

حل مسئلہ

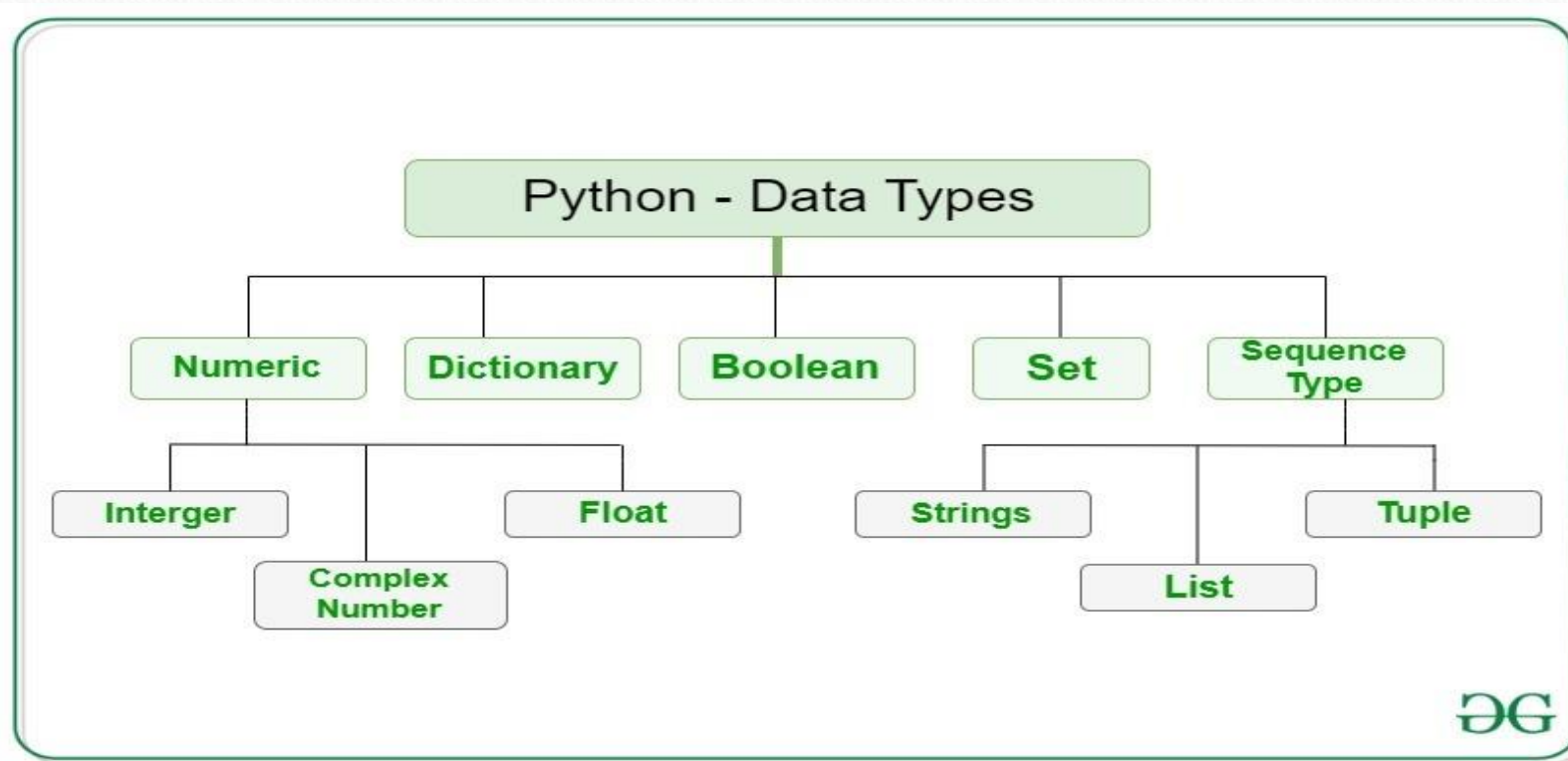
دوره آموزشی پایتون

جلسه پنجم

عبداله صارمی نایینی



Data Type



Numbers

- با تابع `type` می توانیم نوع داده را مشخص کنیم.
- داده `float` با یک نقطه «.» یا به صورت توانی با استفاده از `e` یا `E` نوشته می شود.
- `True` معادل یک و `False` معادل صفر می باشد.
- با تابع `del` می توان یک متغیر را آزاد نمود.

String

- برای نمایش یک رشته به صورت های زیر عمل می کنیم:
 - "Sample" امکان استفاده از نویسه «'»
 - 'Sample' امکان استفاده از نویسه «"»
 - """Sample""" به صورت چند سطری
 - """Sample""" به صورت چند سطری
- مانند زبان C می توان از Escape Code استفاده کرد
 - \n, \t, \\\, \', \", \r, ...
- برای از نادیده گرفتن کد های گریز می توان قبل رشته حرف r گذاشت مانند:
 - print(r"a\nb")
- برای تبدیل شدن به یونیکد از پیشوند u استفاده می کنیم.

✓ Slicing Operator

String sample

```
>>> panagram='The Quick Brown Fox Jumps Over The Lazy Dog'
>>> panagram[0]
'T'
>>> panagram[4:9]
'Quick'
>>> panagram[4:]
'Quick Brown Fox Jumps Over The Lazy Dog'
>>> panagram[-1]
'g'
>>> panagram[1:100]
'he Quick Brown Fox Jumps Over The Lazy Dog'
>>> panagram[100:200]
''
>>> panagram[0]='t'
Traceback (most recent call last):
  File "<pyshell#78>", line 1, in <module>
    panagram[0]='t'
TypeError: 'str' object does not support item assignment
>>>
```

String

- + Operator
 - “Guido”+” van “+”Rossum” →” Guido van Rossum”
- * Operator
 - “A”*5 →”AAAAA”
 - Print(“\n”*50) → Clear Screen
- in
 - ‘a’ in ‘python’ → False
 - ‘a’ not in ‘python’ → True

list

-
- از علامت [] برای نمایش آن استفاده می کنیم و بین عناصر لیست کاما می گذاریم.
 - مانند رشته ها از عملگرهای زیر می توانیم استفاده کنیم:
 - Slicing Operator
 - + Operator
 - * Operator
 - in
 - not in
 - امکان تغییر یک عنصر لیست وجود دارد.

List Sample

```
>>> number=['one','two','three','four','five']
>>> number[0]
'one'
>>> number[1:4]
['two', 'three', 'four']
>>> number[-1]
'five'
>>> number[2:]
['three', 'four', 'five']
>>> number+number
['one', 'two', 'three', 'four', 'five', 'one', 'two', 'three', 'four', 'five']
>>> number*2
['one', 'two', 'three', 'four', 'five', 'one', 'two', 'three', 'four', 'five']
>>> number[1]=2
>>> number
['one', 2, 'three', 'four', 'five']
>>> print(number)
['one', 2, 'three', 'four', 'five']
>>> len(number)
5
```


Range Function

```
range(6)
range(0, 6)
range(1, 6)
range(1, 6)
range(1, 6, 2)
range(1, 6, 2)
x=range(1, 6, 3)
list(x)
[1, 4]
```

مسئله یک: فرض کنید دو تاس به رنگ های قرمز و آبی را همزمان روی زمین می اندازیم چند حالت داریم که مجموع اعداد روی دو تاس ۱۰ باشد؟

Red=1,2,3,4,5,6}

Blue=1,2,3,4,5,6}

Red*Blue=1,1), (1,2), (1,3),(1,4),(1,5),(1,6),

(2,1), (2,2), (2,3),(2,4),(2,5),(2,6),

(3,1), (3,2), (3,3),(3,4),(3,5),(3,6),

(4,1), (4,2), (4,3),(4,4),(4,5),(4,6),

(5,1), (5,2), (5,3),(5,4),(5,5),(5,6),

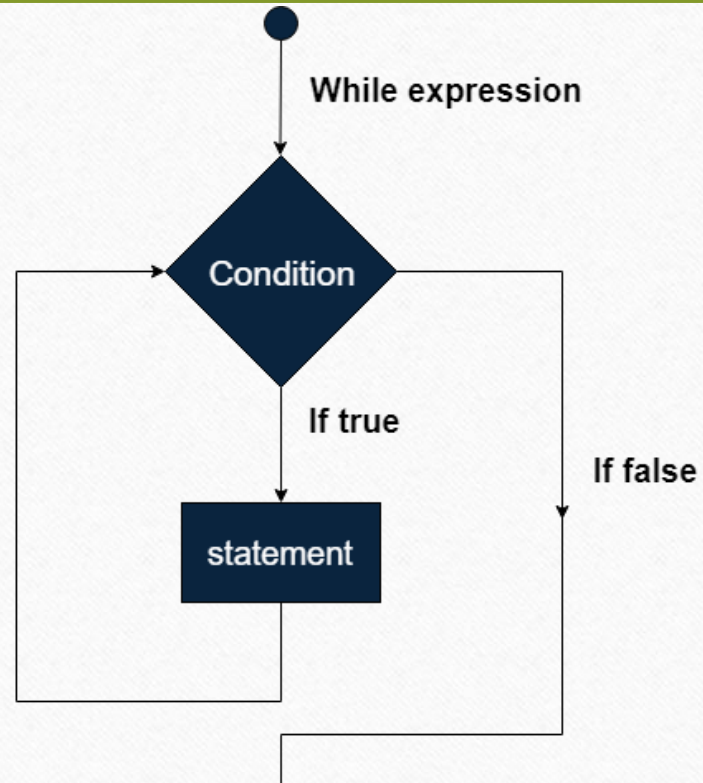
(6,1), (6,2), (6,3),(6,4),(6,5),(6,6)}

{(4,6), (5,5), (6,4)}

مسئله یک: فرض کنید دو تاس به رنگ های قرمز و آبی را همزمان روی زمین می اندازیم چند حالت داریم که مجموع اعداد روی دو تاس ۱۰ باشد؟

```
state=[1,1]
for i in range(36):
    if state[0]+state[1]==10:
        print(state)
    if state[1]<6:
        state[1]=state[1]+1
    else:
        state[0]=state[0]+1
        state[1]=1
```

while

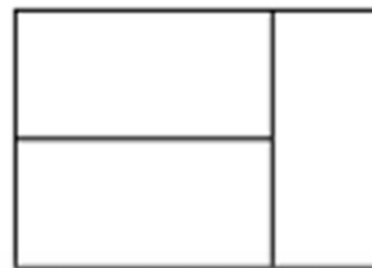
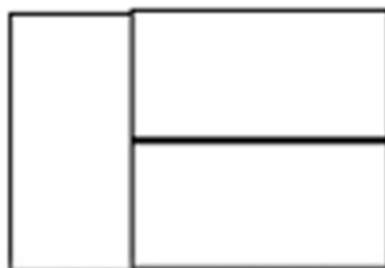
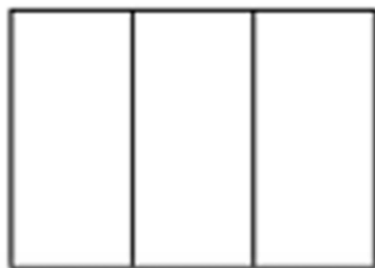


مسئله یک: فرض کنید دو تاس به رنگ های قرمز و آبی را همزمان روی زمین می اندازیم چند حالت داریم که مجموع اعداد روی دوتاس ۱۰ باشد؟

```
state=[1,1]
while state!= [6,6]:
    if state[0]+state[1]==10:
        print(state)
    if state[1]<6:
        state[1]=state[1]+1
    else:
        state[0]=state[0]+1
        state[1]=1
```

آزمون اول

یک راهرو به عرض ۲ و طول n داریم قصد داریم آن را با سرامیک های به طول ۲ و عرض ۱ فرش کنیم. چند مدل فرش داریم؟
برای یک راهرو به طول ۳ مدل های زیر را داریم:



پایان

از توجه شما متشکرم