

# برنامه سازی پیشرفته (ساختارهای کنترلی)

صادق اسکندری - دانشکده علوم ریاضی، گروه علوم کامپیوتر

[eskandari@guilan.ac.ir](mailto:eskandari@guilan.ac.ir)

یک ساختار کنترلی، دستوری است که روند اجرای سایر دستورالعمل‌های برنامه را مشخص می‌کند.

if



تصمیم (Decision)

ساختارهای کنترلی

while, for



تکرار (Repetition)

مورد نیاز در هر

زبان برنامه نویسی

break, continue, return



پرش (Jump)

```
if condition:
    body
else:
    else_body
```

ساختار کلی:

```
1 import math
2 a = float(input("enter a:"))
3 b = float(input("enter b:"))
4 c = float(input("enter c:"))
5 d = b**2 - 4.0*a*c
6 if(d < 0.0):
7     print("No real root :(")
8 else:
9     discRoot = math.sqrt(d)
10    root1 = (-b + discRoot) / (2 * a)
11    root2 = (-b - discRoot) / (2 * a)
12    print()
13    print("The solutions are: ", root1, root2 )
```

مثال: معادله درجه دوم  
معادله درجه دوم

## if های تو در تو

```
1 a = int(input("enter a digit: "))
2 if a == 0:
3     print("ZERO")
4 else:
5     if a == 1:
6         print("ONE")
7     else:
8         if a == 2:
9             print("TWO")
10        else:
11            if a == 3:
12                print("THREE")
13            else:
14                print("More than THREE")
```



## استفاده از elif

(معادل switch در زبان C++)

```
1 a = int(input("enter a digit: "))
2 if a == 0:
3     print("ZERO")
4 elif a == 1:
5     print("ONE")
6 elif a == 2:
7     print("TWO")
8 elif a == 3:
9     print("THREE")
10 else:
11     print("More than THREE")
```

# ساختارهای کنترلی: ساختار تکرار

---

```
while condition:  
    body
```

تکرار بر اساس شرط

```
for var in sequence:  
    body
```

تکرار با تعداد گام مشخص

مثال: برنامه ای بنویسید که عددی را از ورودی گرفته و تعداد ارقام آن را مناسبه کند.

```
1 n = int(input("enter a number:"))
2 m = n
3 num_digits = 1
4 while m >= 10:
5     num_digits += 1
6     m //= 10
7
8 print("Num of digits for %d: %d"%(n,num_digits))
```

enter a number:874837

Num of digits for 874837: 6

مثال: برنامه ای بنویسید که دو عدد را از ورودی دریافت کرده و بزرگترین مقسوم علیه مشترک آنها را محاسبه کند.

```
1 # Euclidean Algorithm for GCD(m,n)
2 n = int(input("enter a number: "))
3 m = int(input("enter another number:"))
4 while(n % m != 0):
5     r = n % m
6     n = m
7     m = r
8
9 print(m)
```

enter a number: 24

enter another number:16

8

مثال: برنامه ای بنویسید که عدد  $n$  را از ورودی گرفته و از 1 تا  $n$  را چاپ کند.

```
1 # Printing 1 to n
2 n = int(input("enter a number: "))
3 i = 1
4 while(i <= n):
5     print(i)
6     i += 1
```

تعداد دفعات اجرا مشخص است 😊😊

```
1 n = int(input("enter a number: "))
2 for i in range(1,n+1):
3     print(i)
```



مثال: برنامه ای بنویسید که عدد  $n$  را از ورودی گرفته و اعداد فرد از 1 تا  $n$  را چاپ کند.

```
1 n = int(input("enter a number: "))
2 for i in range(1,n+1,2):
3     print(i)
```

enter a number: 10

1

3

5

7

9

ساختارهای کنترلی پرش در پایتون کاملاً مشابه ساختارهای  
کنترلی پرش در زبانهای C++ و جاوا عمل می کنند.

مثال: برنامه ای بنویسید که تعدادی عدد را از کاربر دریافت کرده و مجموع ارقام آنها را مناسبه کند. این کار باید تا زمانی ادامه یابد که مجموع ارقام برابر ۱۰ نباشد.

```
while True:
    n = int(input("enter a number: "))
    sum_digits = 0
    if(n<0):
        n = -n
    while n>0:
        sum_digits += n%10
        n //= 10
    if sum_digits == 10:
        break
    print("Sum of digits is %d"%(sum_digits))
```

مثال: برنامه ای بنویسید که تعداد ۱۰ عدد را از کاربر دریافت کرده و مجموع ارقام اعداد زوج از این ۱۰ عدد را چاپ کند.

```
for i in range(10):
    n = int(input("enter a number: "))
    if n % 2 == 1:
        continue
    sum_digits = 0
    if(n < 0):
        n = -n
    while(n>0):
        sum_digits += n%10
        n //= 10
    print(sum_digits)
```