

# برنامه سازی پیشرفته (ادامه انواع داده های اولیه)

صادق اسکندری - دانشکده علوم ریاضی، گروه علوم کامپیوتر

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

نکات ...

آخرین مهلت ارسال تمرین اول: دوشنبه هفته آتی، تا ساعت ۲۴

## یادآوری ...

رشته یک کالکشن مرتب از کاراکترها است که به منظور ذخیره و پردازش داده های متنی مورد استفاده قرار می گیرد.

Single quotes: `'spam'`

Double quotes: `"spam"`

Escape sequences: `"s\tp\na\\m"`

Raw sequences: `r"s\tp\na\\m"`

Triple quotes: `'''... spam ...'''`, `"""... spam ..."""`

# رشته (String)

اندریس گذاری، رشته ها (Indexing)

```
title = 'hello world'
```

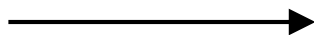
← (right Indexing)

-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1

h	e	l	l	o		w	o	r	l	d
---	---	---	---	---	--	---	---	---	---	---

0 1 2 3 4 5 6 7 8 9 10

(left Indexing)



## رشته (String)

برش رشته ها (Slicing): می توان یک زیر دنباله از یک رشته را استخراج کرد.

string[start:end]      شکل کلی اول:

string[start:end:step]      شکل کلی دوم:

عملیات پایه ای پایتون بر روی رشته ها

توضیحات	عملگر
الاق رشته ها	+
تکرار رشته ها	*
طول رشته ها	len

# لیست (List)

عملیات پایه ای پایتون بر روی لیستها


توضیحات	عملگر
الحاق لیست ها	+
تکرار یک لیست	*
طول لیست	len
تبدیل یک sequence دیگر به لیست	list
بررسی عضویت در لیست	in

# لیست (List)

عملیات پایه ای پایتون بر روی لیستها

```
1 l1 = list('hello')
2 print(l1) -----> ['h', 'e', 'l', 'l', 'o']
3
4 l2 = list(range(-4,5,2))
5 print(l2) -----> [-4, -2, 0, 2, 4]
6
7 l3 = l1 + l2
8 print(l3) -----> ['h', 'e', 'l', 'l', 'o', -4, -2, 0, 2, 4]
9
10 l4 = l2*2
11 print(l4) -----> [-4, -2, 0, 2, 4, -4, -2, 0, 2, 4]
12
13 a = 'e' in l1
14 b = 'e' in l2
15 print('a:',a) -----> a: True
16 print('b:',b) -----> b: False
```

## لیست (List)

علاوه بر عملیات پایه ای، تعداد فراوانی عملیات غیراولیه (متد) نیز برای لیستها تعریف شده است. بحث بیشتر در باره این نوع عملیات به بعد از مباحث برنامه نویسی شیءگرا موكول می گردد. 



# لیست (List)

لیست ها نیز نوعی دنباله (sequence) هستند و در نتیجه می توان از آنها در `for` استفاده کرد 😊

```
1 l = [-1,4,3,2,9]
2 for i in l:
3     print(i**2)
```

```
1
16
9
4
81
```

# لیست (List)

مثال: برنامه ای بنویسید که یک لیست از مقادیر صحیح را از کاربر گرفته و آن را معکوس کند.

```
1 lst = [] → لیست خالی
2 n = int(input('Enter the number of elements:'))
3 for i in range(0,n):
4     element = int(input('enter item %d:'%(i)))
5     lst = lst + [element]
6 print(lst)
7
8 for i in range(0,n//2):
9     temp = lst[i]
10    lst[i] = lst[-1-i]
11    lst[-1-i] = temp
12 print(lst)
```

می توان از متد استفاده کرد

`lst.append(element)`

`Lst = lst + element`

باعث خطا می شود

Enter the number of elements:5

enter item 0: 1

enter item 1: 2

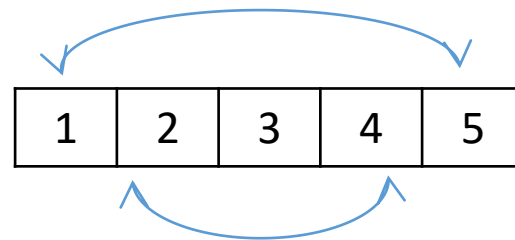
enter item 2: 3

enter item 3: 4

enter item 4: 5

[1, 2, 3, 4, 5]

[5, 4, 3, 2, 1]



# لیست (List)

```
1 prime_lst = []
2 n = int(input('enter an integer:'))
3
4 for i in range(2,n):
5     flag = True
6     for item in prime_lst:
7         if i%item == 0:
8             flag = False
9             break
10    if(flag):
11        prime_lst = prime_lst + [i]
12
13 print(prime_lst)
```

مثال: برنامه ای بنویسید که یک عدد صحیح را از کاربر گرفته و لیستی از اعداد اول کوچکتر از آن را ایجاد کند.

آیا اول است؟ اگر اول است، آن را به لیست اضافه کن

**نکته:** اگر عددی مرکب باشد، حداقل یک شمارنده اول کوچکتر از خود دارد. بنابراین، به جای بررسی تمامی اعداد کوچکتر از آن، بهتر است تنها اعداد اول کوچکتر از آن را بررسی کنیم. آیا می توان این بلاک را به شکل بهتری نوشت؟؟؟

enter an integer:40

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]

## لیست (List)

تمرین: برنامه ای بنویسید که یک لیست از اعداد صحیح را از ورودی گرفته و آن را مرتب کند.

تمرین: برنامه ای بنویسید که دو لیست از اعداد صحیح را از ورودی دریافت کرده و آنها را در قالب یک لیست مرتب ترکیب کند.

تمرین: برنامه ای بنویسید که یک لیست از اعداد صحیح را از ورودی دریافت کرده و یک لیست صحیح جدید تولید کند. عناصر لیست جدید، شامل تعداد ارقام هر یک از عناصر لیست ورودی می باشد.

# لیست (List)

ایجاد لیست از روی لیست های دیگر به کمک **Comprehension**

شکل کلی:

متغیری که نشان دهنده عناصر L1 است

$L2 = [f(x) \text{ for } x \text{ in } L1 \text{ if condition}]$

لیست جدید

دنباله یا لیست موجود

شرطی که دارای نتیجه

True یا False است (اختیاری)

یک تابع (عبارت) بر روی عناصر L1

# لیست (List)

ایجاد لیست از روی لیست های دیگر به کمک **Comprehension**

$L2 = [f(x) \text{ for } x \text{ in } L1 \text{ if condition}]$  **شکل کلی:**

**تفسیر:** از لیست L1، هر عنصری که در شرط condition صدق می کند، را انتخاب کرده و تابع  $f(x)$  را بر روی آن اعمال کرده و نتیجه را به لیست L2 اضافه کن.

دستور فوق معادل کد زیر است:

```
for x in L1:  
    if condition:  
        L2 = L2 + [f(x)]
```

## لیست (List)

```
1 lst = []
2 n = int(input('Enter the number of elements:'))
3 for i in range(0,n):
4     element = int(input('enter item %d:'%(i)))
5     lst = lst + [element]
6 print(lst)
7
8 new_lst = [i for i in lst if i%2!=0]
9 print(new_lst)
```

Enter the number of elements:6

enter item 0: 1

enter item 1: 4

enter item 2: 3

enter item 3: 6

enter item 4: 4

enter item 5: 3

[1, 4, 3, 6, 4, 3]

[1, 3, 3]

مثال: برنامه ای بنویسید که یک لیست صحیح را از کاربر گرفته و عناصر فرد آن را مشخص کند.

## لیست (List)

مثال: برنامه ای بنویسید که لیست توان های سوم اعداد کوچکتر از ۱۰۰ که هم بر ۳ و ۵ یا بر ۱۱ بخش پذیر هستند را تولید کند.

```
1 lst = [x**3 for x in range(0,100) if (x%3==0 and x%5==0) or x%11==0 ]  
2 print(lst)
```

```
[0, 1331, 3375, 10648, 27000, 35937, 85184, 91125, 166375, 216000, 287496, 421875, 456533, 681472, 729000, 970299]
```



## لیست (List)

مثال: برنامه ای بنویسید که یک لیست از اعداد صحیح را از کاربر گرفته و عنصر وسط آن را استخراج کند (فرض کنید عنصر وسط  $X$  باشد)، سپس عناصر لیست را به گونه ای جابجا کند که تمامی عناصری که بعد از  $X$  قرار می گیرند، از  $X$  بزرگتر و تمامی عناصری که قبل از  $X$  قرار می گیرند، از آن کوچکتر باشند.

```
lst = [8,1,10,21,10,13,2,10,8,34,11,23,1,4,2]
center = lst[len(lst)//2]
print(center)
```

```
l1 = [i for i in lst if i<center]
l2 = [i for i in lst if i==center]
l3 = [i for i in lst if i>center]
```

```
lst = l1+l2+l3
print(lst)
```

10

[8, 1, 2, 8, 1, 4, 2, 10, 10, 10, 21, 13, 34, 11, 23]

**نکته:** عمل فوق پایه و اساس یکی از سریعترین الگوریتم های مرتب سازی، تحت عنوان **quicksort** است. این الگوریتم را پس مطالعه توابع بازگشتی مورد بررسی قرار خواهیم داد 😊

## لیست (List)

تمرین: برنامه ای بنویسید که یک لیست از رشته ها را از ورودی گرفته و رشته هایی که طول آنها فرد است را در یک لیست و مابقی رشته ها را در لیستی دیگر قرار داده و آنها را چاپ کند.

تمرین: برنامه ای بنویسید که دو لیست از اعداد صحیح را از ورودی دریافت کرده و اشتراک آنها را در یک لیست جدید قرار داده و آن را چاپ کند.

تمرین: برنامه ای بنویسید که دو لیست از اعداد صحیح را از ورودی دریافت کرده و اجتماع آنها را در یک لیست جدید قرار داده و آن را چاپ کند.

## دیکشنری (Dictionary)

یک کالکشن نامرتب از اشیاء که در آن دسترسی به عناصر از طریق کلید ها (نه اندیس ها) انجام می گیرد:

```
{key1 : val1, key2 : val2, ... , keyn : valn}
```

کلید ها از نوع string هستند ولی مقادیر می توانند از هر نوعی باشند.

```
D ={'name': 'John', 'major': 'Computer Science', 'age': 25,  
'weight': 77.4, 'family': {'father': 'Peter', 'mother': 'Sara'},  
'Grades': {'AP': 18.75, 'Math': 12.5}, 'phone': [0911..8 , 0911..3]}
```

# دیکشنری (Dictionary)

مثالی از تعریف و دسترسی به عناصر دیکشنری

```
1 D = {'name': 'John', 'major': 'Computer Science',  
2     'age': 25, 'weight': 77.4,  
3     'family': {'father': 'Peter', 'mother': 'Sara'} ,  
4     'Grades': {'AP': 18.75, 'Math': 12.5} ,  
5     'phone': [91198438, 912838484]}  
6  
7  
8 print(D['name'])  
9 print(D['family']['mother'])  
10 print(D['phone'][-1])  
11
```

John  
Sara  
912838484

# دیکشنری (Dictionary)

مثالی از عملیات بر روی یک دیکشنری

```
1 D = {'name': 'John', 'major': 'Computer Science',  
2     'age': 25, 'weight': 77.4,  
3     'family': {'father': 'Peter', 'mother': 'Sara'} ,  
4     'Grades': {'AP': 18.75, 'Math': 12.5} ,  
5     'phone': [91198438, 912838484]}
```

```
7 D['name'] = 'Bob' -----> تغییر عناصر دیکشنری
```

```
9 D['friends'] = ['Ali', 'Ahmad'] -----> افزودن یک عنصر جدید
```

```
11 del D['weight'] -----> حذف یک عنصر از دیکشنری
```

```
13 print(D)  
14 print(len(D))
```

```
{'name': 'Bob', 'major': 'Computer Science', 'age': 25, 'family': {'father': 'Peter', 'mother': 'Sara'}, 'Grades': {'AP': 18.75, 'Math': 12.5}, 'phone': [91198438, 912838484], 'friends': ['Ali', 'Ahmad']}
```

**جلسه آینده: توابع**