دارد و شرکت های بزرگ مانند گوگل، ناسا، دراپ باکس و بسیاری از دانشگاه ها و مراکز علمی برای پروژه های تحقیقاتی خود از این زبان استفاده کرده که نشان دهنده موفقیت و قدرت آن است.

نصب پایتون

پایتون نرم افزار متن باز است که کدهای آن به صورت رایگان برای استفاده یا ویرایش در دسترس عموم قرار دارد. نسخه پایدار و قابل اطمینان یک نرم افزار نسخه Stable است که توصیه می شود آن را نصب کنید. آخرین نسخه پایدار مناسب ویندوز در زمان نوشتن این کتاب، پایتون 3.10.1 است که این نسخه در سایت با گذشت زمان به روز می شود.

مراحل نصب

۱) ابتدا در صورتی که از نسخه سیستم عامل نصب شده روی سیستم مورد نظر آگاهی ندارید، روی آیکن This PC یا Computer راست کلیک کرده و گزینه Properties را انتخاب کنید و سپس در پنجره ظاهر شده نسخه سیستم عامل ویندوز را بررسی کنید.

۳۲ بیتی

View basic information about your computer

Windows edition
Windows 7 Ultimate ← سیستم عامل نصب شده
Copyright © 2009 Microsoft Corporation. All rights reserved.
Service Pack 1

System

Rating: Windows Experience Index

Processor: Genuine Intel(R) CPU T2130 @ 1.80GHz 1.87 GHz

Installed memory (RAM): 4.00 GB (3.24 GB usable)

System type: 32-bit Operating System

Pen and Touch: No Pen or touch input is available for this Display

نسخه سیستم عامل

۶۴ بیتی

View basic information about your computer

Windows edition
Windows 10 Enterprise ← سیستم عامل نصب شده
© 2017 Microsoft Corporation. All rights reserved.

System

Manufacturer: Novin Pendar

Processor: Intel(R) Core(TM) i7-4702MQ CPU @ 2.20GHz 2.20 GHz

Installed memory (RAM): 8.00 GB (7.76 GB usable)

System type: 64-bit Operating System, x64-based processor

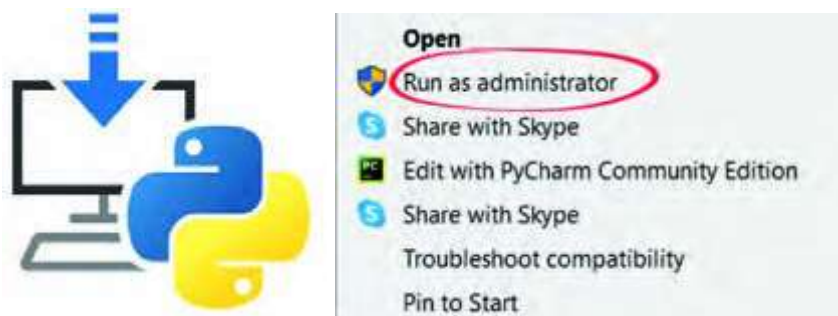
Pen and Touch: Pen Support

نسخه سیستم عامل

۲) به نشانی Python.org بروید. از فهرست Downloads گزینه Windows را برای انتخاب نسخه ویندوز کلیک کنید.



۳) پس از دانلود، روی فایل نصب راست کلیک و گزینه Run as Administrator را انتخاب کنید. سپس پیغام ظاهر شده را با انتخاب Yes یا Run تایید کنید.

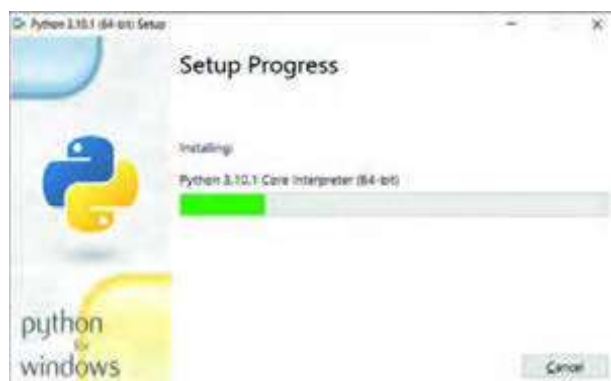


۴) توجه داشته باشید مطابق شکل زیر، هر دو گزینه را فعال کنید.

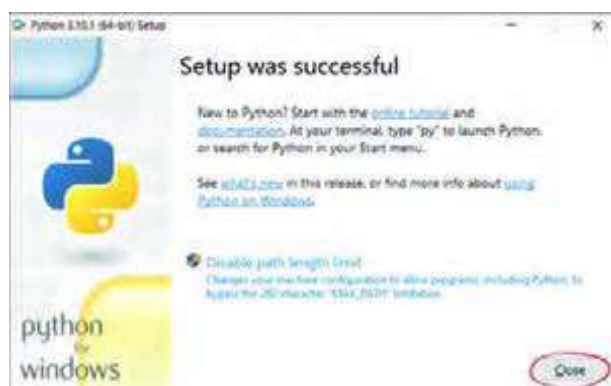
- با فعال کردن گزینه Install launcher for all users برنامه پایتون برای همه کاربران تعریف شده در ویندوز اجرا خواهد شد.
- در صورتی که گزینه Add Python 3.10 to PATH را فعال کنید، مسیر برنامه Python به متغیر PATH ویندوز اضافه می شود و می توانید به صورت مستقیم برنامه Python.org را از هر مسیری با CMD اجرا کنید. با انتخاب گزینه Install Python 3.10.1 نصب را آغاز کنید.



۵) منتظر بمانید تا فایل های لازم روی سیستم کپی شود.



۶) در پایان با ظاهر شدن پیغام "Setup was successful" دکمه close را بزنید.



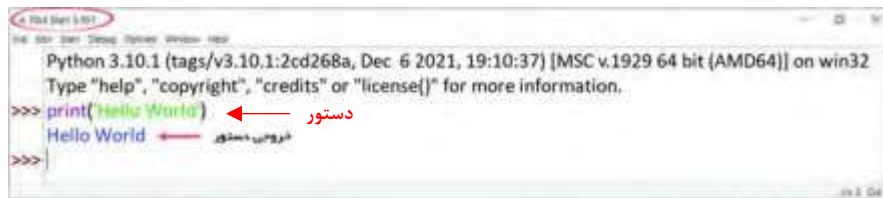
بیشتر بدانید

از دیگر IDE های محبوب برای نوشتن و اجرای برنامه پایتون ، نرم افزار Jupyter Netbook ، Visual Studio ، Code و PyCharm است.

شروع کار با پایتون

از فهرست **Start** گزینه (Python 3.10) IDLE را انتخاب می کنیم. IDLE یک محیط توسعه یکپارچه است که همه ابزارهای مورد نیاز برای نوشتن و اجرای برنامه را در اختیار ما قرار می دهد. این محیط با نصب پایتون در اختیار کاربران قرار می گیرد که به صورت زیر استفاده می شود:

الف) با اجرای IDLE، ابتدا محیط Shell Window ظاهر می شود که برای اجرا و مشاهده نتیجه دستورات به صورت خط به خط دستور `print('Hello Worlds')` را بنویسید و سپس دکمه Enter را بزنید تا نتیجه مشاهده شود. دستور `print()` برای چاپ خروجی است.



```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello World')
Hello World
>>>
```

ب) از فهرست File گزینه New File را انتخاب می کنیم تا محیط ویرایشگر متن ظاهر شود. در این محیط می توانیم پس از نوشتن مجموعه ای از دستورات، آنها را ویرایش، ذخیره و به صورت یک جا اجرا و نتیجه را مشاهده کنیم.

کار کلاسی (صفحه ۵۸ پایه هفتم، صفحه ۲۹ پایه هشتم و صفحه ۴۳ پایه نهم)



گام ۱. محیط ویرایشگر را باز کنید (متناسب با توضیحات قسمت شروع کار با پایتون)

گام ۲. دستورات مورد نظر را مطابق شکل زیر بنویسید. (untitled نشان دهنده این است که شما تاکنون نامی برای برنامه خود در نظر نگرفته اید.)



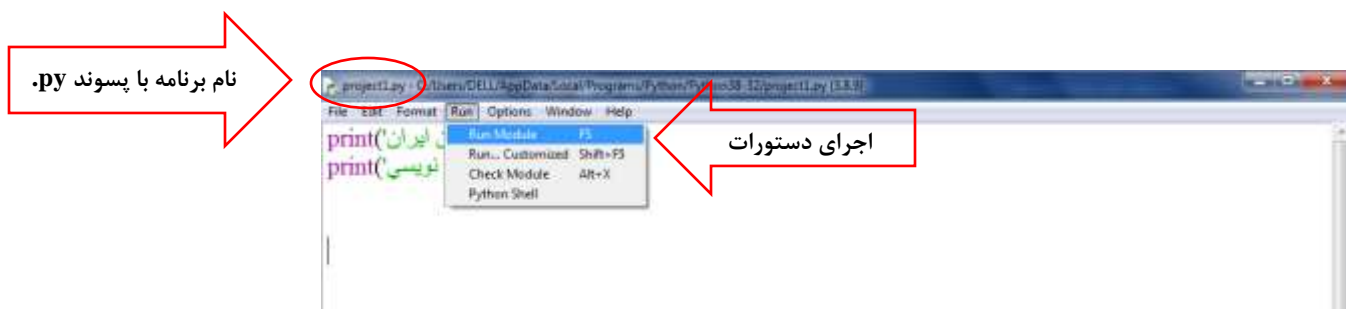
```
untitled
File Edit Format Run Options Window Help
print('سلام بر آینده سازان ایران')
print('زنگ برنامه نویسی')
```

گام ۳. از فهرست File گزینه Save را انتخاب و برنامه را با نام دلخواه در محل مناسب ذخیره کنید.

پسوند فایل ذخیره شده با برنامه پایتون **.py** است.

نکته

گام ۴. برای اجرای دستورات، از فهرست Run گزینه Run Module یا F5 را می‌زنیم.



گام ۵. خروجی برنامه در پنجره Shell نمایش داده می‌شود.



برای ویرایش برنامه های ذخیره شده از فهرست File گزینه open را انتخاب می‌کنیم و برنامه مورد نظر را فراخوانی می‌کنیم.

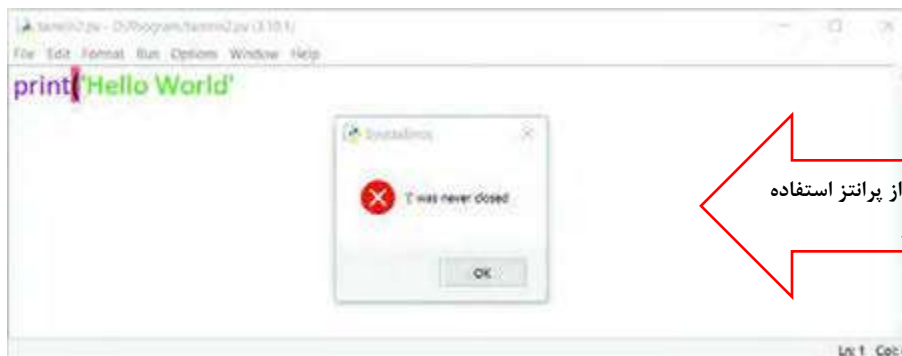
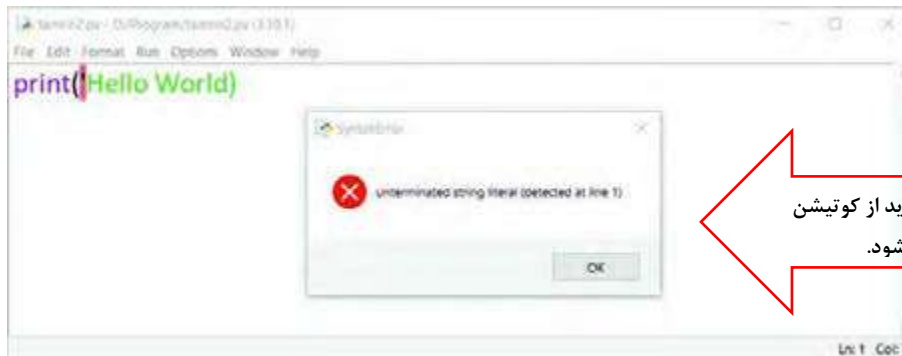
نکته

کار کلاسی (صفحه ۳۰ پایه هفتم، صفحه ۵۹ پایه هشتم و صفحه ۴۴ پایه نهم)



یکی از خطاهای رایج در برنامه نویسی، خطاهای نوشتاری است که به آن خطای نحوی (Syntax Error) گفته می شود. این خطا به دلیل اشتباهات حروف چینی رخ می دهد.

با هم کلاسی های خود، خطاهای شکل زیر را بررسی و دلیل آن را بیان کنید.



متغیر

حافظه رایانه از تعدادی خانه حافظه تشکیل شده است. متغیر (Variable) محل ذخیره اطلاعاتی مانند اعداد، متن، مجموعه ای از اعداد و... در خانه حافظه رایانه است. می توان یک متغیر را به عنوان یک برچسب برای هر خانه حافظه در نظر گرفت.

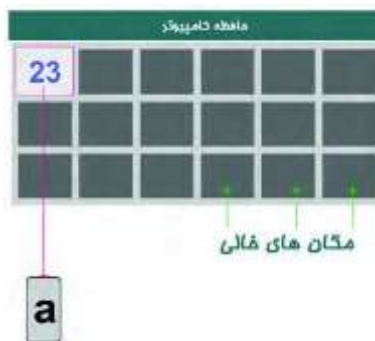
برای تعریف یک متغیر در پایتون کافی است نامی به آن داده و مقداری را در آن ذخیره کنیم. بدین منظور، نام متغیر سمت چپ، مساوی و مقدار آن سمت راست مساوی قرار می گیرد.

مقدار متغیر = نام متغیر

مثال ها:

Age=23

n=12



قوانین نام گذاری متغیرها در پایتون:

- پایتون نسبت به حروف کوچک و بزرگ حساس است. (بدین معنی که n و N ، متغیرهای متفاوتی هستند.)
- در نام گذاری متغیرها می توانیم از **حروف بزرگ** و **کوچک الفبای انگلیسی**، **اعداد** و **زیر خط** (_) استفاده کنیم.
- نام متغیرها حتما باید با حروف الفبای انگلیسی و یا زیر خط شروع شود و استفاده از اعداد در ابتدای نام متغیر مجاز نیست. (مثال: **متغیرهای مجاز**: age_2 , _age2 و **متغیر غیر مجاز**: 2age_)
- استفاده از کاراکتر فاصله در نام متغیر غیر مجاز است.
- استفاده از کلمات کلیدی زبان پایتون در نام متغیر غیر مجاز است، این کلمات کلیدی در پایتون به رنگ بنفش می باشند.(شکل زیر)

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

روش نام گذاری متغیرها:

مثال	روش	ردیف
<u>user</u> Name	کوهان شتری (Camel Case)	۱
<u>User</u> Name	پاسکال (Pascal Case)	۲
user_name	مار (Snake Case)	۳

مثال: مساحت و محیط مربع با ضلع ۱۰ سانتی متر چقدر است؟

برای نوشتن این برنامه، ابتدا حرف a را به عنوان نام متغیر ضلع و مقدار ۵ را به عنوان مقدار متغیر در نظر می گیریم. در ادامه برای محاسبه محیط مربع (p) از فرمول $a*4$ و برای مساحت (s) از فرمول $a*a$ استفاده کرده و در نهایت محیط و مساحت را نمایش می دهیم.

پاسخ: محیط مربع = ۴۰ سانتی متر و مساحت مربع = ۱۰۰ سانتی متر مربع

محاسبه محیط و مساحت مربع #

a=10 **ضلع مربع #**

p=4*a **محیط مربع #**

s=a*a **مساحت مربع #**

print ('محیط مربع', p)

print ('مساحت مربع', s)

روش دوم:

```
a=10
print ('محیط مربع', a*4)
print('مساحت مربع', a*a)
```

نکته

- ۱- از علامت # در جهت نوشتن یادداشت و توضیحات استفاده کرده و خط مرتبط با آن اجرا نمی شود.
- ۲- از روش های مختلفی برای حل مسئله می توان استفاده کرد و مهم بدست آوردن نتیجه درست است.

کار غیر کلاسی (صفحه پایه هفتم ۶۱، صفحه ۳۲ پایه هشتم و صفحه ۴۶ پایه نهم)



برنامه های زیر را بنویسید:

۱ فاطمه کتاب داستانی را در ۶ ساعت مطالعه کرد و ۱۰ صفحه از آن باقی ماند. اگر این کتاب ۱۰۰ صفحه داشته باشد، فاطمه به طور متوسط در هر ساعت چند صفحه از آن را مطالعه کرده است؟ **(پاسخ:** ۱۵ صفحه در هر ساعت)

محاسبه تعداد صفحات مطالعه شده در ساعت #

h=6 **تعداد صفحات مطالعه شده #**

r=10 **تعداد صفحات باقی مانده #**

n=100 **تمام صفحات #**

p=(n-r)/h **تعداد صفحات مطالعه شده در ساعت #**

print ('تعداد صفحات مطالعه شده در ساعت', p)

روش دوم:

```
print('(100-10)/6', 'تعداد صفحات مطالعه شده در ساعت')
```

۲ محاسبه کنید یک ساعت و بیست دقیقه و سی ثانیه، چند ثانیه است. (پاسخ: ۴۸۳۰ ثانیه)

محاسبه تعداد ثانیه ها

```
h=1          # تعداد ساعت
m=20        # تعداد دقیقه
s=30        # تعداد ثانیه
total_seconds= h*60*60 + m*60 + s # محاسبه تعداد ثانیه
print ('تعداد ثانیه در یک ساعت و ۲۰ دقیقه و ۳۰ ثانیه =', total_seconds)
```

روش دوم:

```
r= 5
print ('تعداد ثانیه=', 1*3600+20*60+30)
```

۳ مساحت و محیط یک دایره با شعاع ۵ چقدر است؟ (پاسخ: محیط دایره=۳۱.۴ و مساحت دایره ۷۸.۵)

محاسبه محیط و مساحت دایره

```
r=5          # شعاع دایره
pi=3.14      # عدد پی
p=2*pi*r     # محیط دایره
s=pi*r**2    # مساحت دایره
print ('محیط دایره =', p)
print ('مساحت دایره =', s)
```

روش دوم:

```
r= 5
print('محیط دایره=', 2*3.14*r)
print('مساحت دایره=', 3.14*r*r)
```

۴ پس انداز هفتگی محمد، ۳۰۰۰ تومان است. او حساب کرد ۵ هفته پس انداز او، نصف قیمت کیفی است که دوست دارد آن را بخرد. برنامه ای بنویسید که قیمت کیف را محاسبه و چاپ کند.؟ (پاسخ: ۳۰۰۰۰ تومان)

محاسبه قیمت کیف

```
i=3000      # پس انداز هفتگی
w=5         # تعداد هفته
p=2*w*I     # محاسبه قیمت کیف
print ('قیمت کیف =', p)
```

روش دوم:

```
r= 5
print('قیمت کیف =', 2*5*3000)
```

متن یا رشته:

نوع داده ای متنی که در زبان برنامه نویسی به آن رشته هم گفته می شود، در کنار نوع عددی از رایج ترین انواع داده مورد استفاده انسان است. این نوع داده دنباله ای از حروف، اعداد یا کاراکترها را شامل می شود که در میان "یا" قرار می گیرند.

```
Message= 'Hello, World'
```

دریافت مقدار از ورودی :

تاکنون برنامه هایی که نوشتیم همگی با مقدار ثابتی که در برنامه در نظر گرفته شده بود، اجرا می شد و نتایج آن را مشاهده می کردیم. در صورتی که بخواهیم برنامه مقداری را از کاربر دریافت کنیم، از دستور `input()` استفاده می کنیم. برای اینکه در زمان دریافت از ورودی پیغامی مناسب به کاربر داده شود، می توانیم پیغام را در یک رشته در داخل پرانتز جلوی دستور `input()` بنویسیم.

```
n= input('مقداری را وارد کنید=')
```

مثال: برنامه زیر را اجرا کرده و خروجی آن را بررسی کنید.

محاسبه مساحت مستطیل با دریافت طول و عرض از کاربر#

```
x= input('طول=')
```

```
y=input('عرض=')
```

```
s=x*y
```

```
print ('مساحت مستطیل =', s)
```

پاسخ: با اجرای این برنامه و ورودی طول ۱۲ و عرض ۴، با خطای زیر روبرو می شوید.

```
= RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python38-32/project1.py =
طول=12
عرض=4
Traceback (most recent call last):
  File "C:/Users/DELL/AppData/Local/Programs/Python/Python38-32/project1.py", line 4, in <module>
    s=x*y
TypeError: can't multiply sequence by non-int of type 'str'
>>>|
```

مقادیری که در ورودی توسط `input()` دریافت می شود، به صورت رشته داخل متغیر قرار می گیرد. در صورتی که بخواهیم مقدار عددی صحیح را دریافت کنیم، از تابع `int()` و برای اعداد اعشاری از تابع `float()` استفاده می کنیم.

نکته

حال مثال را بصورت زیر بازنویسی می کنیم:

محاسبه مساحت مستطیل با دریافت طول و عرض از کاربر#

```
x=int(input('طول='))
```

```
y=int(input('عرض='))
```

```
s=x*y
```

```
print ('مساحت مستطیل=', s)
```

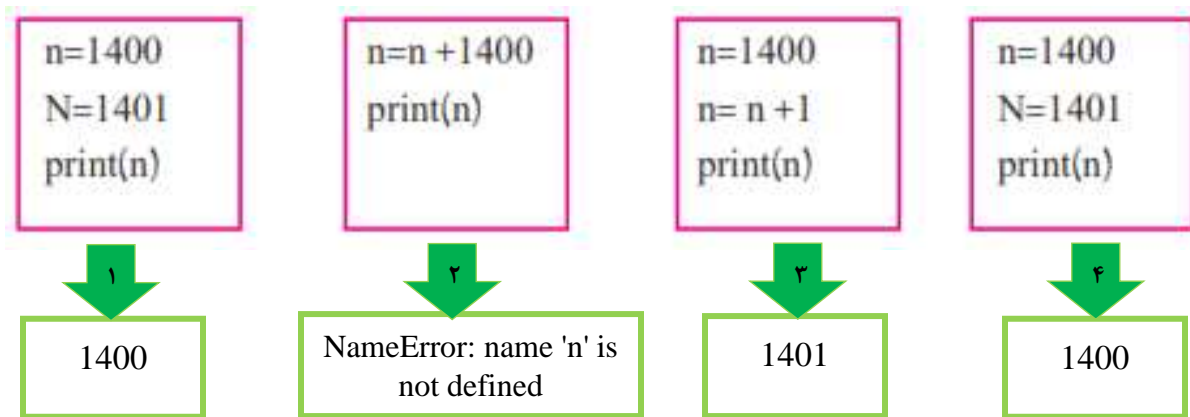
پاسخ: با اجرای این برنامه و ورودی طول ۱۲ و عرض ۴، مساحت مستطیل ۴۸ خواهد شد.

```
>>>
= RESTART: C:/Users/DELL/AppData/Local/Programs/Python/Python38-32/project1.py =
طول=12
عرض=4
مساحت مستطیل=48
>>>
```

فعالیت گروهی (صفحه پایه هفتم ۶۱، صفحه ۳۳ پایه هشتم و صفحه ۴۷ پایه نهم)



خروجی هر کد را با دوستان خود حدس بزنید و درباره آن بحث و گفت و گو کنید.



کد	توضیحات
۱	متغیر در پایتون به حروف بزرگ و کوچک حساس است، بنابراین n و N متفاوت بوده و پاسخ ۱۴۰۰ می شود.
۲	با توجه به اینکه متغیر n در ابتدا مقدار دهی نشده است، با خطای عدم مشخص بودن n مواجه می شویم.
۳	با توجه به مقدار دهی n به ۱۴۰۰، در خط دوم n با عدد ۱ جمع شده و حاصل ۱۴۰۱ می شود.
۴	متغیر در پایتون به حروف بزرگ و کوچک حساس است، بنابراین n و N متفاوت بوده و پاسخ ۱۴۰۰ می شود.

عملگرهای ریاضی:

یکی از کاربردهای اولیه رایانه، انجام عملیات ریاضی و مقایسه ای است. برای انجام محاسبات ریاضی در پایتون از عملگرهای زیر استفاده می کنیم.

مثال		شرح	عملگر
مثال عددی (a=10,b=3)	مثال با متغیر		
۱۳	a+b	جمع	+
۷	a-b	تفریق	-
۳۰	b*a	ضرب	*
۳.۳۳	a/b	تقسیم	/
۳	a//b	تقسیم صحیح	//
۱	a%b	باقیمانده صحیح	%
59049	b**a	توان	**

تقدم عملگرهای ریاضی به صورت زیر است:

نکته: پرانتز اولویت عملگرها را تغییر می دهد.

اولویت	عملگر
۱	**
۲	%, //, /, *
۳	+,-

اگر عدد ۲۵ به عنوان ورودی در دو دستور زیر داده شود، خروجی ها را با هم مقایسه و درباره آن گفت و گو کنید.

```
n=float(input('مقداری را وارد کنید = '))
print(n)
```

عدد اعشاری

25.0

```
n=int(input('مقداری را وارد کنید = '))
print(n)
```

عدد صحیح

25



برنامه های زیر را بنویسید:

۱ برنامه ای بنویسید که شعاع قاعده و ارتفاع منبع آب استوانه ای را از ورودی دریافت و سپس محاسبه کند که چند متر مکعب آب می گیرد.

محاسبه حجم مخزن استوانه ای

```
r=float(input('شعاع قاعده بر حسب متر='))
h=float(input('ارتفاع استوانه بر حسب متر='))
pi=3.14
v=pi*r**2*h
print('حجم مخزن استوانه ای بر حسب متر مکعب=', v)
```



۲ در درس علوم آموختید که کار انجام شده با مقدار نیرو در اندازه جابجایی برابر است. این رابطه را با تساوی $W=F.d$ نشان می دهیم. برنامه ای بنویسید که کار انجام شده و مقدار نیرو را از ورودی دریافت و سپس میزان جابجایی را محاسبه و چاپ کند.

محاسبه میزان جابجایی

```
w=float(input('مقدار کار انجام شده به ژول='))
f=float(input('نیروی وارد شده به نیوتون='))
d=w/f
print('جابجایی به متر=', d)
```

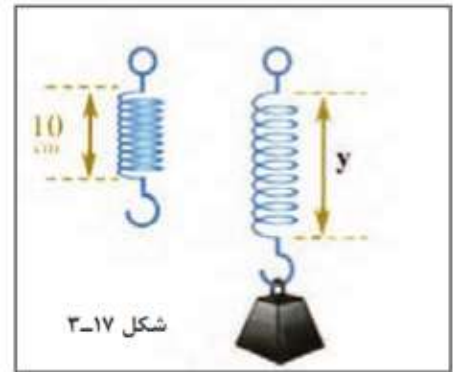
۳ طول یک فنر ۱۰ سانتی متر است. وقتی وزنه ای به جرم x به آن وصل می شود، طول فنر از رابطه $y=0.8x+10$ محاسبه می شود. برنامه ای بنویسید که از ورودی جرم وزنه ای بر حسب کیلوگرم که به آن وصل شده دریافت و سپس، طول فنر را محاسبه و چاپ کند.

محاسبه طول نهایی فنر

```
X=float (input ('جرم آویزان شده از فنر بر حسب کیلوگرم'))
```

```
y=0.8*x+10
```

```
print ('طول نهایی فنر', y)
```



عملگرهای مقایسه ای:

رایانه ها برای مقایسه از نمادهایی استفاده می کنند. برنامه نویسان به نمادهای جدول زیر عملگرهای مقایسه ای می گویند. نتایج عملگرهای مقایسه ای True (درست) و یا False (غلط) است.

مثال		شرح	علامت
مثال عددی (a=10,b=3)	مثال با متغیر		
False	a==b	مساوی	==
True	a!=b	نامساوی	!=
False	a<b	کوچک تر	<
True	a>b	بزرگ تر	>
False	a<=b	کوچکتر مساوی	<=
True	a>=b	بزرگتر مساوی	>=

شرط:

بسیاری از کارهایی که ما روزانه انجام می دهیم، بستگی به این دارد که شرطی برقرار باشد و یا برقرار نباشد. مثلا اگر هوا سرد باشد لباس های گرم می پوشیم. در برنامه نویسی نیز گاهی نیاز است قسمتی از کدها در صورت برقراری شرطی اجرا شود. برای بررسی یک شرط از دستور if به شکل زیر استفاده می کنیم.

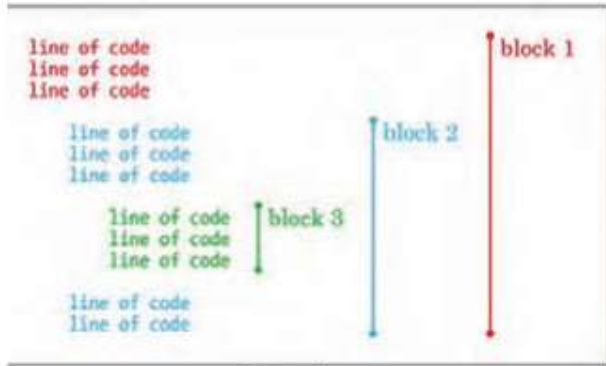


شرط if :
دستورات

در صورتی که نتیجه عبارت شرط درست (True) باشد، دستورات بلاک داخل if اجرا می شود.

بلاک کد، یک یا چند دستور است که در یک گروه قرار می گیرد و از لحاظ تو رفتگی مشابه اند. مثلا

مجموعه کدهایی که در صورت برقراری شرط if اجرا می شوند، یک بلاک کد را تشکیل می دهند. میزان تو رفتگی (indentation) دستورات هر بلاک کد به یک میزان است.



میزان تو رفتگی به اندازه یک Tab یا ۴ فاصله خالی (Space) است.

نکته

خروجی قطعه کد زیر را حدس بزنید: در این مثال temp درجه هوا است. در صورتی که درجه هوا کمتر از ۱۰ درجه باشد هوا سرد است. اگر مقدار temp=12 باشد برنامه چه خروجی خواهد داشت؟ کدام دستورات در بلاک کد قرار دارد.

```
print('سلام')
temp=8
if temp<10 :
    print('هوا سرد است')
    print('لباس گرم بپوشید')
print('خدانگهدار')
```

دستوراتی که در یک بلاک قرار دارند.

پاسخ: از آنجایی که دما ۱۲ می باشد و شرط برقرار نمی باشد، خروجی برنامه به صورت زیر است:

سلام

خدانگهدار

در صورتی که دما کمتر از ۱۰ بود و شرط برقرار بود، خروجی به صورت زیر تغییر می کرد:

سلام

هوا سرد است

لباس گرم بپوشید

خدانگهدار

کار کلاسی (صفحه ۶۴ پایه هفتم، صفحه ۳۵ پایه هشتم و صفحه ۴۹ پایه نهم)



با هم کلاسی خود برنامه ای بنویسید که یک عدد از ورودی دریافت کند و سپس مشخص کند که این عدد زوج است یا فرد.؟ **راهنمایی:** عدد زوج، عددی است که بر ۲ بخش پذیر باشد.

نمایش زوج یا فرد بودن عدد ورودی #

```
x=int(input('یک عدد دلخواه را وارد کنید'))
```

```
if (x%2==0):
```

```
    print('عدد ورودی زوج است')
```

```
if (x%2!= 0):
```

```
    print('عدد ورودی فرد است')
```

در مواقعی که شرط دارای فقط دو حالت باشد از دستور if-else استفاده می کنیم. در این دستور در صورتی که شرط دستور if برقرار باشد، بلاک کد مربوط به if اجرا و در غیر این صورت بلاک کد مربوط به else اجرا می شود.

```
if شرط :  
    دستورات  
else :  
    دستورات
```

کار کلاسی (صفحه ۶۴ پایه هفتم، صفحه ۳۵ پایه هشتم و صفحه ۴۹ پایه نهم)



با هم کلاسی های خود برنامه قبلی را با استفاده از if-else بنویسید.

نمایش زوج یا فرد بودن عدد ورودی #

```
x=int(input('یک عدد دلخواه را وارد کنید'))
```

```
if (x%2==0):
```

```
    print('عدد ورودی زوج است')
```

```
else:
```

```
    print('عدد ورودی فرد است')
```

کار غیر کلاسی (صفحه پایه هفتم ۶۵، صفحه ۳۶ پایه هشتم و صفحه ۵۰ پایه نهم)



برنامه ای بنویسید که دو عدد از ورودی دریافت و سپس عدد بزرگتر را چاپ کند.

چاپ عدد بزرگتر از دو عدد ورودی

```
x1=float (input ('عدد اول='))
```

```
x2=float (input ('عدد دوم='))
```

```
if x1>x2:
```

```
    print ('عدد بزرگتر=', x1)
```

```
else:
```

```
    print ('عدد بزرگتر=', x2)
```

کار غیر کلاسی (صفحه پایه هفتم ۶۵، صفحه ۳۶ پایه هشتم و صفحه ۵۰ پایه نهم)



برنامه ای بنویسید که سه عدد از ورودی دریافت و سپس عدد بزرگتر را چاپ کند.

نمایش بزرگترین عدد از ۳ عدد ورودی

```
x1=float (input ('عدد اول='))
```

```
x2=float (input ('عدد دوم='))
```

```
x3=float (input ('عدد سوم='))
```

```
max=x1
```

```
if x2>max:
```

```
    max=x2
```

```
if x3>max:
```

```
    max=x3
```

```
print ('عدد بزرگتر=', max)
```

پروژه (صفحه پایه هفتم ۶۵، صفحه ۳۶ پایه هشتم و صفحه ۵۰ پایه نهم)



برنامه ای بنویسید که نام کاربری و گذرواژه را برای ورود به سیستم دریافت کند و در صورتی که نام کاربری Admin و گذرواژه ۱۲۳۴۵۶۷۸ بود پیغام "خوش آمدید" صادر شود و در غیر این صورت پیغام "دسترسی غیر مجاز است" صادر شود.

دریافت نام کاربری و گذرواژه و اجاره دسترسی یا عدم آن #

```
u=input('نام کاربری را وارد نمایید')
```

```
p=input('گذرواژه را وارد نمایید')
```

```
if u== 'Admin'
```

```
    if p== '12345678':
```

```
        print('خوش آمدید')
```

```
else:
```

```
    print('دسترسی غیر مجاز')
```

روش دوم: برای برقراری همزمان دو شرط می توان از and استفاده نمود.

```
if u== 'Admin' and p== '12345678':
```

```
    print('خوش آمدید')
```

پروژه (صفحه پایه هفتم ۶۵، صفحه ۳۶ پایه هشتم و صفحه ۵۰ پایه نهم)



برنامه ای بنویسید که طول سه پاره خط را از ورودی دریافت و سپس مشخص کند که آیا می توان با این سه پاره خط مثلثی رسم کرد یا خیر.

نمایش امکان یا عدم امکان ترسیم مثلث با سه پاره خط #

```
a=float(input('طول پاره خط اول '))
```

```
b=float(input('طول پاره خط دوم '))
```

```
c=float(input('طول پاره خط سوم '))
```

```
if a< (b+c) and b<(a+c) and c<(a+b):
```

```
    print('با این سه پاره خط می توان مثلث رسم کرد')
```

```
else:
```

```
    print('با این سه پاره خط نمی توان مثلث رسم کرد')
```

کتاب خانه لاک پشت:

یکی از این کتابخانه‌های استاندارد Turtle نام دارد که یک کتابخانه پایتونی از پیش نصب شده است که کاربران را قادر می‌سازد با ایجاد بوم مجازی، تصاویر و اشکالی را ایجاد کنند. قلم روی صفحه که برای نقاشی استفاده می‌کنید لاک پشت نامیده می‌شود و این همان نام کتابخانه است. به طور خلاصه، کتابخانه لاک پشت پایتون به برنامه‌نویسان جدید کمک می‌کند تا درک کنند که برنامه‌نویسی با پایتون به روشی سرگرم کننده و تعاملی چگونه است.

توجه: وقتی می‌خواهیم لاک پشت کاری را انجام دهد، یک نقطه جلوی نام آن قرار می‌دهیم و سپس آن کار را می‌نویسیم. برخی از کدهای مورد استفاده در این کتاب خانه در جدول زیر آورده شده است.

import turtle	فراخوانی کتابخانه turtle و تعریف صفات شخصیت آن
laki=turtle.Turtle()	نام گذاری شخصیت
laki.shape('turtle')	انتخاب شکل
laki.color('نام رنگ')	تعریف و انتخاب رنگ
laki.width(ضخامت) یا laki.pensize(ضخامت)	مشخص کردن ضخامت قلم
laki.speed(مقدار عددی سرعت)	مشخص کردن سرعت اجرا (عدد سرعت وارد شود)
laki.speed('slowest')	مشخص کردن سرعت اجرا (خیلی آهسته)
laki.speed('slow')	مشخص کردن سرعت اجرا (آهسته)
laki.speed('fast')	مشخص کردن سرعت اجرا (سریع)
laki.speed('fastest')	مشخص کردن سرعت اجرا (خیلی سریع)
laki.forward(تعداد پیکسل)	حرکت به سمت جلو
laki.backward(تعداد پیکسل)	حرکت به سمت عقب
laki.left(اندازه زاویه)	چرخش به سمت چپ
laki.right(اندازه زاویه)	چرخش به سمت راست
laki.goto(x,y)	تغییر موقعیت لاک‌ی به مختصات (x,y)
laki.pendown()	دستور قلم پایین (ترسیم خطوط حین حرکت)
laki.penup()	دستور قلم بالا (عدم ترسیم خطوط حین حرکت)
laki.write('متن مورد نظر', font('نام فونت', اندازه))	نوشتن متن
laki.circle(شعاع)	ترسیم دایره
t=turtle.textinput()	دریافت از ورودی توسط لاک‌ی
laki.begin_fill()	شروع رنگ آمیزی
laki.end_fill()	پایان رنگ آمیزی
laki.hideturtle()	مخفی شدن لاک‌ی

کلمات مخفف در کدهای پایتون :

کد	مخفف
forward()	fd()
backward()	bk()
right()	rt()
left()	lt()
hideturtle()	ht()
stepposition(,) = goto(,)	setpos(,)

لیست رنگ برای استفاده در کد color :

black	bisque	forestgreen	slategrey
dimgray	darkorange	limegreen	lightsteelblue
gray	burlwood	darkgreen	cornflowerblue
grey	antiquewhite	green	royalblue
darkgray	tan	lime	ghostwhite
darkgrey	navajowhite	seagreen	lavender
silver	blanchedalmond	mediumseagreen	midnightblue
lightgray	papayawhip	springgreen	navy
lightgrey	moccasin	mintcream	darkblue
gainsboro	orange	mediumspringgreen	mediumblue
whitesmoke	wheat	mediumaquamarine	blue
white	oldlace	aquamarine	slateblue
snow	floralwhite	turquoise	darkslateblue
rosybrown	darkgoldenrod	lightseagreen	mediumslateblue
lightcoral	goldenrod	mediumturquoise	mediumpurple
indianred	cornsilk	azure	rebeccapurple
brown	gold	lightcyan	blueviolet
firebrick	lemonchiffon	paleturquoise	indigo
maroon	khaki	darkslategray	darkorchid
darkred	palegoldenrod	darkslategrey	darkviolet
red	darkkhaki	teal	mediumorchid
mistyrose	ivory	darkcyan	thistle
salmon	beige	aqua	plum
tomato	lightyellow	cyan	violet
darksalmon	lightgoldenrodyellow	darkturquoise	purple
coral	olive	cadetblue	darkmagenta
orangered	yellow	powderblue	fuchsia
lightsalmon	olivedrab	lightblue	magenta
sienna	yellowgreen	deepskyblue	orchid
seashell	darkolivegreen	skyblue	mediumvioletred
	greenyellow	lightskyblue	deeppink

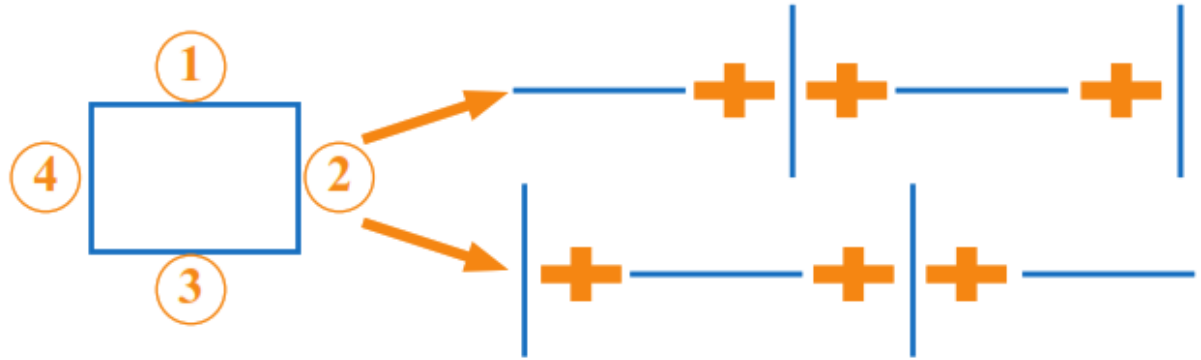
قصد داریم با کدنویسی پایتون یک کارت شناسایی مشابه شکل روبرو طراحی کنیم. برای شروع چه کاری باید انجام دهیم؟ از کجا باید شروع کنیم؟

Name: Ali
 Last Name: Bahrami
 Student Code: 1234567890
 Mobile: 09131111111

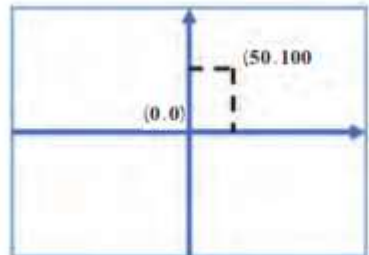
یکی از ایده های حل مسئله، تقسیم آن به قسمت های کوچک تر است. برای رسم کادر مستطیل شکل کارت شناسایی به مراحل زیر دقت کنید.



حالا باز هم کار را به قسمت های کوچک تر تقسیم می کنیم.



برای رسم بهتر است از محور مختصات استفاده می کنیم.



محور مختصات

کوچک ترین عنصر قابل مشاهده در صفحه، پیکسل است.

نکته



کارت شناسایی

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
laki.color('blue')
laki.width(3)
laki.up()
laki.goto(-300,100)
laki.down()
laki.forward(300)
laki.left(90)
laki.forward(200)
laki.left(90)
laki.forward(300)
laki.left(90)
laki.forward(200)
laki.left(90)
laki.up()
laki.color('black')
laki.goto(-250,270)
laki.write('Name: Ali', font=('times new roman',15))
laki.goto(-250,220)
laki.write('Last Name: Bahrami', font=('times new roman',15))
laki.goto(-250,170)
```

```
laki.write('Student Code: 1234567890', font=('times new roman',15))
```

```
laki.goto(-250,120)
```

```
laki.write('Mobile: 09131111111', font=('times new roman',15))
```

```
laki.ht()
```

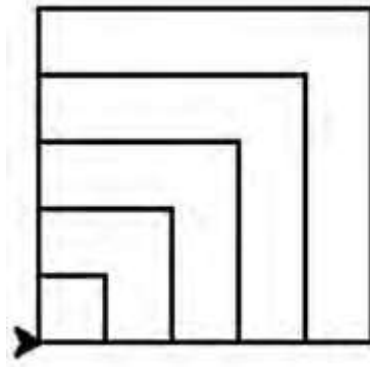
پروژه (صفحه پایه هفتم ۶۸، صفحه ۳۹ پایه هشتم و صفحه ۵۳ پایه نهم)



شکل های زیر را با لاکی ترسیم کنید.

پاسخ: این شکل از ۵ مربع تشکیل شده است که به ترتیب اضلاع آنها را ۲۰، ۴۰، ۶۰، ۸۰ و ۱۰۰ در نظر می گیریم. در ابتدا به مختصات (۰,۰) رفته و مربع اول را رسم می کنیم و برای رسم مربع های دیگر مجدد به همین نقطه می رویم.

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
laki.color('black')
laki.width(3)
laki.speed('fast')
laki.goto(0,0)
laki.forward(20)
laki.left(90)
laki.forward(20)
laki.left(90)
laki.forward(20)
laki.left(90)
laki.forward(20)
laki.left(90)
laki.up()
laki.goto(0,0)
laki.down()
```



laki.forward(40)
laki.left(90)
laki.forward(40)
laki.left(90)
laki.forward(40)
laki.left(90)
laki.forward(40)
laki.left(90)
laki.up()
laki.goto(0,0)
laki.down()
laki.forward(60)
laki.left(90)
laki.forward(60)
laki.left(90)
laki.forward(60)
laki.left(90)
laki.forward(60)
laki.left(90)
laki.up()
laki.goto(0,0)
laki.down()
laki.forward(80)
laki.left(90)
laki.forward(80)
laki.left(90)
laki.forward(80)
laki.left(90)

laki.forward(80)

laki.left(90)

laki.up()

laki.goto(0,0)

laki.down()

laki.forward(100)

laki.left(90)

laki.forward(100)

laki.left(90)

laki.forward(100)

laki.left(90)

laki.forward(100)

laki.left(90)

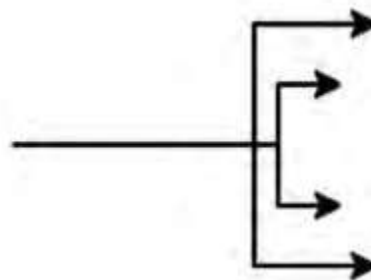
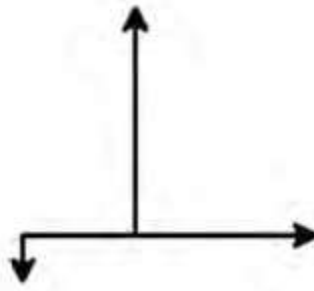
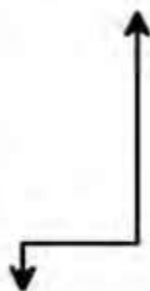
laki.ht()

پروژه (صفحه پایه هفتم ۶۸، صفحه ۳۹ پایه هشتم و صفحه ۵۳ پایه نهم)



شکل های زیر را با لاکي ترسيم كنيد.

برای رسم شکل های زیر چند لاکي تعريف كنيد.



ترسیم شکل سمت چپ#

```
import turtle  
laki1=turtle.Turtle()  
laki1.color ('black')  
laki1.width(3)  
laki1.left(90)  
laki1.forward(100)
```

```
laki2=turtle.Turtle()  
laki2.color ('black')  
laki2.width(3)  
laki2.forward(-50)  
laki2.left(-90)  
laki2.forward(20)
```

ترسیم شکل وسط#

```
import turtle  
laki1=turtle.Turtle()  
laki1.color('black')  
laki1.width(3)  
laki1.forward(100)
```

```
laki2=turtle.Turtle()  
laki2.color('black')  
laki2.width(3)  
laki2.left(90)  
laki2.forward(100)
```

```
laki3=turtle.Turtle()
laki3.color('black')
laki3.width(3)
laki3.forward(-50)
laki3.left(-90)
laki3.forward(20)
```

ترسیم شکل سمت راست#

```
import turtle
laki1=turtle.Turtle()
laki1.color('black')
laki1.width(3)

laki2=turtle.Turtle()
laki2.color('black')
laki2.width(3)

laki3=turtle.Turtle()
laki3.color('black')
laki3.width(3)

laki4=turtle.Turtle()
laki4.color('black')
laki4.width(3)

laki1.penup()
laki1.goto(0,100)
laki1.pendown()
laki1.forward(200)
laki1.left(90)
```

laki1.forward(50)

laki1.left(-90)

laki1.forward(50)

laki2.penup()

laki2.goto(200,100)

laki2.pendown()

laki2.left(-90)

laki2.forward(50)

laki2.left(90)

laki2.forward(50)

laki3.penup()

laki3.goto(185,100)

laki3.pendown()

laki3.left(90)

laki3.forward(80)

laki3.left(-90)

laki3.forward(80)

laki4.penup()

laki4.goto(185,100)

laki4.pendown()

laki4.left(-90)

laki4.forward(80)

laki4.left(90)

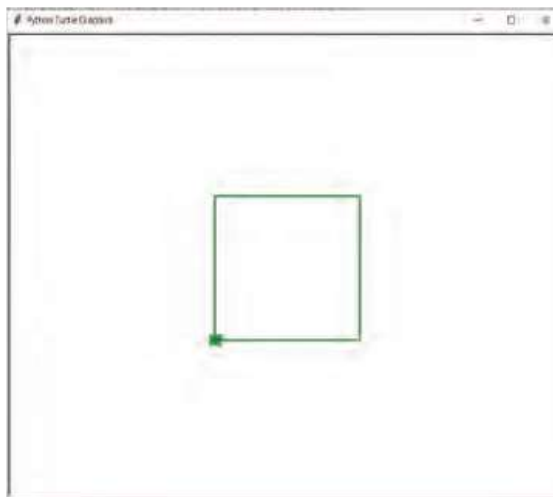
laki4.forward(80)

حلقه تکرار

به کدهای زیر نگاه کنید و کدهای تکراری متوالی را مشخص کنید.

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.penup()
laki.goto(-100,-100)
laki.pendown()
laki.forward(200)
laki.left(90)
laki.forward(200)
laki.left(90)
laki.forward(200)
laki.left(90)
laki.forward(200)
laki.left(90)
```

} کدهای تکراری



کد بالا یک مربع در وسط صفحه رسم می کند. حالا خودتان یک شش ضلعی و سپس یک هشت ضلعی رسم کنید. در کدهای جدید نیز، کدهای تکراری بیشتری وجود دارد. برای جلوگیری از تکرار آنها از حلقه for استفاده می کنیم.

ترسیم ۶ ضلعی

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.forward(200)
laki.left(60)
laki.forward(200)
laki.left(60)
laki.forward(200)
laki.left(60)
```



```
laki.forward(200)
```

```
laki.left(60)
```

```
laki.forward(200)
```

```
laki.left(60)
```

```
laki.forward(200)
```

```
laki.left(60)
```

```
# ترسیم ۸ ضلعی
```

```
import turtle
```

```
laki=turtle.Turtle()
```

```
laki.shape('turtle')
```

```
laki.color('green')
```

```
laki.width(3)
```

```
laki.forward(200)
```

```
laki.left(45)
```

```
laki.forward(200)
```

```
laki.left(45)
```

```
laki.forward(200)
```

```
laki.left(45)
```

```
laki.forward(200)
```

```
laki.left(45)
```

```
laki.forward(200)
```

```
laki.left(45)
```

```
laki.forward(200)
```

```
laki.left(45)
```

```
laki.forward(200)
```

```
laki.left(45)
```

```
laki.forward(200)
```

```
laki.left(45)
```

شکل کلی دستور for

در کد زیر متغیر i شمارنده حلقه است. کلمه range به همراه عدد داخل پرانتز، بیانگر اعداد از صفر تا یک کمتر از عدد داخل پرانتز است. یعنی range(5) به معنای اعداد ۰، ۱، ۲، ۳ و ۴ است که تعداد دفعات تکرار دستورات داخل بلاک for را مشخص می کند.

```
>>> for i in range(5):  
        print("hello world!")
```

```
hello world!  
hello world!  
hello world!  
hello world!  
hello world!
```

برای رسم مربع از قطعه زیر به جای دستورات تکراری استفاده می کنیم.

```
For i in range(4):  
    laki.forward(200)  
    laki.left(90)
```

کار کلاسی (صفحه ۶۹ پایه هفتم، صفحه ۴۱ پایه هشتم و صفحه ۵۵ پایه نهم)



با استفاده از دستور for، کدهای رسم شش ضلعی و هشت ضلعی را بازنویسی کنید.

ترسیم ۶ ضلعی #

```
import turtle  
laki=turtle.Turtle()  
laki.shape('turtle')  
laki.color('green')  
laki.width(3)  
for i in range (6):
```

```
laki.forward(200)
```

```
laki.left(360/6)
```

ترسیم ۸ ضلعی #

```
import turtle
```

```
laki=turtle.Turtle()
```

```
laki.shape('turtle')
```

```
laki.color('green')
```

```
laki.width(3)
```

```
for i in range (8):
```

```
    laki.forward(200)
```

```
    laki.left(360/8)
```

توجه: پروژه ترسیم ۵ مربع (صفحه ۶۸ پایه هفتم، صفحه ۳۹ پایه هشتم و صفحه ۵۳ پایه نهم) را می توان با استفاده از حلقه for به صورت ساده تر بازنویسی کرد.

```
import turtle
```

```
laki=turtle.Turtle()
```

```
laki.shape('turtle')
```

```
laki.color('black')
```

```
laki.width(3)
```

```
laki.speed('fast')
```

```
for j in range(20,120,20):
```

```
    for i in range(4):
```

```
        laki.forward(j)
```

```
        laki.left(90)
```

```
laki.hideturtle()
```

کار غیر کلاسی (صفحه پایه هفتم ۶۹، صفحه ۴۱ پایه هشتم و صفحه ۵۵ پایه نهم)

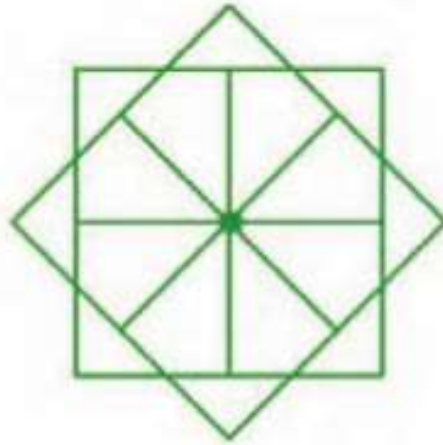


شکل زیر را با کدهایی که تا اینجا آموختید، رسم کنید.

راهنمایی: این شکل از ۸ مربع با زاویه چرخش ۴۵ درجه تشکیل شده است.

ترسیم ۸ مربع با زاویه چرخش ۴۵ درجه #

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.speed('fast')
for i in range (8):
    for j in range(4):
        laki.forward(200)
        laki.left(90)
    laki.left(45)
laki.ht()
```



پروژه (صفحه پایه هفتم ۷۰، صفحه ۴۲ پایه هشتم و صفحه ۵۶ پایه نهم)



برنامه ای بنویسید که تعداد ضلع های شکل و تعداد تکرار آن را از ورودی دریافت و سپس طرح کامل را ترسیم

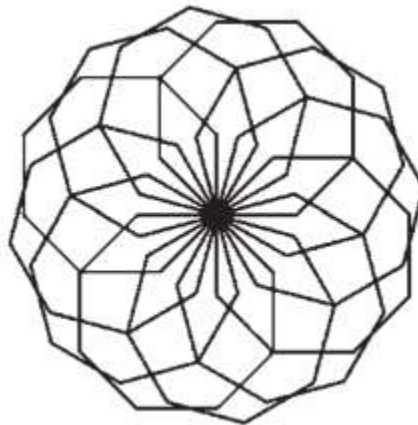
کند. از دستورات جدول زیر کمک بگیرید.

laki.speed (سرعت)	تغییر سرعت لاکه
laki.circle (شعاع)	رسم دایره با لاکه
laki.textinput ('پیام', 'عنوان کادر')	دریافت از ورودی توسط لاکه

راهنمایی: شکل زیر یک هشت ضلعی است که ۱۲ مرتبه تکرار شده است.

پروژه ترسیم چند ضلعی با دریافت تعداد اضلاع و تعداد تکرار از کاربر#

```
import turtle
laki=turtle.Turtle()
laki.color('black')
laki.width(3)
laki.speed('fastest')
x=int(turtle.textinput('تعداد اضلاع','چند ضلعی'))
y=int(turtle.textinput('تعداد تکرار شکل','چند شکل'))
for j in range(y):
    for i in range(x):
        laki.forward(100)
        laki.left(360/x)
    laki.left(360/y)
```



" پایان "