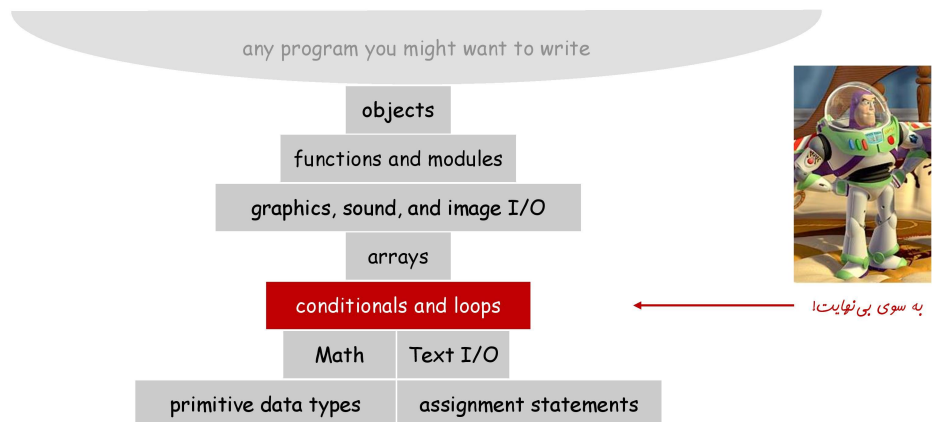


## اجزای برنامه‌نویسی

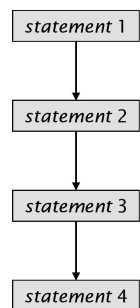


## جریان کنترل

□ جریان کنترل.

□ دنباله‌ای از دستورات در برنامه که واقعاً اجرا می‌شوند.

□ دستورات شرطی و حلقه‌ها: تغییر جریان کنترل.



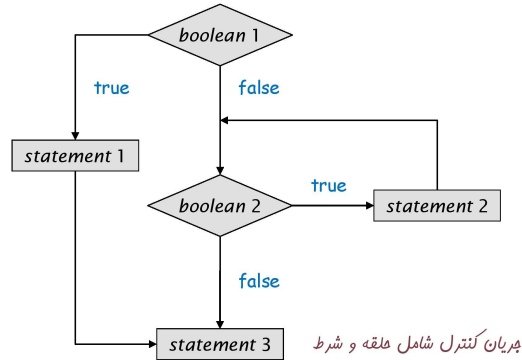
جریان کنترل خط مستقیم

دستورات یکی پس از دیگری به ترتیب مشخص شده اجرا می‌شوند

# جریان کنترل

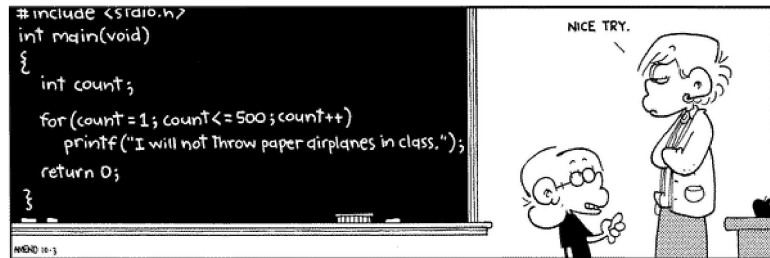
## □ جریان کنترل.

- دنباله‌ای از دستورات در برنامه که واقعاً اجرا می‌شوند.
- دستورات شرطی و حلقه‌ها: تغییر جریان کنترل.



بسیاری از برنامه‌ها به جریان کنترل پیچیده‌تری نیاز دارند

## حلقه for



```
In [ ]: ▶ """
Loop:
for
while
"""
```

```
In [ ]: ▶ for i in range(4):
print(i , end = ' ') # 0 1 2 3
```

```
In [ ]: ▶ for i in range(3,8):  
        print(i , end= ' ')      # 3 4 5 6 7
```

```
In [ ]: ▶ for j in range(4,10,2):  
        print(j , end = ' ')    # 5 7 9
```

```
In [ ]: ▶ s = 'amin'  
        for ch in s:  
            print(ch, end = ' ')  #a m i n
```

```
In [ ]: ▶ for _ in range(3):  
        print('hello', end = ' ') #hello hello hello
```

```
In [ ]: ▶ for i in range(9,2,-3):  
        print(i , end=' ')      # 9 6 3
```

```
In [ ]: ▶ word = 'python'  
        c = 0  
        for i in word:  
            c+=1  
        print(c)                # 6
```

```
In [ ]: ▶ word = 'alireza'  
        c = 0  
        for i in word:  
            if i =='a':  
                c+=1  
        print(c)                # 2
```

```
In [ ]: ▶ name = 'amin'  
        v = 'aeiou'  
        c = 0  
        for ch in name:  
            if ch in v:  
                print(ch)      # a i  
                c += 1  
  
        print(c)                # 2
```

```
In [ ]: ▶ name = 'amin'  
        v = 'aeiou'  
        a = [ch for ch in name if ch in v]  
        print(a)                # ['a', 'i']
```

## ساختارهای تو در تو



```
In [ ]: ▶ for i in range(1,4):  
          for j in range(2,4):  
              print(j,end=' ' )  
          print()
```

```
# 2 3
```

```
# 2 3
```

```
# 2 3
```

```
...
```

```
i = 1 : j=2 , j=3
```

```
i = 2 : j=2 , j=3
```

```
i = 3 : j=2 , j=3
```

```
...
```

```
In [ ]: ▶ for i in range(1,4):  
          for j in range(2,4):  
              print(i,end=' ' )  
          print()
```

```
# 1 1
```

```
# 2 2
```

```
# 3 3
```

```
In [ ]: ▶ for i in range(2,5):
          for j in range(i):
              print(j,end=' ')
          print()

# 0 1
# 0 1 2
# 0 1 2 3

'''
i = 2 : j=0 , j=1
i = 3 : j=0 , j=1 , j=2
i = 4 : j=0 , j=1 , j=2 , j=3
'''
```

```
In [ ]: ▶ for i in range(2,5):
          for j in range(1,i):
              print(j , end = ' ')
          print()

# 0 1
# 0 1 2
# 0 1 2 3
```

```
In [ ]: ▶ print('\n # break #')
          for i in range(5):
              if i == 3 :
                  break
              else:
                  print(i,end=' ') # 0 1 2
```

```
In [ ]: ▶ print('\n # continue #')

          for i in range(5):
              if i == 3 :
                  continue
              else:
                  print(i,end=' ') # 0 1 2 4
```

```
In [ ]: ▶ for n in range(10,15):
          for i in range(2,n):
              if n % i == 0 :
                  print(n , end = ' ') # 10,12,14
              break
```

```
In [ ]: ▶ for i in range(3,8):
        for j in range(2,i):
            if i % j == 0:
                break
            else:
                print(i , end = ' ') # 3 5 5 5 7 7 7 7
```

## حلقه while



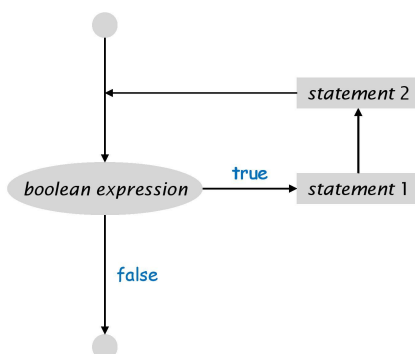
## حلقه while

□ حلقه `while` یک ساختار تکرار متداول.

□ یک عبارت بولی ارزیابی می‌شود.

□ اگر حاصل `true` بود، تعدادی دستور اجرا می‌شود.

□ تکرار.



`while boolean expression:`

```
statement 1 | ← برنه حلقه
statement 2
```

```
In [ ]: ▶ print('# while #')

i = 1
while i <= 3:
    print(i , end= ' ')    # 1 2 3
    i += 1
```

```
In [ ]: ▶ n = 7
while n >= 3:
    print(n , end = ' ')    # 7 6 5 4 3
    n -= 1
```

```
In [ ]: ▶ s = 'abcdef'
i = 0
while True:
    if s[i] == 'd' :
        break
    print(s[i] , end= ' ') # a b c
    i +=1
```

```
In [ ]: ▶ n = 8
while n > 2:
    n -= 1
    if n == 5:
        break
    print(n , end = ' ')    #7 6
```

```
In [ ]: ▶ n = 8
while n > 2:
    n -= 1
    if n == 5:
        continue
    print(n , end = ' ')    #7 6 4 3 2
```

```
In [ ]: ▶ a = 0
b = 1
while a < 10:
    print(a,end=' ')    # 0 1 2 4 8
    a = b
    b = a+b
```

```
In [ ]: ▶ a = 0
b = 1
while a < 10:
    print(a,end=' ')    # 0 1 1 2 3 5 8
    a , b = b , a+b
```

```
In [ ]: ▶ # PEP8
n = 1
while n <= 3 : print(n) ; n+=1
```

```
In [ ]: ▶ import random
n = random.randrange(0,10)
f = 'no'

print(n)

while f == 'no' :
    a = int(input('enter:'))
    if a < n :
        print('>')
    elif a > n:
        print('<')
    else:
        print('correct')
        f = 'yes'

print('Thank you.')
```

## جریان کنترل: خلاصه

### □ جریان کنترل.

□ دنباله‌ای از دستورات در برنامه که واقعاً اجرا می‌شوند.

□ دستورات شرطی و حلقه‌ها: تغییر جریان کنترل.

مثال‌ها	توضیحات	جریان کنترل
	تمام دستورات برنامه به ترتیب داده شده اجرا می‌شوند	خط مستقیم
if if-else if-elif	برخی از دستورات برنامه بسته به مقدار بعضی از متغیرهای خاص اجرا می‌شوند	دستورات شرطی
while for	تا زمانی که شرایط خاصی برقرار باشد، برخی از دستورات به طور مکرر اجرا می‌شوند	حلقه‌های تکرار

دانشگاه شهید مدنی آذربایجان  
برنامه نویسی مقدماتی با پایتون  
امین گلزاری اسکویی  
۱۴۰۰-۱۴۰۱

[Codes and Projects \(click here\) \(https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021\)](https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021), [slides and videos \(click here\) \(https://drive.google.com/drive/folders/1ZsQjBJJ4UAAp9zrGxm3c4qrhmvGBUYHw\)](https://drive.google.com/drive/folders/1ZsQjBJJ4UAAp9zrGxm3c4qrhmvGBUYHw)



