اصول برنامه سازی به زبان python

فصل اول – مبانی

تهیه و تنظیم : محمد نعیمی

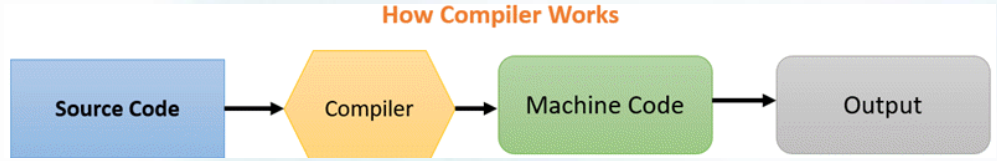
استادیار دانشگاه آزاد اسلامی

مفاهیم پایه ای در برنامه نویسی

یک «برنامه» در حقیقت دستورالعمل‌های متوالی است که می‌تواند توسط یک کامپیوتر اجرا شود.
معمولا از **زبانهای سطح بالا**(high level language) برای برنامه نویسی استفاده می شود.
زبان سطح بالا: به زبانهایی گفته می شود که دستورات آن برای انسان قابل فهم است یا به بیان دقیق تر شامل کلمات و عبارات معمول در زبانهایی مثل انگلیسی و ... می باشد.
زبان ماشین: کامپیوتر درکی از برنامه نوشته شده در زبان سطح بالا ندارد و تنها میتواند برنامه هایی که در فرم باینری که تنها حاوی ارقام 0 و 1 هستند **(زبان ماشین)** را درک نماید.

- برنامه نوشته شده در زبان سطح بالا (Source Code) باید طی فرایندی به زبان ماشین تبدیل شود.
- برای این منظور دو روش وجود دارد که بر اساس آن زبان‌های برنامه‌نویسی به دو نوع اصلی زبان‌های مفسری (interpreter) و زبان‌های کامپایلری (compiler) تقسیم می‌شود.

کامپایلر (compiler) :
یک برنامه کامپیوتری است که **کل برنامه** نوشته شده به زبان سطح بالا را به **کد ماشین** تبدیل میکند. جهت کامپایل، برنامه باید با قوانین نحوی (syntax) زبان برنامه نویسی مربوطه مطابقت داشته باشد. کامپایلر تنها یک برنامه است و نمی تواند خطاهای موجود در برنامه شما را برطرف کند. بنابراین، اگر در برنامه خود خطای نحوی داشته باشید باید آن را اصلاح کنید در غیر این صورت کامپایل نمی شود.



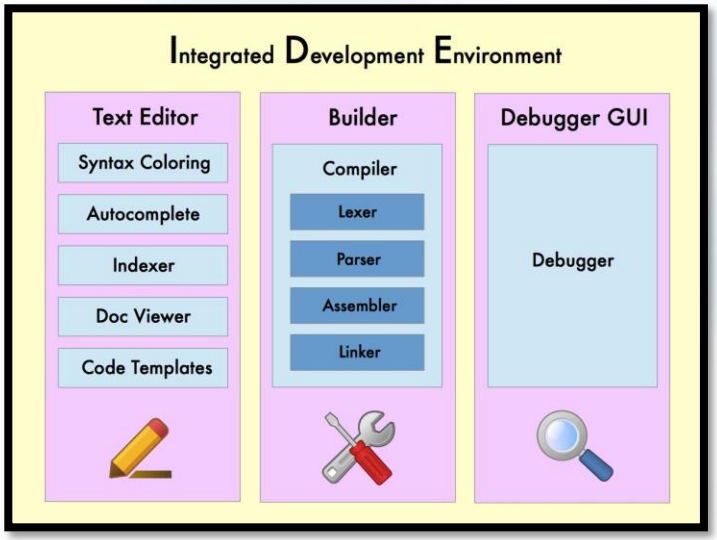
مفسر (interpreter) :
یک برنامه کامپیوتری است که **هر دستور برنامه** نوشته شده به زبان سطح بالا را به **کد ماشین** تبدیل می کند. هم کامپایلر و هم مفسر کار یکسانی را انجام می دهند که عبارت است از تبدیل زبان برنامه نویسی سطح بالا به کد ماشین اما، یک کامپایلر **قبل از اجرای برنامه**، برنامه را به کد ماشین تبدیل می کند (یک فایل اجرایی ایجاد می کند) اما مفسر، هنگامی که برنامه اجرا می شود، دستورات برنامه را به کد ماشین تبدیل می کنند.

تفاوت کامپایلر و مفسر

کامپایلر	مفسر
کامپایلر کل برنامه را اسکن میکند و کل آن را به یکباره به کد ماشین تبدیل نموده و بعد کد ماشین اجرا می شود	مفسر در هر لحظه یک دستور از برنامه سطح بالا را به کد ماشین تبدیل و اجرا می کند
کامپایلر زمان زیادی را برای تجزیه و تحلیل کد برنامه صرف می کند. اما، در زمان اجرا بسیار سریعتر است.	مفسر زمان بسیار کمتری برای تجزیه و تحلیل کد برنامه صرف می کند. اما در زمان اجرا بسیار کندتر است.
کامپایلر کد واسط تولید می کند لذا از نظر مصرف حافظه بدتر می باشد	مفسر کد واسط تولید نمی کند لذا از نظر مصرف حافظه بهتر می باشد
C، C++ و Java از کامپایلر استفاده می کنند.	Python و Matlab از مفسر استفاده می کنند.

IDE یا محیط توسعه یکپارچه(Integrated Development Environment) :

یک محیط برنامه نویسی است که مجموعه ای از نرم افزارهای که معمولا شامل ویرایشگر کد برنامه (Code Editor)، کامپایلر یا مفسر، اشکال زدا (Debugger) و در نهایت رابط گرافیکی یا GUI می باشند را در یکجا جمع کرده است و در اختیار برنامه نویس قرار می دهد.



- برای زبان C++ محیط های IDE های مختلفی وجود دارد که استفاده از آنها متناسب با سلیقه افراد می باشد.
- codeblocks و vs code دو نمونه از نمونه های مختلف IDE جهت زبان C++ می باشند.
- برخی از IDE ها توسط چندین زبان مورد استفاده قرار میگیرند نظیر vs code

متغیر – variable

متغیر(variable):

فضایی از حافظه اصلی است که ۱– یک نام دارد ۲– دارای نوع است
از متغیر ها برای ذخیره و بازیابی مقادیر استفاده میکنیم

شناسه (identifier):

نامی است برای مشخص نمودن متغیرها، توابع و ... مورد استفاده قرار میگیرد.
شناسه های مجاز با حروف یا _ شروع شده و در ادامه میتواند حروف، رقم یا _ قرار گیرد.
جای خالی بین حروف شناسه مجاز نیست و
python یک زبان Case Sensitive است یعنی A وa را یکی نمی‌داند.
چهار نام مقابل برای python متفاوت در نظر گرفته میشود AB Ab aB ab
بلند ترین طول مجاز برای نام یک متغیر ۳۱ کاراکتر است.
انتخاب نامهایی که با کلمات کلیدی python یکی باشد مجاز نیست. برخی از این کلمات (if for while end)

نامهای زیر برای نام گذاری متغیر ها در python معتبر نمی باشند

(کاراکتر غیر مجاز) a4*e (کاراکتر غیر مجاز) we.4 (نام کلیدی) for (شروع با رقم) 1x

نمونه ای از چند نام معتبر

X12sx2 _t2 a_3cc3 a2323 for1 if12

قواعدی با عنوان clean code مطرح میشود که میتوان از آنها برای نام گذاری بهتر متغیر ها استفاده نمود.

برخی توصیه ها در نام گذاری متغیر ها:

- **نام متغیر ها با حروف کوچک شروع شوند**
- **در نامهایی که ترکیب چند کلمه هستند حروف ابتدایی کلمات دوم به بعد حروف بزرگ باشند** studentName
- **از کلمات اختصاری نا مفهوم استفاده نکنید**

نوع داده – datatype

	datatype	نوع مقدار	مثال
عددی	int	صحیح	a1=5 a3=548 a4=3251425 a5=1215451515
	float	اعشار دار	f1=2.0 f2=2.56 f3=15.845 f4=21.958747
	complex	مختلط	d1=2+7j d2=3-5j d3=2j
منطقی	bool	صحیح و غلط	x=True y=False
دارای ترتیب	str	مجموعه ای از کارکتر	s1="string in a double quote" s2='string in a single quote'
	list	آرایه با محتوای داری نوع متفاوت – دارای ترتیب–قابل تغییر ساختار []	L1=[1,2,3,4,5,6] L2=["hello","john","reese"] L3= ["hey","you",1,2,3,"go"]
	tuple	آرایه با محتوای داری نوع متفاوت – دارای ترتیب–غیر قابل تغییر ساختار ()	t1=(1,2,3,4,5,6) t2=("hello","john","reese") t3= ("hey","you",1,2,3,"go")
	range	دنباله اعداد	range(first,end,step) range(6)
نگاشت	dic	دوتایی (key/value) دارای ترتیب ساختار {} key منحصر به فرد جهت دسترسی به value	D={key1:val1,key2:val2, … ,keyn:valn} std={'name':'Ali','family':'Hasani','age':31,12:'ok'} **باشد مثال : keyجهت دسترسی به مقادیر اندیس باید مقدار** std['name'] std[12]
مجموعه	set	عناصر غیرتکراری و بدون ترتیب	set1 = set(["Geeks",12 , 6 , "Geeks"])

رشته

۱– برنامه ای بنویسید که عبارت رو به رو را چاپ کند. In the name of GOD

این برنامه به دو صورت قابل نوشتن است. سمت چپ به صورت چاپ مستقیم رشته و سمت راست با استفاده از متغیر

code
```
print('In the name of GOD')
```
output
```
In the name of GOD
```

code
```
message='In the name of GOD'
print(message)
```
output
```
In the name of GOD
```

برنامه زیر دارای خطا می باشد زیرا نام متغیر در قسمت تعریف و مقدار دهی (خط ۱) با قسمتی که از آن استفاده شده (خط ۲) یکسان نمیباشد لذا از خط ۲ بابت استفاده از نامی که شناخته شده نیست خطا گرفته است.
❶اعلام میکند خطای در خط ۲ از فایل pro001.py رخ داده است ❷بیان میکند در قسمت masage خطا رخ داده است یا این قسمت برای سیستم دارای مشکل است ❸خطا از نوع NameError است و بیان میکند نام mesage شناخته شده نیست و بر اساس برنامه می پرسد آیا منظور شما کلمه message است (چون متغیر message) را ساختیم

```
message='In the name of GOD'
print(mesage)
```

```
❶File "d:\pro001.py", line 2, in <module>
❷    print(mesage)    در خط 2    در <ماژول> تابع
           ^^^^^^
❸NameError: name 'mesage' is not defined. Did you mean: 'message'?
```

نام ’mesage‘ شناخته شده نیست

رشته (۲)

۲- برنامه ای بنویسید که عبارت رو به رو را چاپ کند.

Hello Python world!
I love Python.

این برنامه به دو صورت قابل نوشتن است. سمت چپ به صورت چاپ مستقیم رشته و سمت راست با استفاده از متغیر

```
print('Hello Python world!')
print('I love Python.')
```

```
Hello Python world!
I love Python.
```

```
message='Hello Python world!'
print(message)
message='I love Python.'
print(message)
```

```
Hello Python world!
I love Python.
```

رشته را میتوان بین ‘ ‘ یا بین “ “ قرار داد و این باعث می شود بتوانید از خود این نشانه ها در رشته استفاده کنید.

```
s1"This is a string."
msg='Also this is a string.’
a='I told my friend, "I love Python!“’
sen="Python's a functional language."
```

متن چند خطی را میتوان بین “”” """ یا ‘’’ ’’’ قرار داد

```
m="""This is a string.
Also this is a string."""
```

رشته – ترکیب و کاراکتر های کنترلی

f-string : این ساختار جهت افزودن و ترکیب مجموعه ای از رشته ها و محتوای متغیر ها می باشد. در این ساختار متغیر ها درون {} قرار داده میشوند تا مشخص شود محتوای متغیر مورد نیاز است نه خود عبارت.
این ساختار در دو فرم قابل استفاده است که درمثال های زیر نمایش داده شده است.

```
name='Mohammad'
family='Naimi'
fullName=f'I am {name} {family}'
print(fullName)
```

I am Mohammad Naimi

```
name='Mohammad'
family='Naimi'
fullName='I am {} {}.'.format(name,family)
print(fullName)
```

I am Mohammad Naimi.

البته میتوان در print چند مقدار جهت چاپ وارد نمود. تابع print مقادیر داده شده را با یک فاصله بعد از هر قسمت کنار هم چاپ میکند. البته جهت ساخت رشته ترکیبی باید از یکی از دو روش بالا استفاده نمود.

```
name='mohammad'
family='naimi'
print(f'I am {name} {family}.')
print('I am',name,family,'.')
```

I am mohammad naimi.
I am mohammad naimi .

در قسمت خروجی برنامه روبرو، برای درک بهترموضوع، جاهای خالی که دستور print بعد از هر قسمت اضافه میکند را قرمز نموده ایم

نکته برنامه نویسی:
خطوطی که ابتدایی آنها # باشد به عنوان comment در نظر گرفته می شود و اجرا نمیشود.
از هرجای خط # قرار دهیم از آنجا به بعد به عنوان comment در نظر گرفته میشود.
در صورت انتخاب چند خط باز زدن Ctrl+/ خطوط انتخابی به صورت comment و بر عکس خواهند شد.

رشته – توابع

با قرار دادن . بعد از رشته، توابعی که روی رشته قابل عمل هستند لیست می شود. در زیر عملکرد برخی از آنها را نمایش داده ایم

```
fullName='mohammad naimi'
print(fullName.title())   حروف اول کلمات را بزرگ میکند
print(fullName.upper())   تمام حروف کلمات را بزرگ میکند
print(fullName.lower())   تمام حروف کلمات را کوچک میکند
print(fullName)
```

```
Mohammad Naimi
MOHAMMAD NAIMI
mohammad naimi
mohammad naimi
```

```
fullName=' python '
print(fullName.rstrip())   حذف جاخالی های انتهای رشته(سمت راست)
print(fullName.lstrip())   حذف جاخالی های ابتدای رشته(سمت چپ)
print(fullName.strip())    حذف جا خالی از دو طرف رشته
print(fullName)
```

```
 python
python
python
 python
```

print – برخی نکات

پس از اجرای هر دستور print مکان نما به صورت خودکار به خط بعد خواهد رفت. به عبارت دیگر پایان دستور print به صورت پیشفرض \n می باشد (end='\n')
با مقدار دهی به متغیر end، هنگام استفاده از print میتوان کارکتر دیگری به انتهای

\n	برو به خط جدید
\t	فضای خالی به اندازه tab
\"	کاراکتر "
\'	کاراکتر '
\\	کاراکتر \

کاراکتر های کنترلی: برخی کاراکتر ها به صورت ترکیب یک کاراکتر با \ می باشد که معنی انجام یک عملیات یا یک کاراکتر را میدهد. این کاراکتر ها در هنگام دستور چاپ یا درون برنامه داری معنی خاصی هستند.
مثلا:
برای اینکه کاراکتر \ را بخواهیم چاپ کنیم لازم است \\ را بنویسیم.
هنگام چاپ کاراکتر \n به خط جدید جهت ادامه چاپ خواهیم رفت.

در مثال زیر چون رشته با ' ' مشخص شده استفاده از ' در رشته ایجاد خطا میکند لذا باید \' استفاده کنیم یا رشته را کلا درون " " قرار دهیم.

پس از اجرای هر دستور print مکان نما به صورت خودکار به خط بعد خواهد رفت. به عبارت دیگر پایان دستور print به صورت پیشفرض \n می باشد (end='\n')
با مقدار دهی به متغیر end، هنگام استفاده از print میتوان کارکتر دیگری به انتهای

```
relation='son'
name='Ali'
age=21
sen=f'My {relation}\'s name is {name}.\nHe is {age} years old.'
print(sen)
```

```
My son's name is Ali.
He is 21 years old.
```

عملگر انتساب =

برای مقدار دهی به یک متغیر از عملگر = استفاده میکنیم. نحوه استفاده از این عملگر به شکل زیر می باشد

; مقدار = نام متغیر

1-عدد	x = 15	عدد 15 را در متغیر x قرار میدهد
2-متغیر	x = y	محتوای متغیر y را در متغیر x قرار میدهد
3-رشته	s = 'y'	رشته y را در متغیر s قرار میدهد
4-فرمول	x = 2 * a + 5	حاصل عملیات را در متغیر x قرار میدهد

میتوان چند انتساب با یک عملگر انجام داد — عدد 5 را در x و 3 را در y و 12 را در z قرار میدهد. — x,y,z=5,3,1

میتوان از چند عملگر انتساب در یک دستور استفاده نمود. — عدد 15 را در سه متغیر x، y و a قرار میدهد. — x = y = a = 15

برای رشته ها لازم است آن را درون ' ' قرار دهیم وگرنه برنامه آن را به صورت نام متغیر در نظر میگیرد
به تفاوت مفهوم x=y و s='y' دقت نمایید.

تابع ورودی

نام متغیر= input('پیغامی که هنگام گرفتن مقدار میخواهیم چاپ شود')

```
Name=input("what's your name:")
```

تابع input همیشه رشته ای به عنوان خروجی بر میگرداند. لذا نوع متغیر به صورت رشته در نظر گرفته میشود. جهت ورود اعداد لازم است تابع input در توابع int یا float قرار داده شوند تا مقدار ورودی را تبدیل به فرم عددی نمایند. به تبدیل نوع داده ها به یکدیگر casting گفته می شود. str نیز برای تبدیل اعداد به رشته استفاده می شود.

```
a=float(input('a:'))
b=int(input('enter a number:'))
num=input('how many books?');
```

پیام a: چاپ شده و مقداری وارد شده در فرم عدد اعشار دار در متغیر a قرار داده میشود

پیام enter a number: چاپ و مقداری که وارد کنید در فرم عدد صحیح در متغیر b قرار داده میشود

پیام how many books? چاپ شده و مقداری که وارد کنید در متغیر num قرار داده میشود(این دستور غلط است زیرا متغیر num را به عنوان رشته در نظر میگیرد و امکان انجام محاسبات ریاضی روی این متغیر وجود ندارد)

عملگر های ریاضی و انتساب

عملگر	مفهوم	عبارت ریاضی	pythin
+	جمع دو عدد	$x + y$	x+y
-	عدد اول منهای عدد دوم	$x - y$	x-y
*	ضرب دو عدد	xy	x*y
/	عدد اول تقسیم بر عدد دوم	$\frac{x}{y}$ $x \div y$ x/y	x/y
%	باقی مانده تقسیم عدد اول بر دوم	$x \bmod y$	x%y
**	توان	x^y	x**y
//	باقی مانده صحیح (خارج قسمت بدون اعشار)	x div y	x//y

عملگر	عبارت نمونه	عبارت معادل	عملگر	عبارت نمونه	عبارت معادل
=	x=a	x=a	/=	x/=a	x=x/a
+=	x+=a	x=x+a	%=	x%=a	x=x%a
-=	x-=a	x=x-a	**=	x**=a	x=x**a
*=	x*=a	x=x*a	//=	x//=a	x=x//a

اگر شما واقعا یک مسئله را به خوبی درک کنید

پاسخ مسئله از درون آن استخراج میشود

زیرا پاسخ جدای از مسئله نیست

Krishnamurti

سوال هوش:

فردی ساعت 7:30 AM از چادر خود خارج شده ابتدا 10 کیلومتر به سمت جنوب حرکت میکند. سپس 20 کیلومتر به سمت شرق رفته سپس 10 کیلومتر به سمت جنوب رفته و 1 ساعت استراحت میکند. پس از صرف نهار 10 کیلومتر به سمت غرب رفته سپس 20 کیلومتر به سمت شمال رفته و به چادر خود میرسد. در چادرش یک خرس میبیند. خرس چه رنگی است؟

راهنمایی: پاسخ سوال را با دقت به مسیر طی شده پیدا خواهید کرد.

چند برنامه ساده

یک عدد گرفته آن را چاپ کند.

```
a=int(input('a:'))
print (a)
```

یک عدد اعشاری گرفته آن را با دقت 4 رقم اعشار چاپ کند.

```
a=float(input('a:'))
print (f"{a:.4}")
```

a	خروجی
15.2134947	15.2135

یک عدد گرفته دو برابرآن را چاپ کند.

```
a=int(input('a:'))
b=a*2
print (b)
```

روش دوم

```
a=int(input('a:'))
print (2*a)
```

دو عدد گرفته حاصلجمع آنها را چاپ کند.

```
a=int(input('a:'))
b=int(input('b:'))
c=a+b
print (c)
```

روش دوم

```
a=int(input('a:'))
b=int(input('b:'))
print (a+b)
```

اولویت عملگر ها

گاهی ترتیب اعمال عملگرها ممکن است باعث پاسخهای متفاوتی گردد.

جدول زیر برای اولویت عملگر ها مشخص گردید است(عملگر های در یک سطح، از نظر اولویت با هم برابرند)

توان	**
ضرب و تقسیم	% * /
جمع و تفریق	+ -

بالاترین اولویت

پایین ترین اولویت

قوانین اولویت عملگر ها

الف)بین دو عملگر با اولویت متفاوت انکه اولویتش بالاتر است ابتدا اعمال میشود.

ب) بین دو عملگر هم اولویت آنکه در سمت چپ عبارت قرار دارد ابتدا اعمال میشود.

اگر بخواهید بر خلاف اولویت جدول عملگر ها عملگری زودتر اعمال شود از پرانتز گذاری استفاده میکنیم.

مثال: در عملیات (a+b)/2 اولویت جمع کمتراز تقسیم است اما باپرانتزگذاری جمع زودتر اعمال میشود

نکته: برای نوشتن عبارات ریاضی کافی است کل صورت را در یک پرانتز تقسیم بر کل مخرج در یک پرانتز کرد (مخرج) / (صورت) . برای توان هم، پایه و توان اگر فرمول بود هرکدام را در یک پرانتز قرار میدهیم.

در سایر موارد همان شکل ریاضیاتی فرمول معمولا بدون پرانتز گذاری استفاده میشود.

مثال: (x+y)**((5*x^3-3*x) / (2*a)) $(x+y)^{\frac{5x^3-3x}{2a}}$ (5*y-3*x) / (2*a) $\frac{5y-3x}{2a}$

با توجه به اولویت ها نوشتن بسیاری از فرمول ها مثل چند جمله ای ها نیازی به پرانتز گذاری ندارد مگر آنکه در فرمول ریاضی آن پرانتز باشد

برای محاسبه دستی و تشخیص چگونگی عملکرد یک فرمول کافیست ابتدا از چپ به راست تمام توانها را اعمال کنیم در مرحله بعد از چپ به راست تمام ضرب و تقسیم ها و در مرحله سوم از چپ به راست تمام جمع و تفریق ها را اعمال کنیم و نتیجه نهایی همان نتیجه برنامه خواهد بود

برنامه مبتنی بر فرمول

طول و عرض مستطیل را گرفته و مساحت آن را چاپ کند.

```
tol=int(input('tol:'))
arz=int(input('arz:'))
s=tol*arz
print(s)
```

روش دوم

```
tol=int(input('tol:'))
arz=int(input('arz:'))
print(tol*arz)
```

شعاع دایره را گرفته
مساحت آن را چاپ نماید.

```
r=int(input('r:'))
s=r**2*3.14
print (s)
```

میخواهیم دوریک استخرمستطیل شکل نرده بکشیم.
برنامه ای بنویسیدکه طول نرده موردنیازراچاپ کند

```
tol=int(input('tol:'))
arz=int(input('arz:'))
p=(tol+arz)*2
print (p)
```

راهنمایی:
-مسئله محاسبه محیط مستطیل است
-بحث اولویت عملگر ها در فرمول محیط

برنامه مبتنی بر فرمول (۲)

مبلغ کالا و درصد تخفیف را گرفته مبلغ پرداختی راچاپ کند.

gh قیمت
t تخفیف
mt مبلغ تخفیف

```
gh=int(input('ghymat kala:'))
t=int(input('darsad takhfif:'))
mt=gh*(t/100)
print (gh-mt)
```

```
gh=int(input('ghymat kala:'))
t=int(input('darsad takhfif:'))
print (gh*((100-t)/100) )
```

ضرایب معادله درجه دو را گرفته و ریشه های آن را بدست آورد(فرض کنید دلتا بزرگتر از صفر می باشد)

تحلیل مسئله:

ابتدا باید ضرایب معادله (a,b,c) را گرفت

دلتا را بر اساس فرمول مقابل به دست می آوریم $\Delta = b^2 - 4ac$

با فرض مثبت بودن Δ **دو ریشه داریم طبق فرمولهای زیر**

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

چاپ نتایج

```
import math
a=int(input('a:'))
b=int(input('b:'))
c=int(input('c:'))
delta=b**2-4*a*c
x1=(-b+math.sqrt(delta))/(2*a)
x2=(-b-math.sqrt(delta))/(2*a)
print (f"x1={x1:.5}")
print (f"x2={x2:.5}")
```

تابع رادیکال : math.sqrt(x) برای استفاده لازم است کتابخانه math را import کنیم

برنامه مبتنی بر فرمول (۳)

تعداد روز را گرفته مشخص کند چند ماه، هفته و روز است

از انجایی که یک ماه تعداد دقیقی از یک هفته نیست (۴ هفته و دو روز میشود یک ماه) لذا روش اول صفحه قبل قابل استفاده نیست. اما همه مضربی از روز هستند لذا به روش دوم یعنی از بالا به سمت پایین حساب میکنیم یعنی اول ماه بعد هفته و هر چه ماند می شود روز

```
a=int(input('a:'))
m=a // 30;
a=a % 30;
h=a // 7;
r=a % 7;
print(m , h , r)
```

a	m	h	r
259	8		
19		2	
			5

مکان فعلی و سرعت خودرو را گرفته و با گرفتن زمان، مکان نهایی خودرو را مشخص کند.

تحلیل مسئله:
ابتدا باید ورودی ها را دریافت نمود

مکان نهایی خودرو بر اساس فرمول زیر محاسبه میشود

$$x = vt + x_0$$

چاپ نتایج

```
x0=int(input('x0:'))
v=int(input('v:'))
t=int(input('t:'))
x=v*t+x0
print ('x:',x)
```

برنامه مبتنی بر فرمول (۴)

تعداد ثانیه را گرفته مشخص کند چند روز، ساعت،دقیقه و ثانیه است

```
a=int(input('a:'))
s=a % 60;
a=a //60;
m=a % 60;
a=a // 60;
h=a % 24;
d= a // 24;
print( d ,h ,m ,s)
```

هر 60 ثانیه میشود 1 دقیقه لذا هرچه باقی بماند میشود ثانیه های اضافی ما

هر 60 ثانیه میشود 1 دقیقه پس کل دقایق ما میشود. از این لحظه a دقایق است

هر 60 دقیقه میشود 1 ساعت لذا هرچه باقی بماند میشود دقایق اضافی ما

هر 60 دقیقه میشود 1 ساعت پس کل ساعات ما میشود. از این لحظه a ساعات است

هر 24 ساعت میشود 1 روز لذا هرچه باقی بماند میشود ساعات اضافی ما

هر 24 ساعت میشود 1 روز پس کل روزهای ما میشود.

a	s	m	h	d
523614	54			
8726		26		
145			1	
				6

روش اول: چون واحد ها همه مضرب دقیقی از هم هستند (هر ۶۰ ثانیه یک دقیق–هر ۶۰ دقیقه یک ساعت – هر ۲۴ ساعت یک روز) عملا همه مضربی از واحد های کوچکتر هستند، لذا **باقی مانده تقسیم** بر 60 میشود ثانیه هایی که نمیتوانند به واحد بزرگتر از خود یعنی دقیقه تبدیل شوند و **خارج قسمت** تقسیم میشود **تبدیل هر ۶۰ ثانیه به دقیقه**.

روش دوم: از بالا به پایین محاسبه میکنیم یعنی میدانیم هر روز چند ثانیه می شود (24*60*60=86400) لذا ابتدا **خارج قسمت تقسیم** بر این عدد میشود تعداد روز و **باقی مانده تقسیم** میشود ثانیه هایی که باید تبدیل به ساعت، دقیقه و ثانیه شوند. حال همین کار را برای تبدیل ثانیه های باقی مانده به واحد های ساعت و دقیقه و ثانیه میکنیم

```
a=int(input('a:'));
d=a // (24*60*60);
a=a %(24*60*60);
h=a // (60*60);
a=a % (60*60);
m=a // (60);
s= a% (60);
print( d ,h ,m ,s)
```

هر روز 24*60*60 ثانیه است. هر تعداد از این ثانیه میشود روز

مقداری که باقی بماند میشود ثانیه هایی که کمتر از روز (a)

هر ساعت 60*60 ثانیه است. هر تعداد از این ثانیه میشود ساعت

مقداری که باقی بماند میشود ثانیه هایی که کمتر از ساعت (a)

هر دقیقه 60 ثانیه است. هر تعداد از این ثانیه میشود دقیقه

مقداری که باقی بماند میشود ثانیه هایی که کمتر از دقیقه است

a	s	m	h	d
523614				6
5214			1	
1614		26		
	54			

برنامه مبتنی بر فرمول (۵)

مسئله	فرمول مسئله	کد قسمت اصلی برنامه
تبدیل درجه (C) به فارنهایت (F)	$F = \frac{9}{5}C + 32$	c=int(input('c:')) f=9/5*c+32 print(f)
تبدیل فارنهایت (F) به درجه (C)	$C = \frac{5}{9}(F - 32)$	f=int(input('f:')) c=5/9.0*(f-32) print(c)
محاسبه بیمه B (۵٪) و مالیات M(۱۰٪) مربوط به حقوق H	B=0.05*H M=0.1*H	c=int(input('h:')) b=0.05*h m=0.1*h print(b,m)
محاسبه وتر (C) مثلث قائم الزاویه با داشتن دو ضلع A و B	$C = \sqrt{A^2 + B^2}$	A=int(input('A:')) B=int(input('B:')) C=math.sqrt(A*A+B*B) print(C)
محاسبه ضلع مقابل(A) به زاویه θ در مثلث قائم الزاویه با وتر (C) دقت کنید که در برنامه متغیری به نام θ نداریم چون طبق قواعد، نام متغیر میتواند ترکیبی از حروف انگلیسی، ارقام و _ باشد لذا در باید نام مجاز در نظر بگیرید مثلا teta یا t یا هر نام مجازی	A=C*sin(θ)	c=int(input('c:')) teta=int(input('teta:')) A=c*math.sin(teta) print(A)

سه عدد گرفته میانگین اعداد را چاپ نماید.

$$\text{میانگین} = \frac{a+b+c}{3}$$

```
a=int(input('a:'))
b=int(input('b:'))
c=int(input('c:'))
sum=a+b+c
avg=sum/3
print(avg)
```

```
a=int(input('a:'))
b=int(input('b:'))
c=int(input('c:'))
avg=(a+b+c)/3
print(avg)
```

```
a=int(input('a:'))
b=int(input('b:'))
c=int(input('c:'))
print((a+b+c)/3)
```

اگر صورت را در پرانتز نگذاریم ابتدا عدد سوم تقسیم بر 3 می شود بعد با دو عدد اول جمع (بحث اولویت عملگرها)در حالی که قرار است ابتدا سه عدد جمع شوند بعد حاصل بر 3 تقسیم شود.

برنامه هایی که جواب آنها پاسخ فوق است(سه عدد گرفته میانگین اعداد را چاپ نماید.)

میانگین سنی سه نفر را چاپ کند.
متوسط بارندگی در سه روز را چاپ کند.
میانگین حقوق دریافتی سه استاد دانشگاه
معدل نمرات دانش آموزی که سه درس دارد.
متوسط یا میانگین سه عدد جوابی مشابه مسئله فوق دارد

متوسط متراژ سه ویلا
متوسط شیر سه گاو
متوسط وزن سه فیل
میانگین وزن سه نان

دو متغیر گرفته و محتوای آن دو ر ا جابجا کند.

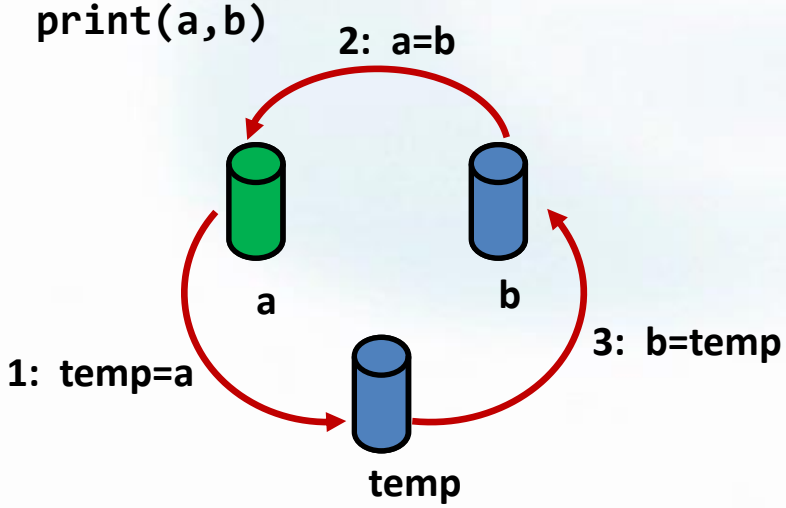
```
a=int(input('a:'))
b=int(input('b:'))
a,b=b,a
print(a,b)
```

پایتون این دستور را برای این کار دارد

روش کلاسیک:

فرض کنید محتوی یک لیوان a و لیوان b را قرار است جا بجا کنیم. برای این کار نیاز به لیوان سوم داریم نام آن را temp میگذاریم و به شکل زیر کار را انجام میدهیم

```
a=int(input('a:'))
b=int(input('b:'))
temp=a
a=b
b=temp
print(a,b)
```



نمره و تعداد واحد سه درس یک دانشجو را گرفته معدل او را حساب کند

فرمول میانگین وزنی

$$avg = \frac{\sum_{i=1}^{n}(w_i * grade_i)}{\sum_{i=1}^{n} w_i}$$

معدل ۳ درس (واحدی)

$$avg = \frac{vahed_1 * num_1 + vahed_2 * num_2 + vahed_3 * num_3}{vahed_1 + vahed_2 + vahed_3}$$

```
num1=float(input('num1:'))
v1=int(input('vahed1:'))
num2=float(input('num2:'))
v2=int(input('vahed2:'))
num3=float(input('num3:'))
v3=int(input('vahed3:'))
print((num1*v1+num2*v2+num3*v3)/(v1+v2+v3))
```

```
sorat=num1*v1+num2*v2+num3*v3
makhraj=v1+v2+v3;
print(sorat/makhraj)
```

معادل دستور بالا