

۱- تبدیل نوع داده ای

گاهی لازم است بین نوع داده های درون ساخته یا توکار پایتون، عملیات تبدیل انجام داد و نوعی را به نوع دیگر تبدیل نمود. کافی است اسم نوع را به عنوان تابع بکار ببرید.

توابع توکار متعددی در پایتون وجود دارد که عملیات تبدیل از نوعی به نوع دیگر را انجام می دهد.

این توابع شی جدیدی باز می گردانند که نشانگر مقدار تبدیل شده می باشد.

در جدول زیر چندتا از توابع پر کاربرد در پایتون را که وظیفه آنها تبدیل نوع داده ها می باشد را معرفی می کنیم:

تابع	شرح
<code>int(x)</code>	X را به عدد صحیح integer تبدیل می کند.
<code>long(x)</code>	X را به یک عدد صحیح بزرگ long integer تبدیل می کند.
<code>float(x)</code>	X را به یک عدد ممیز شناور تبدیل می کند.
<code>complex(real)</code>	یک عدد مختلط ایجاد می کند.
<code>str(x)</code>	شی X را به صورت یک رشته نمایش می دهد.
<code>tuple(s)</code>	به نوع tuple تبدیل می کند.
<code>list(s)</code>	به نوع داده ای list تبدیل می کند.
<code>set(s)</code>	به یک (set مجموعه) تبدیل می کند.
<code>dict(d)</code>	یک dictionary ایجاد می کند. d بایستی یک مجموعه ی پست سرهم یا دنباله ای از tuple ها (به صورت جفت کلید، مقدار) باشد.
<code>chr(x)</code>	یک integer را به کاراکتر تبدیل می کند.
<code>unichr(x)</code>	یک integer را به کاراکتر یونیکد تبدیل می کند.
<code>ord(x)</code>	یک تک کاراکتر را به معادل یا مقدار عدد صحیح (integer) آن تبدیل می کند.
<code>hex(x)</code>	یک عدد صحیح را به رشته ی شانزده شانزدهی تبدیل می کند.
<code>oct(x)</code>	یک عدد صحیح را به رشته ی هشت هشتی تبدیل می کند.
<code>Bin(x)</code>	یک عدد صحیح را به رشته ی دودویی یا باینری تبدیل می کند.

مثال:

```
num1 = int(input("Enter a number:"))  
print(bin(num1))  
print(oct(num1))  
print(hex(num1))  
print(chr(num1))
```

خروجی این دستورات:

```
Enter a number:67  
0b1000011  
0o103  
0x43  
C
```

فصل ۲

ساختارهای تصمیم و تکرار

در برنامه هایی که تاکنون نوشته ایم، دستورات به صورت پشت سرهم (از اولین دستور به آخرین دستور) اجرا می گردیدند. در برنامه های واقعی و پیچیده نیاز است بعضی از دستورات تحت شرایط خاصی اجرا شوند، و برخی دیگر از دستورات اجرا نشوند یا بعضی از دستورات چندین بار اجرا گردند. برای پیاده سازی چنین برنامه هایی از ساختارهای کنترلی استفاده می شود. ساختارهای کنترلی دو نوع هستند که عبارت اند از:

۱ - ساختارهای تصمیم گیری ۲ - ساختارهای تکرار

۲-۱ - ساختارهای تصمیم گیری

این ساختارها برای مواقعی به کار می روند که بخواهید با برقرار شدن شرط خاصی، مجموعه ای از دستورات اجرا شوند یا بعضی از دستورات دیگر اجرا نشوند. در ادامه ساختارهای تصمیم را می آموزیم.

ساختار تصمیم if

در این ساختار ابتدا **شرطی** ارزیابی می شود، اگر نتیجه ارزیابی شرط درست یا اصطلاحاً True باشد، یک مجموعه از دستورات اجرا می شوند، وگرنه، مجموعه دیگری از دستورات اجرا خواهند شد. این ساختار به صورت های زیر به کار می رود:

الف) ساختار ساده تک انتخابی: این دستور یک دستور مرکب است که به صورت زیر به کار می رود:

شرط if:
مجموعه دستورات

در این ساختار، ابتدا شرط ارزیابی می شود، اگر نتیجه ارزیابی شرط (True درست) باشد، **مجموعه دستورات** اجرا خواهند شد، در غیر این صورت، از اجرای **مجموعه** دستورات صرف نظر خواهد شد و اولین دستور بعد از if اجرا خواهد شد.

نکته ۱:

در این ساختار، شرط می تواند مرکب باشد. یعنی، می توان با عملگرهای and ، or شرط های مرکب را ایجاد نمود. به عنوان مثال، شرط مرکب می تواند به صورت زیر بیان گردد.

$x > 10$ and $x < 19$

این شرط بررسی می کند که X بین ۱۰ تا ۱۹ است یا خیر؟

نکته ۲:

برای تست برابری در دستورات شرطی باید از عملگر == استفاده کرد.

مثال ۱-۲. برنامه‌ای که عددی صحیح را خوانده، قدرمطلق آن را نمایش می‌دهد (هدف این برنامه آشنایی با ساختار if ساده است).

مراحل طراحی و اجرا:

۱. ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
n = int(input("Enter a number:"))
if n < 0 :
    n = -n
print( n)
```

متغیر	هدف
n	عدد ورودی

۲. ماژول را ذخیره و اجرا کنید. اکنون جلوی number: عدد ۱۰- را وارد نمایید تا خروجی زیر را

ببینید:

Enter a number:-10

10

ب) ساختارهای دو انتخابی **if**: در ساختار if می‌توان کلمه کلیدی else را به کاربرد، در این صورت این

ساختار به صورت زیر استفاده می‌شود:

```
شرط:
    مجموعه دستورات ۱
else:
    مجموعه دستورات ۲
```

در این ساختار ابتدا شرط ارزیابی می‌شود. اگر نتیجه ارزیابی شرط درست یا True باشد، مجموعه دستورات ۱ و گرنه (نتیجه ارزیابی شرط False باشد)، مجموعه دستورات ۲ اجرا خواهند شد.

مثال ۲-۲. برنامه‌ای که عددی را خوانده، تشخیص می‌دهد زوج است یا فرد (هدف این برنامه آشنایی با ساختار دو انتخابی if - else است).

مراحل طراحی و اجرا:

۱. ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
n = int(input("Enter a number:"))
if n % 2 == 1:
    print("Odd")
else:
    print("Even")
```

متغیر	هدف
n	عدد ورودی

۲. ماژول را ذخیره و اجرا کرده، عدد ۱۵ را وارد کنید تا خروجی زیر را ببینید:

```
Enter a number: 15
Odd
```

ج) ساختارهای چند انتخابی: در این ساختار دستور if را می توان توسعه داد و بخش های بیشتری را با شرط

های مختلف ایجاد نمود، این دستور به صورت زیر به کار می رود:

شرط ۱:

مجموعه دستورات ۱

شرط ۲:

مجموعه دستورات ۱

شرط n:

مجموعه دستورات n

Else :

مجموعه دستورات n+1

در این ساختار ابتدا شرط ۱ ارزیابی می شود، چنانچه برابر True (درست) باشد، مجموعه دستورات ۱ اجرا می شوند و سپس دستور بعد از مجموعه دستورات n+1 اجرا خواهند شد، اگر شرط ۱ درست نباشد، شرط ۲ ارزیابی می شود، اگر شرط ۲ درست (True) باشد، مجموعه دستورات ۲ اجرا می شوند، و سپس دستور بعد از مجموعه دستورات n+1 اجرا می گردد، اگر شرط ۲ درست نباشد، شرط ۳ ارزیابی می شود و این روند تا شرط n ادامه می یابد. اگر هیچ یک از شرط های ۱ تا n درست نباشد، مجموعه دستورات n+1 اجرا خواهند شد.

در هنگام استفاده از این ساختار به نکات زیر دقت کنید:

۱- تعداد بخش های elif اختیاری است و محدودیتی در آن وجود ندارد.

۲- بخش elif نمی تواند قبل از if یا بعد از else قرار بگیرد.

۳- در این ساختار وجود else اختیاری است.

مثال ۳-۲. برنامه‌ای که عددی را خوانده، تشخیص می‌دهد، مثبت، صفر یا منفی است (هدف برنامه آشنایی با if - elif - else است).

مراحل طراحی و اجرا:

۱. ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
val = input("Enter a number:");
val = int(val)
if val > 0 :
    print ("Entered value is positive")
elif val == 0:
    print ("Entered value is zero")
else:
    print ("Entered value is negative")
```

هدف	متغیر
رشته و عدد	val
ورودی	

۲. ماژول را ذخیره و اجرا کرده، عدد ۱۲- را وارد کنید تا خروجی زیر را ببینید:

```
Enter a number:-12
Entered value is negative
```

مثال ۴-۲. برنامه‌ای که ضرایب یک معادله درجه ۲ را خوانده، ریشه‌های آن را محاسبه می‌کند و نمایش می‌دهد.

توضیح: برای محاسبه ریشه‌های معادله درجه ۲ ابتدا ضرایب a, b, c را می‌خوانیم. سپس دلتا

(delta) را به صورت زیر محاسبه می‌کنیم:

$$\text{delta} = b^2 - 4*a*c$$

اگر delta کوچک‌تر از صفر باشد، معادله ریشه ندارد.

وگرنه، اگر delta برابر صفر باشد، معادله دو ریشه مساوی دارد و ریشه‌های آن برابر است با:

$$x_1 = x_2 = -b / (2*a)$$

وگرنه (اگر $\text{delta} > 0$) معادله دارای دو ریشه مختلف است و ریشه‌های معادله به صورت زیر

محاسبه می‌شوند:

$$x_2 = \frac{-b - \sqrt{\text{delta}}}{2*a} \text{ و } x_1 = \frac{-b + \sqrt{\text{delta}}}{2*a}$$

مراحل طراحی و اجرا: ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

هدف	متغیر
ضریب x^2	a
ضریب x	b
عدد ثابت معادله	c
مقدار دلتا	delta
ریشه اول	x1
ریشه دوم	x2

```
a = int(input("Enter a:"))
b = int(input("Enter b:"))
c = int(input("Enter c:"))
delta = b ** 2 - 4 * a * c
if delta < 0:
    print("Not root")
elif delta == 0:
    print(" x1 = x2 = ", -b / (2.0 * a))
else:
    print(" x1 = ", (-b + delta ** 0.5) / (2.0 * a))
    print(" x2 = ", (-b - delta ** 0.5) / (2.0 * a))
```

۲. ماژول را ذخیره و اجرا کنید. اکنون جلوی a:، b:، و c: به ترتیب مقادیر ۲، ۴ و ۲ را وارد

کرده تا خروجی زیر را مشاهده نمایید:

```
Enter a:2
Enter b:4
Enter c:2
x1 = x2 = -1.0
```

تمرین: (دانشجویان گرامی برنامه ها را در محیط پایتون نوشته و از کدها و خروجی عکس گرفته و به ایمیل من ارسال نمایید. Khorshid.ci@gmail.com)

- ۱- برنامه ای بنویسید که ۲ عدد دلخواه دریافت کرده سپس با پیغام مناسب بزرگتر یا کوچکتر یا مساوی بودن آن دو عدد را چاپ کند.
- ۲- برنامه ای بنویسید که عدد دلخواهی دریافت کرده و بررسی کند عدد زوج است یا فرد و با پیغام مناسبی آن را نمایش دهد.
- ۳- برنامه ای بنویسید که عددی را به عنوان سن از شما دریافت کرده سپس بر اساس بازه های دلخواهتان پیامهای مناسبی مثل جوان - میانسال و پیر را نمایش دهد.
