

خلاصه هفته سوم

مفاهیم:

فلوچارت، آشنایی مقدماتی با پایتون، انواع داده‌های اصلی در پایتون

در این هفته (برای یادآوری) برای چند نمونه مسئله دیگر الگوریتم ارائه می‌دهیم. همچنین با طرز نمایش الگوریتمها با استفاده از فلوچارت آشنا می‌شویم. سپس به سراغ زبان پایتون می‌رویم و بطور مقدماتی با شیوه اجرای برنامه‌های پایتون و محیط برنامه‌نویسی پایتون آشنا می‌شویم. در انتها به معرفی چند نوع داده اصلی در پایتون می‌پردازیم.

۱ فلوچارت

همانطور که در اثباتهای ریاضی بعضی اوقات از شکل و نمودار استفاده می‌شود، برای نمایش بهتر و سریعتر طرز کار یک الگوریتم، می‌توان از شکل و نمودار استفاده کرد. برای این منظور از زبان نمایشی فلوچارت استفاده می‌شود. فلوچارت به طراح الگوریتم کمک می‌کند تا راه حلش ایده‌هایش را با استفاده از اشکال هندسی و خطوط جهتدار به نمایش بگذارد. با داشتن فلوچارت یک الگوریتم، برنامه‌نویس سریعتر می‌تواند ایده و منطق راه حل مسئله را به یک برنامه کامپیوتری تبدیل کند.

۱.۱ چند قرارداد برای طراحی فلوچارت

- برای نمایش شروع و پایان الگوریتم از یک بیضی استفاده می‌شود.

شروع

start

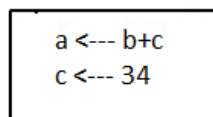
پایان

end

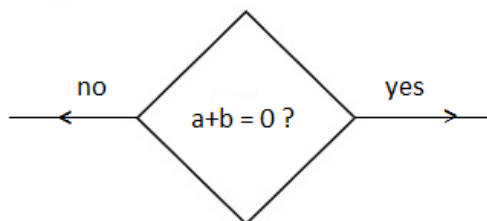
- برای گرفتن مقدار از ورودی و چاپ در خروجی از متوازی الاضلاع استفاده می‌شود.

n را بخوان

n را چاپ کن

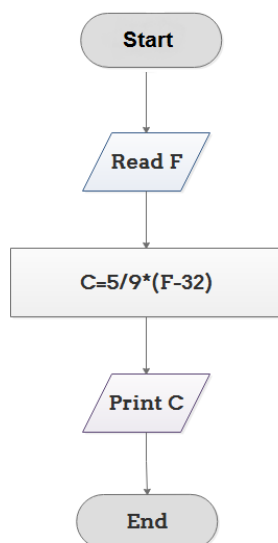


- برای نمایش انجام محاسبات و بدنه برنامه از مستطیل استفاده می‌شود.
- برای نشان دادن تصمیم و عملگر شرطی از لوزی استفاده می‌شود

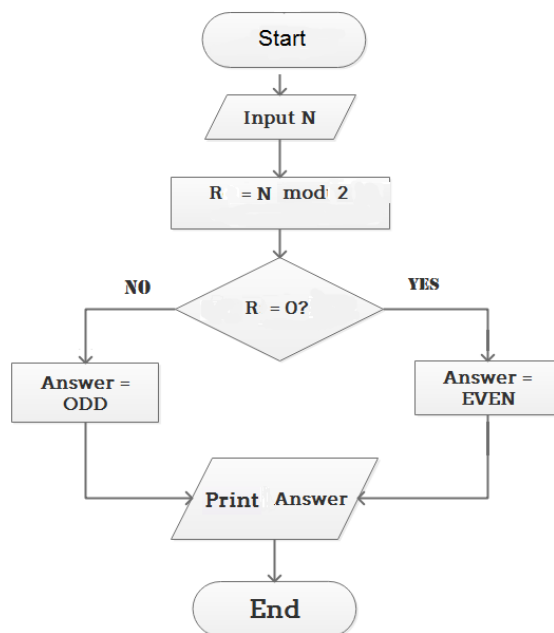


۲.۱ چند مثال

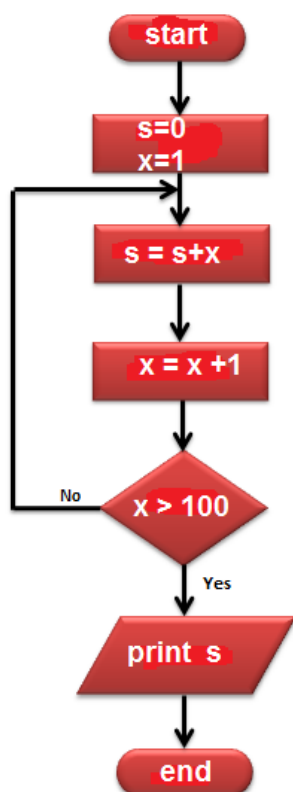
فلوچارت زیر الگوریتمی را نمایش می‌دهد که درجه فارنهایت را به سلسیوس تبدیل می‌کند.



فلوچارت زیر الگوریتمی را نمایش می‌دهد که عددی را خوانده و در صورتی که زوج باشد چاپ می‌کند
 EVEN و در صورتی که فرد باشد چاپ می‌کند ODD



فلوچارت زیر الگوریتمی را نمایش می‌دهد که مجموع اعداد بین ۱ تا ۱۰۰ را چاپ می‌کند.



۲ زبان پایتون

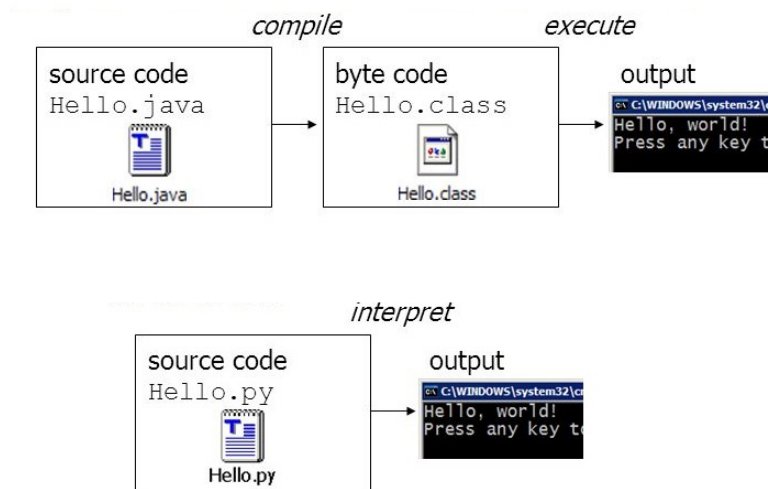
پایتون python یک زبان برنامه‌نویسی است که توسط خیدو وان روسوم Guido Van Rossom در سال ۱۹۹۱ طراحی شده است. در زیر چند ویژگی این زبان را ذکر می‌کنیم.

۱. ویژگی اصلی پایتون خوانایی و سادگی استفاده از آن است. مخصوصاً نسبت به زبانهای برنامه‌نویسی مثل C++ یا جاوا، یادگیری پایتون آسانتر و سریعتر است. به همین دلیل، امروزه در خیلی از محیطهای دانشگاهی و تجاری از پایتون برای آموزش برنامه‌نویسی و تولید سریع نرم افزار استفاده می‌شود.

۲. به دلیل وجود دستورات سطح بالا در پایتون، غالباً کد پایتون نسبت به کد زبانهای دیگر کوتاهتر و قابل فهمتر است.

۳. متغیرها در زبان پایتون نیازی به تعیین نوع ندارند. در اکثر زبانهای برنامه‌نویسی متغیرها باید نوع داشته باشند و نوع آنها باید در برنامه صریحاً ذکر شود. (برای مثال از نوع عدد صحیح هستند یا عدد اعشاری). در پایتون مقداری که به متغیر داده می‌شود نوع متغیر را تعیین می‌کند که ممکن است در طول برنامه تغییر کند. پس نوع داده متغیر در زبان پایتون می‌تواند همگام با محتوای آن تغییر کند.

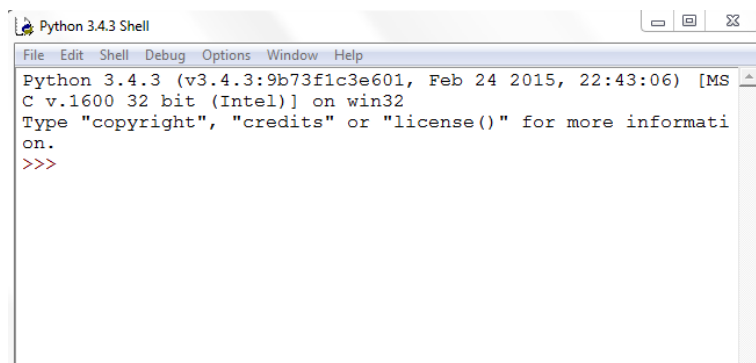
۴. پایتون یک زبان تفسیر شده interpreted است. این بدین معنی است که نرم‌افزاری به نام مفسر interpreter وجود دارد که کد پایتون را دریافت کرده و آن را اجرا کرده و مستقیماً خروجی اش را تولید می‌کند. برخلاف پایتون، برنامه‌هایی که با زبانهایی مثل C++، فورترن Fortran و یا جاوا نوشته می‌شوند ابتدا توسط یک نرم‌افزار به که آن کامپایلر compiler گفته می‌شود تبدیل به کد ماشین (یا یک کد میانی دیگر) شده و سپس کد تولید شده توسط کامپایلر اجرا می‌شود. در حالیکه مفسر پایتون مستقیماً کد پایتون را اجرا کرده و خروجی آن را تولید می‌کند.



به دلیل سطح بالا بودن زبان و انعطاف زیاد، اجرای برنامه‌های پایتون در مقایسه با اجرای برنامه‌های زبانهای دیگر مثل C++ کندتر است. به همین دلیل در موقعیتهایی که به اجرای بسیار سریع نیاز داریم، بهتر است برای برنامه‌نویسی از زبانی دیگر استفاده شود. با این وجود به دلیل راحتی و انعطاف زیاد آن، در خیلی از موقعیتهای پایتون برای تولید نرم افزار استفاده می‌شود.

۱.۲ اجرای برنامه پایتون

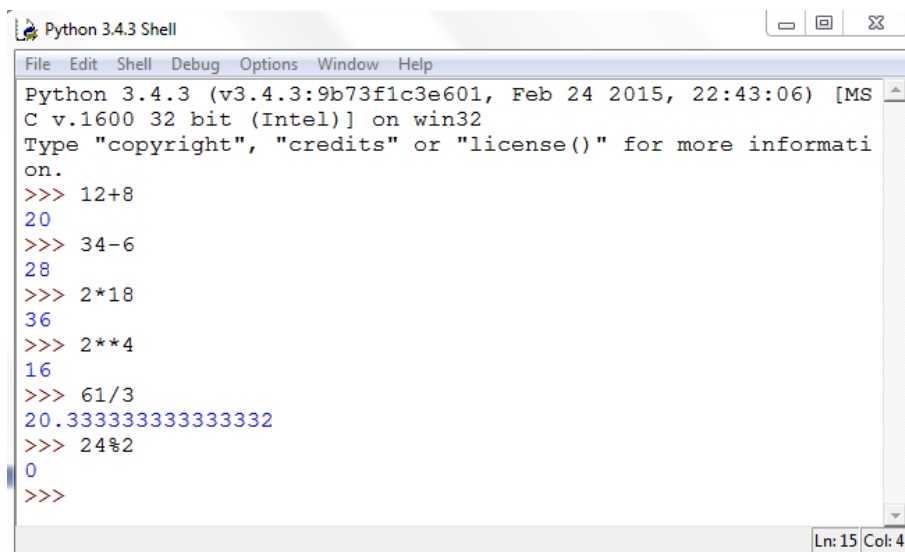
همانطور که گفته شد، کد پایتون توسط نرم‌افزاری به نام مفسر پایتون اجرا می‌شود. برنامه مفسر پایتون در سکوها‌ی نرم‌افزاری متنوع (مثل ویندوز، لینوکس، مک) موجود است. البته در سیستم عامل ویندوز، این برنامه بطور پیش فرض وجود ندارد. به همین دلیل کاربر باید آن را از جایی دیگر تهیه کرده و نصب کند. وبسایت python.org آخرین نسخه‌های پایتون را ارائه می‌کند. با مراجعه به این سایت می‌توان جدیدترین نسخه پایتون را دانلود کرد. در این درس ما از نسخه پایتون 3.7 به بالا استفاده می‌کنیم. بعد از نصب پایتون در ویندوز، برنامه مفسر پایتون از طریق فایل python.exe که در پوشه اصلی برنامه نصب شده قرار دارد، قابل اجراست. با اجرای این فایل، محیط اجرای دستورات پایتون قابل مشاهده است.



شکل ۱: محیط مفسر پایتون در سیستم عامل ویندوز

به طور کلی به دو شیوه می‌توان از مفسر پایتون استفاده کرد.

- **حالت تعاملی Interactive Mode** در این حالت کاربر بصورت تعاملی با مفسر پایتون کار می‌کند. کاربر یک دستور (یا عبارت) را به برنامه می‌دهد و مفسر آن را اجرا کرده و نتیجه را برمی‌گرداند. در این حالت فقط می‌توان یک عبارت یا دستور را برای اجرا به مفسر فرستاد. به همین دلیل از حالت تعاملی بیشتر برای آموزش زبان و یا تست دستورات و عبارات ساده استفاده می‌شود. در زیر یک نمونه از کار با مفسر پایتون به صورت تعاملی نشان داده شده است.



- حالت برنامه مکتوب Script Mode این شیوهی برنامه نویسی سنتی و معمول است. در این حالت برنامه‌ای که با زبان پایتون نوشته شده و در فایل با پسوند .py ذخیره شده به مفسر پایتون داده می‌شود. مفسر پایتون کد مربوطه را اجرا کرده و خروجی برنامه را نشان می‌دهد.

```
C:\Python37\python example.py
```

۳ انواع داده اصلی در پایتون

چهار نوع داده اصلی در پایتون وجود دارد:

- عدد صحیح integer
- عدد اعشاری floating point
- مقدار منطقی true false
- رشته string

توجه: نوع داده‌های اصلی دیگری نیز در پایتون وجود دارند (مثل عدد موهومی) که در اینجا به آنها پرداخته نمی‌شود.

۱.۳ عدد صحیح

ما با اعداد صحیح بطور معمول در مبنای ۱۰ سروکار داریم ولی در کامپیوترها، اعداد در مبنای ۲ ذخیره می‌شوند. مبنای ۸ و ۱۶ هم (بالاخص مبنای ۱۶) بدلیل اینکه توانی از ۲ هستند و اندازه‌ی یک واحد از حافظه هستند استفاده فراوان دارند. اعداد در مبنای ۲ (اعداد باینری binary) در پایتون با پیشوند 0B شروع می‌شوند (می‌توان از حرف کوچک b نیز استفاده کرد).

```
>>> 0B1101
13
>>> 0b11110
30
```

اعداد در مبنای ۱۶ با پیشوند 0x شروع می‌شوند.

```
>>> 0x3A
52
>>> 0X11
17
```

تبدیل به مبنای دلخواه b

در هفته اول، با شیوه تبدیل مبنا از مبنای دلخواه b به مبنای ۱۰ آشنا شدیم. برای مثال عدد 2136 در مبنای 7 بصورت زیر تبدیل به مبنای 10 می‌شود.

$$(2136)_7 = 2 \times 7^3 + 1 \times 7^2 + 3 \times 7^1 + 6 \times 7^0 = 762$$

برای تبدیل از مبنای 10 به مبنای دلخواه b می‌توان از تقسیم پلکانی استفاده کرد. فرض کنید می‌خواهیم عدد n را به مبنای b ببریم. اگر n از b کمتر باشد، نمایش عدد n در مبنای b خود n خواهد بود. حال فرض کنید r باقیمانده تقسیم n بر b باشد.

$$n = mb + r$$

در این حالت r رقم آخر از سمت راست نمایش n در مبنای b خواهد بود. یعنی

$$n = (\dots r)_b$$

برای پیدا کردن رقم بعدی کافی است همین کار را با m تکرار کنیم. پس هر بار باقیمانده تقسیم را به انتهای سمت چپ اضافه می‌کنیم و خارج قسمت بدست آمده را جایگزین عدد قبلی می‌کنیم و عمل تقسیم را تکرار می‌کنیم.

یک مثال در شکل زیر نشان داده شده است. می‌خواهیم عدد 478 را به مبنای 7 ببریم.

$$\begin{array}{r} 478 \overline{) 7} \\ \underline{42} 68 \overline{) 7} \\ 58 63 \overline{) 9} 7 \overline{) 7} \\ 56 5 7 1 \\ 2 2 \end{array}$$

$$478 = (1252)_7$$

تبدیل باینری به مبنای ۸ و ۱۶

برای تبدیل یک عدد باینری به مبنای ۸ کافی است از سمت راست هر بار سه رقم جدا کنیم و معادلش را در مبنای ۸ بنویسیم. در صورتی که عدد باینری تعداد ارقامش مضربی از سه نباشد، می‌توانیم به تعداد کافی صفر به انتهای سمت چپ اضافه کنیم. برای مثال

$$(01110111001)_2 = (\underbrace{001}_{07} \underbrace{110}_{6} \underbrace{111}_{7} \underbrace{001}_{1})_8 = (1671)_8$$

برای تبدیل باینری به مبنای ۱۶ مانند تبدیل به مبنای ۸ عمل می‌کنیم با این تفاوت که اینجا، هر بار ۴ رقم جدا می‌کنیم.

$$(01110111001)_2 = (\underbrace{0011}_{3B} \underbrace{1011}_{B} \underbrace{1001}_{9})_{16} = (3B9)_{16}$$

نکته: در پایتون برای بزرگترین عدد صحیحی که می‌توان ذخیره کرد محدودیتی وجود ندارد. اکثر زبانهای برنامه‌نویسی برای بزرگی اعداد محدودیتهایی قائلند. برای مثال عدد صحیح از نوع `int` در زبان `C++` نمی‌تواند

بزرگتر از ۲۱۴۷۴۸۳۶۴۷ باشد چون بطور معمول تنها چهار بایت برای عدد صحیح در نظر گرفته شده است. در پایتون چنین محدودیتی وجود ندارد.

۲.۳ عدد اعشاری

اعداد اعشاری (اعدادی که ممیز دارند) هم می‌توانند در مبناهای مختلف نمایش داده شوند. برای مثال عدد زیر، یک عدد باینری اعشاری است.

$$(11101.011)_2 = (29.375)_{10}$$

برای تبدیل عدد اعشاری در مبنای b به مبنای 10 مانند اعداد صحیح عمل می‌کنیم با این تفاوت که اعداد سمت راست ممیز را در توانهای منفی b ضرب می‌کنیم. از توان -1 شروع می‌کنیم.

$$(0.101)_2 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 0.625$$

$$(0.4F)_{16} = 4 \times 16^{-1} + 15 \times 16^{-2} = 0.30859375$$

تبدیل عدد اعشاری در مبنای ۱۰ به مبنای ۲

فرض می‌کنیم عدد داده شده کمتر از یک است (تبدیل قسمت صحیح مانند قبل است). برای تبدیل قسمت اعشاری، عدد مورد نظر را بطور متوالی در 2 ضرب می‌کنیم و قسمت صحیح را ثبت می‌کنیم. اگر قسمت صحیح 1 بود، مرحله بعدی را با قسمت اعشاری ادامه می‌دهیم. مثالهای زیر روش کار را نشان می‌دهند.

$0.625 \times 2 = 1.25$	$\overline{1}$	Generate 1 and continue with the rest.
$0.25 \times 2 = 0.5$	$\overline{0}$	Generate 0 and continue.
$0.5 \times 2 = 1.0$	$\overline{1}$	Generate 1 and nothing remains.

$$0.625_{10} = 0.101_2$$

$0.40625 \times 2 = 0.8125$	$\overline{0}$	Generate 0 and continue.
$0.8125 \times 2 = 1.625$	$\overline{1}$	Generate 1 and continue with the rest.
$0.625 \times 2 = 1.25$	$\overline{1}$	Generate 1 and continue with the rest.
$0.25 \times 2 = 0.5$	$\overline{0}$	Generate 0 and continue.
$0.5 \times 2 = 1.0$	$\overline{1}$	Generate 1 and nothing remains.

$$0.40625_{10} = 0.01101_2$$

در مثال زیر پروسه‌ی ضرب هیچگاه متوقف نمی‌شود چون عدد 0.8 دوباره تولید می‌شود و در یک دور می‌افتد. دلیل این مسئله این است که عدد 0.7 معادلش در مبنای 2 بینهایت رقم دارد.

$0.7 \times 2 = 1.4$	$\overline{1}$	<i>Generate 1 and continue with the rest.</i>
$0.4 \times 2 = 0.8$	$\overline{0}$	<i>Generate 0 and continue.</i>
$0.8 \times 2 = 1.6$	$\overline{1}$	<i>Generate 1 and continue with the rest.</i>
$0.6 \times 2 = 1.2$	$\overline{1}$	<i>Generate 1 and continue with the rest.</i>
$0.2 \times 2 = 0.4$	$\overline{0}$	<i>Generate 0 and continue.</i>
$0.4 \times 2 = 0.8$	$\overline{0}$	<i>Generate 0 and continue.</i>
$0.8 \times 2 = 1.6$	$\overline{1}$	<i>Generate 1 and continue with the rest.</i>
$0.6 \times 2 = 1.2$	$\overline{1}$	<i>Generate 1 and continue with the rest.</i>
...		

$$0.7 = (0.101100110011...)_{2}$$

نتیجه: هر عدد اعشاری (در مبنای ۱۰) را نمی‌توان بصورت دقیق در کامپیوتر ذخیره کرد چون بعضی از اعداد اعشاری وقتی در مبنای ۲ نمایش داده شوند به بینهایت رقم نیاز دارند. پس اعداد اعشاری غالباً بصورت تقریبی در کامپیوتر ذخیره می‌شوند.