



دوره جامع پایتون:
بخش تسلط بر کدنویسی به زبان پایتون
جلسه ششم

دکتر ذبیح اله ذبیحی

تابع max و min

```
list1=[1,-3,12,0,4]  
print(max(list1))  
print(min(list1))
```

مثال

```
mylist=[("ali",1),("reza",2),("neda",3)]  
print(list(zip(*mylist)))
```

خروجی:

```
[('ali', 'reza', 'neda'), (1, 2, 3)]
```

مثال

```
coordinate = ['x', 'y', 'z']  
value = [3, 4, 5]  
result = zip(coordinate, value)  
result_list = list(result)  
print(result_list)  
c, v = zip(*result_list)  
print('c =', c)  
print('v =', v)
```

```
[('x', 3), ('y', 4), ('z', 5)]  
c = ('x', 'y', 'z')  
v = (3, 4, 5)
```

خروجی

مثال: رمز کردن یک جمله با معکوس کردن کلمات و ساختار جمله

```
message ="Hello, Start the attack "
```

```
code="" "
```

```
i=len(message)-1
```

```
while i>=0 :
```

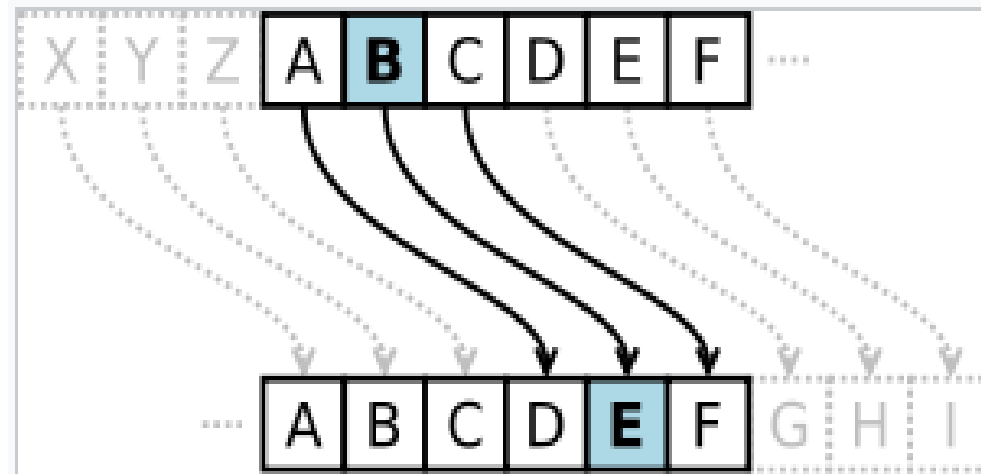
```
    code =code+message[i]
```

```
    i=i-1
```

```
print(code)
```

الگوریتم رمز سزار Caesar cipher

در رمزنگاری ، **رمز سزار** به که با نام‌های **کد سزار** ، **شیفت سزار** - یا **رمز شیفت** نیز شناخته می‌شود، یکی از ساده‌ترین و شناخته‌شده‌ترین تکنیک‌های رمز گذاری است. این رمز یک نوع رمز جانشینی است که در آن هر حرف در متن آشکار با حرف دیگری با فاصله ثابت در الفبای جایگزین می‌شود. برای مثال با مقدار انتقال ۳، **D** به جای **A** می‌نشیند، **E** به جای **B**، و الی آخر. نام این روش از ژولیوس سزار گرفته شده‌است که از آن برای ارتباطات محرمانه خود استفاده می‌کرد.



ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

کدهای اسکی

کد و کارکتر استخراجی از جداول اسکی

```
s=ord("A")
```

```
print(s)
```

```
SS=chr(65)
```

```
print(ss)
```


کد رمز سزار

```
text="Hello, Start the attack"
result = ""
s=4 #displacement
for i in range(len(text)):
    char = text[i]
    if (char.isupper()):
        result=result+ chr((ord(char) + s-65) % 26 + 65)
    else:
        result= result+ chr((ord(char) + s - 97) % 26 + 97)
print(result)
```

باز کردن کد رمز قبلی در صورت داشتن S

```
text="Lipps0$Wxevx$xli$exxego"  
result = ""  
s=4 #displacement  
for i in range(len(text)):  
    char = text[i]  
    if (char.isupper()):  
        result=result+ chr((ord(char) + s-65) % 26 + 65)  
    else:  
        result= result+ chr((ord(char) + s - 97) % 26 + 97)  
print(result)
```

دیکشنری (Dictionary)

- نوع داده دیکشنری در زبان برنامه نویسی پایتون، به صورت لیستی از، کلیدها و valueها است، هر کلید توسط علامت کالن (:) از value جدا می شود، اندیس ها نیز توسط علامت کاما (,) از یکدیگر جدا می شوند. یک دیکشنری خالی بدون اندیس با استفاده از تنها دو آکولاد خالی به صورت {} نوشته می شود.
- کلید ها یکتا می باشد (بدین معنی که نمی توان دو کلید یکسان داشت)، ولی value می تواند تکراری نیز باشد.
- value در دیکشنری می تواند هر نوعی باشد، ولی کلیدها باید یک نوع داده غیر قابل تغییر مانند string, number, یا tuple باشد.
- دیکشنری ها در زبان پایتون نوعی دیگر از لیست ها هستند، اما با یک تفاوت اساسی و آن تفاوت امکان تعیین کلید برای هر یک از آیتم های لیست است که می توان بوسیله این کلید به آیتم های لیست به سرعت دسترسی داشت، یعنی شما مثل لیست ها از اندیس آیتم استفاده نمی کنید که باعث می شود نیازی به مرتب بودن لیست نباشد.

مثال

```
dict = {'first_name': 'Zabiholah', 'Last_name': "Zabihi", "age": 32}
```

```
print ( dict['first_name'])
```

```
dict["age"]=25
```

```
printdict(["age"])
```

حذف در دیکشنری

دستور `dict.clear()` تمام مقادیر داخل دیکشنری را پاک می کند.
دستور `del dict` کل دیکشنری را حذف می کند.

مثال

```
dict = {'first_name': 'Zabiholah', 'Last_name': "Zabihi", "age": 32}
```

```
dict.clear()
```

```
print(dict["age"])
```

```
dict = {'first_name': 'Zabiholah', 'Last_name': "Zabihi", "age": 32}  
del dict
```

```
print(dict["age"])
```

مثال

```
my_dict = {'C1':[1,2,3],'C2':[5,6,7],'C3':[9,10,11]}  
print(my_dict["C1"])
```


مثال: اضافه کردن به دیکشنری

```
mydict = {'first_name': 'Zabiholah', 'Last_name': 'Zabihi', 'age': 32}
```

```
mydict["ID"]=123456
```

```
print(mydict)
```

مثال: حذف عضو از دیکشنری

```
x={"name":["ali","sasan","omid"],"age":[22,33,28]}  
x.pop("name")  
print(x)
```

```
x={"name":["ali","sasan","omid"],"age":[22,33,28]}  
x.clear()  
print(x)
```

```
x={"name":["ali","sasan","omid"],"age":[22,33,28]}  
del x  
print(x)
```

```
x={"name":["ali","sasan","omid"],"age":[22,33,28]}  
a1=x["name"]  
a2=x["age"]  
a1.pop(1)  
a2.pop(2)  
x["name"]=a1  
x["age"]=a2  
print(x)
```

مثال

```
for i in range(4):  
    s=i**3  
    print(i,"",s)
```

روش دیگر

```
s = {}  
for i in range(4):  
    s[i] = i*i  
print(s)
```

روش دیگر

```
s = {i: i**3 for i in range(4)}  
print(s)
```

مثال

```
s= {i: i*i for i in range(15,35) if i%2 == 0}  
print(s)
```

خروجی

توابع توکار دیکشنری

- `len()` طول یک دیکشنری را می دهد.

```
s={1:1, 2:9, 5:12, 3:19}  
print(len(s))
```

- `sorted()` یک لیست مرتب شده جدید از کلیدها می دهد.

```
s={1:1, 2:9, 5:12, 3:19}  
print(sorted(s))
```

خروجی

```
[1,2,3,5]
```

- `str()` یک رشته قابل نمایش از دیکشنری را می‌دهد.

مثال: تبدیل لیست و تاپل به دیکشنری

```
r_list=[["Ali",29],['reza',3]]  
r_dict=dict(r_list)  
print(r_dict)
```

مثال دیگر

```
r_tuple=(("Ali",29),('reza',3))  
r_dict=dict(r_tuple)  
print(r_dict)
```



```
r_tuple=(["Ali",29],['reza',3])  
r_dict=dict(r_tuple)  
print(r_dict)
```

مثال دیگر

```
r_list=[("Ali",29),('reza',3)]  
r_dict=dict(r_list)  
print(r_dict)
```

مثال

```
keys = ['red', 'green', 'blue']  
values = ['#FF0000', '#008000', '#0000FF']  
color_dictionary = dict(zip(keys, values))  
print(color_dictionary)
```

جمع دو دیکشنری

```
dic1={0:1,1:2}  
dic2={2:3,3:4}  
dic3=dic1+dic2  
print(dic3)
```

Output:

error

ترکیب دو دیکشنری

```
mydic1={1:"Ali",2:"reza"}  
mydic2={3:"zahra",4:"neda"}  
mydic1.update(mydic2)  
print(mydic1)
```

لیست کلید ها و مقادیر

```
mydic={1:"Ali",2:"reza"}  
print(mydic.keys())  
print(mydic.values())  
print(mydic.items())
```

مثال

```
my_dict = {'data1':10,'data2':-13,'data3':24}  
result=1  
for i in my_dict:  
    result=result * my_dict[i]  
print(result)
```

مجموعه (set)

- مجموعه دسته ای از اشیاء مختلف متمایز از هم است.
- مجموعه ها همان دیکشنری ها هستند، فقط مقدار ندارند!
- مجموعه ای از مقادیر که امکان درج مقدار تکراری در آن وجود ندارد، مانند کلید ها در دیکشنری که امکان استفاده از دو کلید یکسان در دیکشنری وجود نداشت.
- نمی توانیم عناصر مجموعه را تغییر دهیم عناصر مجموعه، تغییرناپذیر هستند، اما نکته مهم این است که خود مجموعه را می توانیم تغییر دهیم (خود مجموعه تغییرپذیر (mutable) است)، زیرا امکان اضافه کردن عنصر به آن و حذف عنصر از آن، وجود دارد، ولی به دلیل اینکه مثلا نمی توانیم بگوییم که مقدار عنصر پنجم از مجموعه به ۲ تغییر داده شود (مثل امکانی که در فهرست ها داریم) چون در مجموعه، ترتیبی برای عناصر وجود ندارد، گفته می شود که امکان تغییر عناصر مجموعه وجود ندارد.

مثال

```
s_set = {"Ali", "reza", 3}
```

```
for x in s_set:  
    print(x)
```


اضافه کردن عضو به مجموعه

```
s_set = {"Ali", "reza", 3}
```

```
s_set.add(125)
```

```
print(s_set)
```

عضو جدید را بصورت نامنظم به لیست اضافه می کند و در هر بار اجرا در محل جدید عضو جدید را اضافه می کند. برای اضافه کردن تک عضو از `add()` و چند عضو از `update()` استفاده می کنیم

```
s_set = {"Ali", "reza", 3}
```

```
s_set.update([125,"neda"])
```

```
print(s_set)
```

حذف عضو از مجموعه

```
s_set = {"Ali", "reza", 3}  
s_set.remove(3)  
print(s_set)
```

روش دیگر

```
s_set = {"Ali", "reza", 3}  
s_set.discard(3)  
print(s_set)
```

ترکیب مجموعه

```
set1={1,2,3}  
set2={"ali","reza"}  
set1.update(set2)  
print(set1)
```

روش دیگر

```
set1={1,2,3}  
set2={"ali","reza"}  
set3=set1.union(set2)  
print(set3)
```

مثال

```
s_set=set(("ali","reza"))  
print(s_set)
```

اشتراک دو مجموعه

```
set1={1,2,3,6,8,0}  
set2={1,2,3,9,7}  
set3=set1.intersection(set2)  
print(set3)
```

روش دیگر

```
set1={1,2,3,6,8,0}  
set2={1,2,3,9,7}  
set3=set1 & set2  
print(set3)
```

اختلاف دو مجموعه

```
set1={1,2,3,6,8,0}
```

```
set2={1,2,3,9,7}
```

```
set3=set1.difference(set2)
```

```
print(set3)
```

```
set1={1,2,3,6,8,0}
```

```
set2={1,2,3,9,7}
```

```
set3=set1.symmetric_difference(set2)
```

```
print(set3)
```

مجموعه ای از اعضای غیرمشترک دو مجموعه ایجاد می شود.

مثال: دریافت لیست داده های عددی از کاربر

```
my_list = []  
n = int(input("Enter number of elements : "))  
for i in range(0, n):  
    ele = int(input("adding the element"))  
    my_list.append(ele)  
print(my_list)
```


مثال: دریافت لیست داده های از اسامی از کاربر

```
my_list = []  
n = int(input("Enter number of elements : "))  
for i in range(0, n):  
    ele = input("adding the element")  
    my_list.append(ele)  
print(my_list)
```

مثال: اسامی و سن را از کاربران دریافت و لیستی بسازیم

```
my_list = [ ]  
n = int(input("Enter number of elements : "))  
for i in range(0, n):  
    ele = [input("last_name="), int(input("age="))]  
    my_list.append(ele)  
  
print(my_list)
```

تمرین

- از کاربر سن و اسم افراد را دریافت کنید و در خروجی دیکشنری از اطلاعات ورودی چاپ کنید بطوری که اسم key هر عضو و سن value هر عضو باشد.

تمرین

- کدی بنویسید که یک متن از کاربر دریافت کند و لغات خاصی را از آن فیلتر کند.