

بہ نام خدا

دیرستان علامہ حلّی (۱) دورہ اول

جزوہ درس کامپیوتر (کل سال)

مدرس: جناب روستا و سلطانی

تالیف: انجمن کالج کلاس ۷.۴ و ۷.۵ (گروہ نویسندگان جزوہ کلاس ۷.۴ و ۷.۵)

دیران گروہ کامپیوتر: بہراد کودزی، کیاوش فردوسی، علیرضا جعفری، آریں اخترکوان و امیرعباس یقظین

فہرست مطالب

فصل اول: درباره پایتون: (صفحه ۳ تا ۵)

فصل دوم: نوع داده ها: (صفحه ۶ تا ۷)

فصل سوم: متغیر: (صفحه ۸ تا ۱۰)

فصل چهارم: پرینت: (صفحه ۱۱ تا ۱۲)

فصل پنجم: ایف: (صفحه ۱۳ تا ۱۵)

فصل ششم: گرفتن ورودی از کاربر: (صفحه ۱۶ تا ۱۸)

فصل ہفتم: فور: (صفحه ۱۹ تا ۲۱)

فصل ہشتم: وایل: (صفحه ۲۲ تا ۲۳)

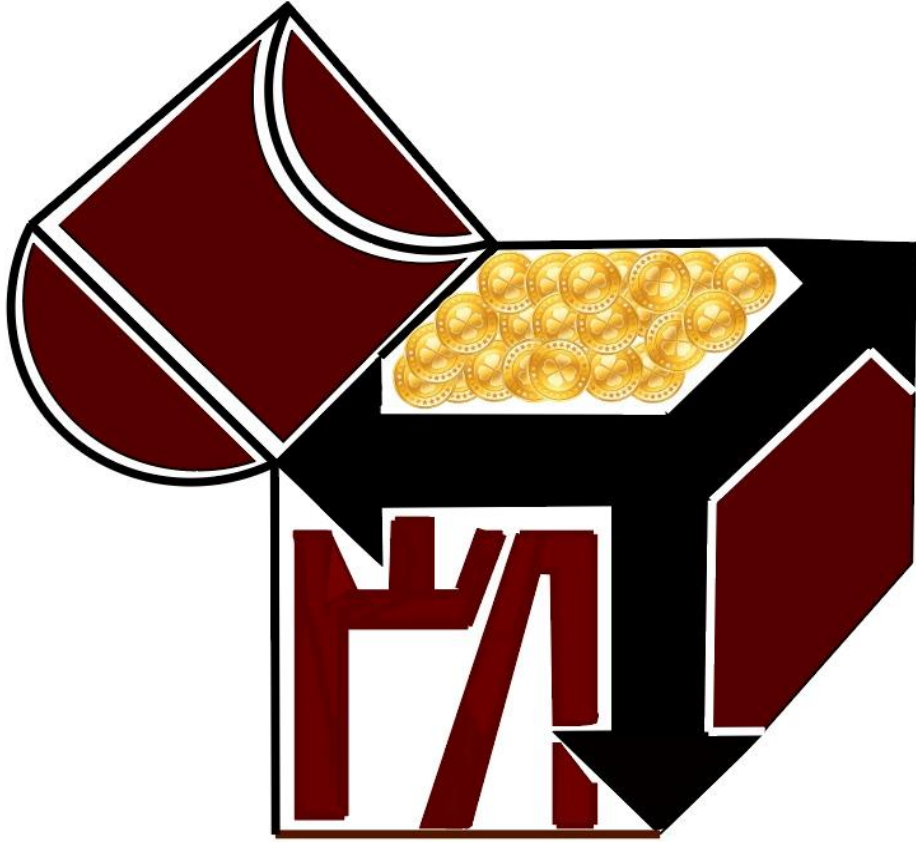
فصل نهم: ایمپوریت : (صفحه ۲۴)

فصل دهم: تریتل: (صفحه ۲۵ تا ۲۸)

فصل یازدهم: رندوم: (صفحه ۲۹)

فصل دوازدهم: شرط پیشرفته: (صفحه ۳۰)

فصل سیزدهم: تابع: (صفحه ۳۱ تا ۳۳)



فصل اول: درباره پایتون

پایتون یک زبان برنامه نویسی سطح بالا هست که توسط خیدو فان روسوم نوشته شده است. پایتون از C ساخته شده و حدود 31 سال است که وجود دارد.

این زبان منبع باز است به همین دلیل همه مردم جهان میتوانند به کد های منبع (source code) آن دسترسی داشته باشند.

در پایتون علائمی مانند {} یا ; اجباری نیستند و به جای آنها TAB برای هر قطعه کد باید زده شود.

تب بطور استاندارد در پایتون ۴ فاصله است.

پایتون case sensitive هست یعنی به اندازه حروف حساس هست.

در پایتون تمام دستورات و توابع پیش فرض
(Built-in functions) با حروف کوچک نوشته شده‌اند.



فصل دوم : نوع داده ها

در همه زبان های برنامه نویسی ما `data type` هایی داریم که اساس برنامه رو تشخیص میدهند.

پایتون شش `data type` وجود دارد که ما خواندیم.

۱- `int`: این نوع مقدار در واقع متشکل از تمام اعداد صحیح

هست. این کلمه مخفف `Integer` به معنای عدد صحیح

هست.

۲- `float`: این نوع مقدار متشکل از تمام اعداد اعشاری و صحیح

هست.

۳- `str`: این نوع مقدار در واقع یک رشته متنی داخل خود دارد و

در واقع یک متن هست.

۴- bool: مقادیر بول که مخفف Boolean هستند مقادیر منطقی هستند که اساس کل برنامه نویسی هستند. متشکل از True, False, None هستند.

۵- Keyword: کی ورد یا دستور اسم مقداری دستورات پایتون هست.

۶- function: فانکشن یا تابع نوع دیگری از دیتا تایپ های پایتون هستند که متشکل از تمام bif ها و تابع هایی هستند که خودمان مینویسیم.

فصل سوم: متغیر

متغیر:

متغیرها مثل ظرف‌هایی هستند که در آنها اعداد و ... ریخته می‌شوند که در طول برنامه با صدا کردن آن می‌توان از محتویات آن ظرف استفاده کرد.

به عمل ریختن اطلاعات در متغیر `Variable Assignment` گفته میشود

چند نکته درباره متغیرها:

۱- پایتون به حروف بزرگ حساس است برای مثال متغیری با نام `ali` با متغیری به نام `Ali` متفاوت است.

۲- شما نمی‌توانید نام متغیر را نام یک دستور بگذارید یعنی اسم متغیر نمی‌تواند `if` یا `def` و ... باشد.

۳- در طول برنامه متغیر می تواند جایگزین شود یعنی اگر متغیری
قبلا تعریف شده باشد و در ادامه دوباره تعریف شود مقدار تعریف
شده جدید در متغیر ذخیره می شود.

به این عمل Reassignment

۴- از عملگرهای ریاضی می توان برای تعریف کردن متغیرها
استفاده کرد یعنی می توان متغیری را $۲+۳$ تعریف کرد.
۵- برای تعریف کردن متغیرها می توانیم از متغیر کمک بگیریم.

عملگرهای ریاضی و کاربرد های آنها:

جمع (+): جمع کردن اعداد با هم. (add)

تفریق (-): کم کردن اعداد از هم. (subtract)

ضرب (*): ضرب کردن اعداد. (multiply)

تقسیم (/): تقسیم کردن اعداد. (divide)

توان (**): انجام عملیات توان برای اعداد. (power)

باقیمانده (%) : دادن باقیمانده تقسیم حاصل از دو عدد. (modulo)

خارج قسمت (//) : دادن خارج قسمت حاصل از تقسیم دو عدد. (absdiv or absolute divide)

نکته: در پایتون اولویت بین عملگرها وجود دارد و بین یک عدد و پرانتز باید یک عملگر ریاضی قرار بگیرد و در صورت نگذاشتن علائم بین اعداد و پرانتز (مانند $1(2+5)$) ، خطا می دهد.

عملگر	مثال	شکل ریاضی آن
=	$x = 5$	$x = 5$
+=	$x += 3$	$x = x + 3$
-=	$x -= 3$	$x = x - 3$
*=	$x *= 3$	$x = x * 3$
/=	$x /= 3$	$x = x / 3$
%=	$x %= 3$	$x = x \% 3$
//=	$x //= 3$	$x = x // 3$
**=	$x **= 3$	$x = x ** 3$
&=	$x \&= 3$	$x = x \& 3$
=	$x = 3$	$x = x 3$
^=	$x ^= 3$	$x = x ^ 3$
>>=	$x >>= 3$	$x = x >> 3$
<<=	$x <<= 3$	$x = x << 3$

فصل چهارم: چاپ (print)

پرینت:

پرینت در فارسی معادل کلمه چاپ است. احتمالاً همانطور که فهمیدید کار پرینت چاپ کردن خروجی برنامه است. پرینت پایتون به صورت زیر نوشته می شود و در بین دو پرانتز عباراتی که می خواهیم چاپ شود را می نویسیم.

print(x)

چند نکته درباره پرینت:

۱- برای فهماندن تفاوت بین متغیر و اعداد با حروف به پایتون می بایست از دو دابل کوتیشن (") و یا از دو کوتیشن (') در ابتدا و انتها حروف و کلمات استفاده کرد.

۲- اگر می خواهید که دو یا چند چیز را در یک خط پرینت، چاپ کنید می بایست از کاما(,) در بین آنها استفاده کرد.

۳- در پرینت می توان از عملگر های ریاضی استفاده کرد اما اگر آنها در بین کوتیشن باشند به همان شکل چاپ می شوند.

فصل پنجم: اگر (if)

شرط:

در واقع شرط یک دستور است که با نوشتن آن می توانید کاری کنید که اگر اتفاق مطلوب بیافتد دستورات زیر شرط اجرا شود. برای مثال برنامه زیر، برنامه ای است که اگر متغیر از صفر کوچکتر بود برنامه `adad manfi ast` را چاپ کند.

```
if -1 < 0:
```

```
    print("adad manfi ast")
```

چند نکته درباره شرط:

- ۱- بعد تعریف کردن شرط ما باید دستوراتی که می خواهیم با برقرار شدن شرط اجرا شوند را یک `tab` که معادل ۴ تا

space است از اول خط فاصله دهیم و حتما باید بعد از

تعریف کردن شرط از دو نقطه (:). استفاده کنیم.

۲- برای تعریف کردن شرط باید از علائم مقایسه ای استفاده

کنیم که به پایتون بفهمانیم در چه شرایطی شرط برقرار

است

علائم شرطی:

==: برابر بودن هر دو عدد.

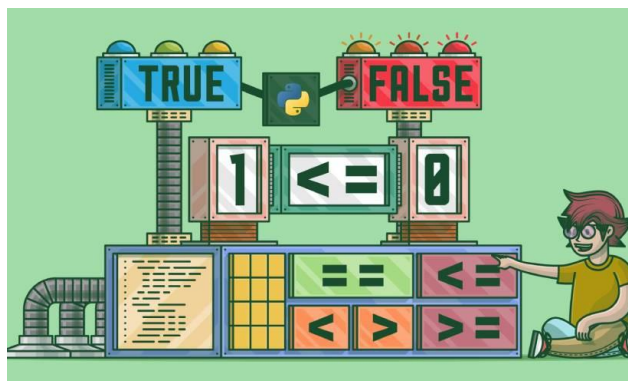
>: بزرگتر بودن عدد سمت چپ علامت.

<: بزرگتر بودن عدد سمت راست علامت.

>=: برابر بودن هر دو عدد یا بزرگتر بودن عدد سمت چپ.

<=: برابر بودن هر دو عدد یا بزرگتر بودن عدد سمت راست.

!=: برابر نبودن.



دستور else :

با این دستور شما می توانید کاری کنید که اگر if شما انجام نشد دستورات زیر else اجرا شوند.

چند نکته درباره else :

۱- مثل if باید بعد از else دو نقطه بگذاریم و دستورات زیر else را با فاصله یک tab از خط بنویسیم.

۲- جلوی else هیچ شرطی نباید بنویسید.

۳- باید else دقیقا زیر if مورد نظر باشد. (*اگر if از خط فاصله داشت باید else به همان مقدار از خط باید فاصله داشته باشد *)

دستور elif :

این دستور که مخفف else if هست اگر if بالای خود برقرار نبود شرط جلوی خود را چک میکند و اجرا میکند.

به مجموعه ۳ دستور بالا If statement گفته میشود.

فصل ششم: گرفتن ورودی از کاربر

برای اینکه بتوانید از کاربر مقدار یک متغیر را بگیرید باید از اینپوت (input) استفاده کنید.

چند نکته درباره اینپوت:

۱- شما باید اینپوت را حتما در یک متغیر بنویسید تا مقدار در آن ذخیره شود.

۲- شما به سه روش کلی می توانید اینپوت را بنویسید:

الف) به روشی که از کاربر فقط عدد صحیح بگیرد:

```
n=int(input())
```

ب) به روشی که از کاربر فقط عدد اعشاری و صحیح بگیرد:

```
n=float(input())
```


پ) به روشی که از کاربر هر چیزی بگیرد اما به عنوان رشته ذخیره کند یا به عبارت ساده تر عبارت شما به صورت یک متن است.

```
n=input()
```

۳- برای اینکه به کاربر بگوید در این اینپوت چه چیزی باید بنویسد شما می توانید از دو روش استفاده کنید:

روش اول:

varied	خروجی
<pre>print("yek adad vared kon:") n=int(input())</pre>	<pre>yek adad vared kon: 26</pre>

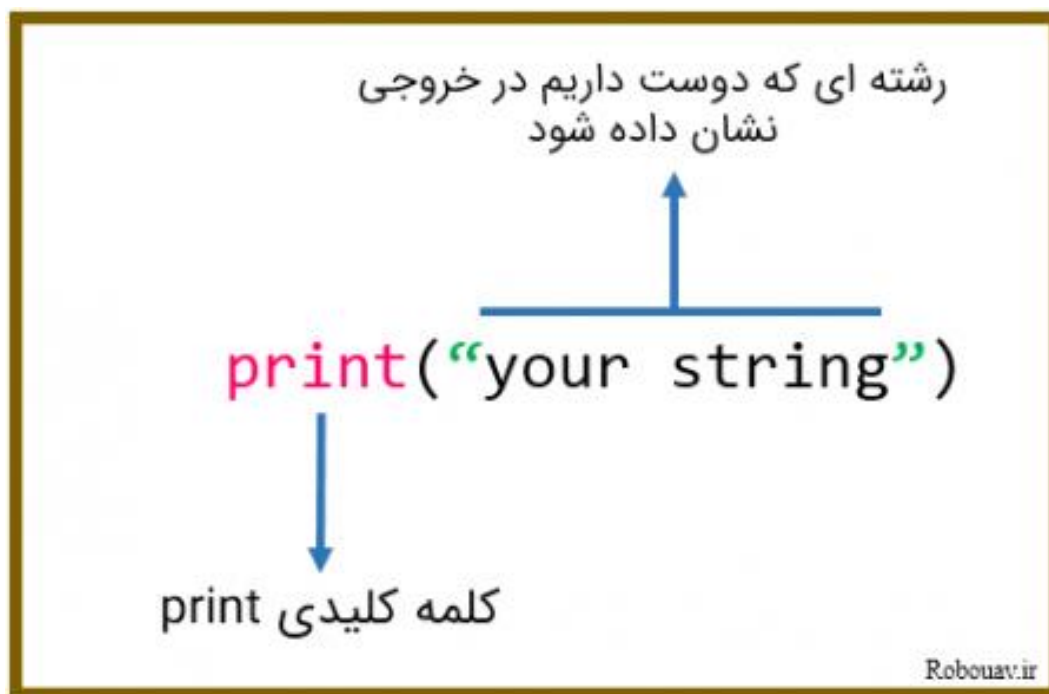
روش دوم:

ورودی	خروجی
-------	-------

```
n=int(input("yek adad  
vared kon:"))
```

```
yek adad vared  
kon:26
```

۴- دستور input یک تابع هست نه یک Keyword



فصل هفتم: دستور for

for:

در واقع for مثل یک حلقه تکرار شونده عمل می کند و تمام دستوراتی که در زیرش هستند را به تعدادی که برایش تعریف می کنیم تکرار می کند. برای اینکه بفهمیم چه مقداری برای تعداد تکرار در for تعریف شده باید اختلاف دو عدد درون پرانتز را پیدا کنیم. برای مثال برنامه زیر ده بار دستور زیر خودش را اجرا می کند:

```
for i in range(1,11):  
    print("salam")
```

چند نکته درباره for:

- ۱- در مثال قسمت قبل حرف **a** در واقع یک متغیر است که نامش چیز دیگری هم می تواند باشد. مقدار **a** در بار اول از عدد سمت چپ پرانتز شروع می شود سپس هر بار که برنامه تکرار می شود یک واحد بیشتر می شود و تا هنگامی که مقدار **a** برابر اختلاف اعداد درون پرانتز شود ادامه می یابد.
- ۲- متغیر تعریف شده در فور در مقدار داده شده به خود **iterate** میشود تا مقادیر به اتمام برسند.
- ۳- تابع **range()** که حداقل یک ورودی میگیرد یک بازه از عدد اول تا عدد آخر میسازد و به **for** پس میدهد.
- ۴- تابع رنج با یک ورودی میشود از ۰ تا ورودی داده شده.
- ۵- با دو ورودی میشه از ورودی اول تا دوم.

۶- با سه ورودی میشه از ورودی اول تا دوم و هربار به اندازه

ورودی سوم برو جلو مثلا خروجی کد زیر برابر است با

۸۶۴۲

```
for i in range(2,10,2):  
    print(i)
```

for *var* **in** *iterator* :

 *loop body*

فصل هشتم : وایل (while)

وایل:

وایل هم مثل for یک حلقه تکرار شونده است به تعداد دفعات برقرار بودن شرط جلوی وایل برنامه زیر آن تکرار می شود. یعنی شرط را چک میکند و اگر درست بود دستورات داخل خود را اجرا میکند سپس شرط را چک میکند و... تا زمانی که شرط برقرار است این عمل اتفاق می افتد.

چند نکته درباره وایل:

۱- مثل if در وایل هم باید جلوی وایل یک شرط بنویسید که در صورت برقرار بودن آن برنامه های زیر وایل اجرا می شوند. در ضمن باید دستورات زیر وایل به اندازه یک tab فاصله داشته باشند.

۲- وایل بسیار شبیه for است اگر می خواهید کار یک for را با وایل انجام دهید باید کار زیر را انجام دهید:

`i=n`

`while i<m:`

`.....`

`i=i+1`

(**نکته:** n همان عدد سمت چپ for و m عدد سمت راست for

است و امثل همان ا درون for است که می توانید نام آن را

خودتان تغییر دهید.)

while *condition* :



loop body

فصل نهم: ایمپورت (Import)

این دستور برای اضافه کردن یک کتابخانه (library) یا پکیج (package) به برنامه استفاده میشود.

وقتی پکیجی یا کتابخانه ای به برنامه اضافه شود ما میتوانیم از فانکشنها و کلاس های داخل برنامه استفاده کنیم.



فصل دهم: ترتل

ترتل:

ترتل یک کتابخانه پایتون است که با آن می توانید کارهای گرافیکی انجام دهید یا به عبارت دیگر ترتل شبیه کشیدن نقاشی در پایتون است.

چند نکته درباره ترتل:

۱- در ابتدا شما باید با دستور `import turtle` کتابخانه ترتل را صدا کنید تا بتوانید از آن استفاده کنید.

۲- ترتل دستورات بسیار زیادی دارد که هم اکنون با مهم ترین دستورات ترتل آشنا می شوید.

مهم ترین دستورات ترتل:

`turtle.forward(x)`

`turtle.fd(x)`

این دستور جلو رفتن است و X طول خط است.

`turtle.backward(x)`

`turtle.bk(x)`

این دستور عقب رفتن است و X طول خط است.

`turtle.right(x)`

`turtle.rt(x)`

این دستور ترتل را X درجه به سمت راست می چرخاند

`turtle.left(x)`

`turtle.lt(x)`

این دستور ترتل را X درجه به سمت چپ می چرخاند

`turtle.penup()`

`turtle.pu()`

این دستور قلم را نامرئی می کند.

turtle.pendown()
turtle.pd()

این دستور قلم را دوباره ظاهر می کند.

turtle.speed("fastest") ← تغییر سرعت ترتل

turtle.color("color") ← تغییر رنگ قلم ترتل

turtle.shape("square") ← کشیدن شکل

turtle.undo() ← Undo کردن آخرین دستور

turtle.pensize(4) ← تغییر ضخامت قلم

turtle.fillcolor("color") ← پر کردن شکل

از دستور turtle.begin_fill() در ابتدای کشیدن شکل و از

دستور turtle.end_fill() در پایان کشیدن شکل استفاده کنید تا

شکل به رنگ قلم شما در بیاید.

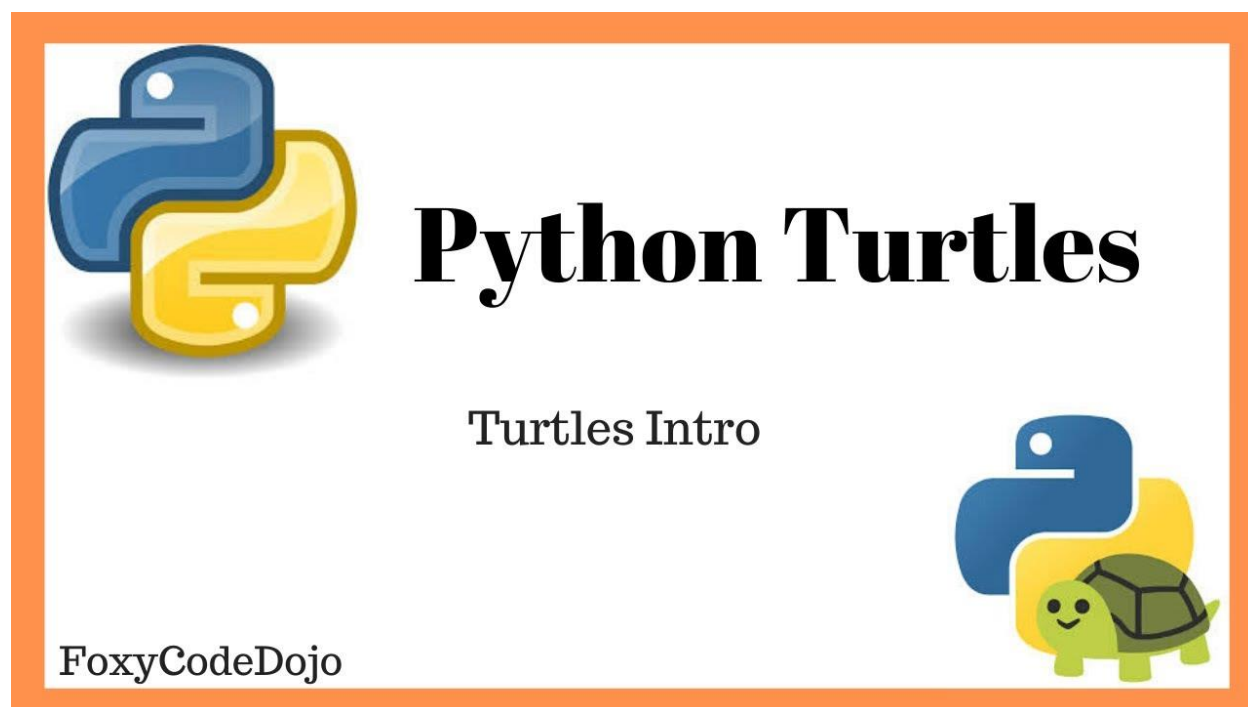
آموزش کشیدن خط منحنی:

خیلی ها می پرسند که چگونه باید در ترتل خط منحنی کشید در واقع در ترتل تعریف خط منحنی قسمتی جدا شده در دایره است و برای کشیدن آن باید از دستور زیر استفاده کنید:

`turtle.circle(r,t)`

(نکته: r شعاع دایره و t مقدار جدا شده از دایره بر حسب درجه

است.)



فصل یازدهم: رندوم

رندوم:

کتابخانه رندوم در پایتون برای کارهایی که به شانس ربط دارد ساخته شده است. به زبان ساده تر رندوم انتخاب تصادفی است.

دستورات رندوم:

`import random`: صدا زدن کتابخانه رندوم برای استفاده از

دستورات آن

`random.random()`: یک عدد تصادفی بین ۰ تا ۱ است که

همیشه با دقت چند اعشار است.

`random.randint(x,y)`: دادن یک عدد تصادفی از اعداد

بین x و y و خود x و y

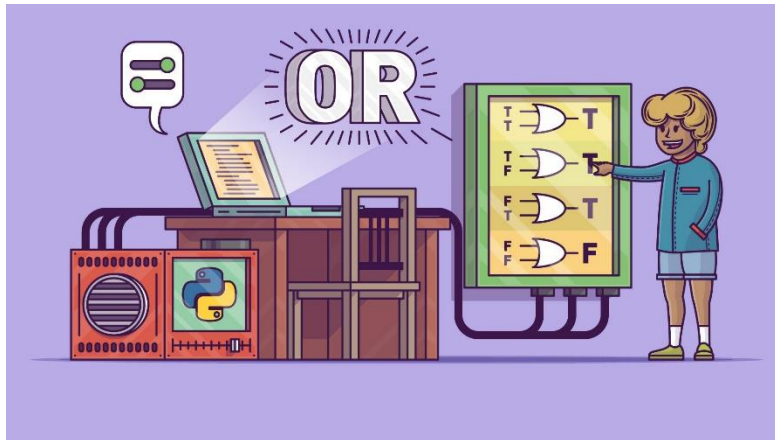
فصل دوازدهم: شرط پیشرفته

(**نکته:** هر شرطی که ما می نویسم یا True است یا False.)

دستورات شرط پیشرفته:

and (و): وقتی از and در if استفاده می کنیم باید خروجی شرط سمت چپ و راست and هر دو True باشد در غیر اینصورت False می شود.

Or (یا): وقتی از or استفاده می کنیم که با True بودن یکی از شرط های سمت راست یا چپ or ، بخواهیم جواب True از برنامه بگیریم.



فصل سیزدهم: تابع

تابع:

گاهی اوقات در برنامه ممکن است یک کد را چند بار تکرار کرده باشید. در تابع می توانید چند خط کد را ذخیره کنید و سپس با صدا زدن نام تابع در برنامه از آن استفاده کنید و دیگر از نوشتن دستور تکراری خسته نشوید.

چند نکته درباره تابع:

۱- در ابتدا باید برای تابع یک اسم بگذارید تا بتوانید آن تابع را در طول برنامه استفاده کنید و بعد اسم باید از پرانتز استفاده کنیم که داخل پرانتز می تواند نام چند متغیر باشد که در `def` از آن استفاده می کنیم و در انتها باید از دو نقطه استفاده کنیم و تمام دستورات زیر `def` را با فاصله یک `tab` از خط بنویسیم

۲- برای اینکه بتوانیم تابع را در یک متغیر ذخیره کنیم باید از دستور return استفاده کنیم تا آن متغیری که return کرده ایم را بتوانیم در یک متغیر تعریف کنیم.

۳- فرض کنید که می خواهید یک متن را در پایتون به نمایش در بیاورید اما سائز فونت متن های شما با یکدیگر متفاوت هستند برای اینکار شما نباید چند تابع تعریف کنید بلکه می توانید یک یا چند متغیر را به عنوان ورودی هنگام صدا کردن تابع به آن بدهید تا خروجی مناسب بگیرید.

۴- به جدول زیر توجه کنید:

ورودی	خروجی
a=10	10
def salam(meghdar): a=meghdar * 2 return a	20


```
b = salam(2)
print(a,b)
```

همانطور که متوجه شدید وقتی دستوری را در داخل `def` تعریف می کنید آن دستور فقط در داخل آن تابع تعریف می شود و متغیر های بیرون تابع را عوض نمی کند.

