



Python lecture 2

جلسه قبل

- بررسی ساختار نحوی و معنایی جملات و دستورات در زبان
- چند نوع خطا داریم
- 4 نوع شیء `int` , `float` , `boolean` , `None`
- تبدیل نوع اشیاء به یکدیگر (`casting`)
- دستور `print`
- عملگرهایی بر روی نوع داده `int` , `float`

امروز

- نحوه تعریف متغیر و مقدار دهی آن
- نوع داده رشته
- ورودی و خروجی با استفاده از دستورات `input` , `print`
- عملگرهای مقایسه‌ای
- عملگرهای منطقی بر روی نوع داده `boolean`
- بلاک بندی کدها
- انواع دستورات شرطی
- حلقه تکرار `while`
- حلقه تکرار `for`

ذخیره سازی مقدار در متغیر

- برای **انتساب** یک مقدار به یک متغیر (ذخیره سازی یک مقدار در یک متغیر) از دستور = استفاده می شود

مقدار نام متغیر

```
name = "Ali"
```

```
year = 1396
```

- مقادیر در **حافظه** کامپیوتر ذخیره می شوند (RAM)
- دستور انتساب یک مقدار را به یک نام نسبت می دهد
- دسترسی به مقدار یک متغیر با استفاده نام آن امکان پذیر است

```
print(name) → Ali
```

مثال

pi = 3.1415

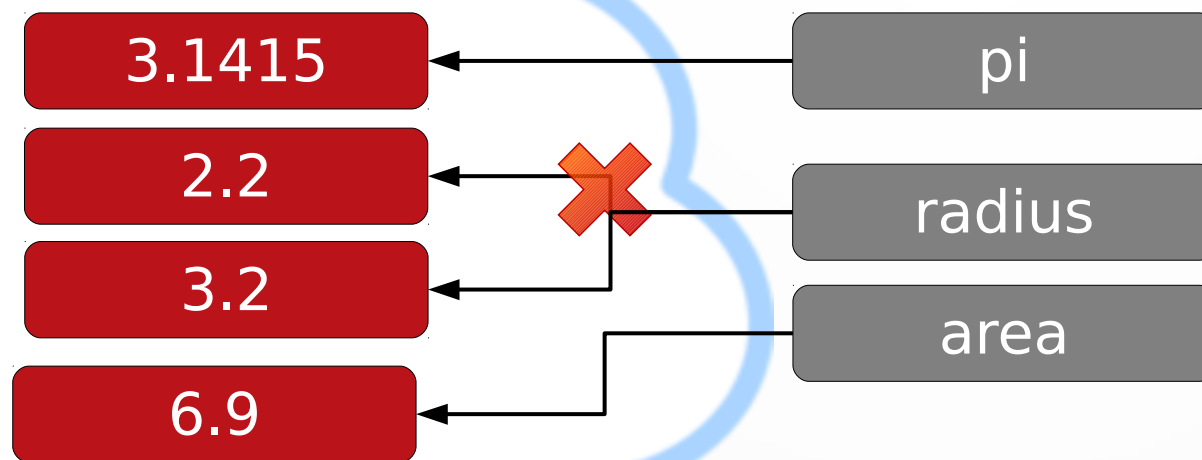
radius = 2.2

area = pi * (radius ** 2)

عبارتی که باید ارزیابی شود متغیر

تغییر مقدار متغیر

- با استفاده از دستور انتساب(=) می‌توان مقدار متغیر تعریف شده را تغییر داد
- مقدار قبلی متغیر ممکن است در حافظه وجود داشته باشد ولی قابل دسترس نیست
- مساحت دایره با تغییر این متغیر تغییر نمی‌کند و محاسبه مساحت باید مجدداً انجام شود



رشته‌ها

- متشکل از حروف الفبا ، اعداد ، فاصله ، حروف خاص
- بین دو " " قرار می‌گیرند

`str1 = "salam"`

`str2 = 'salam'`

- چسباندن دو رشته به هم با عملگر + انجام می‌شود

`name = 'mohammad'`

`greeting = str1 + name → salammohammad`

`greeting2 = str1 + " " + name → salam mohammad`

ورودی/خروجی: print

- جهت نمایش اطلاعات و برقراری ارتباط بین کاربر و برنامه به کار می‌رود.

```
pi = 3.14
```

```
print(pi)
```

```
str_pi = str(pi)
```

```
print("pi number is :", pi) # print each part of input separately
```

```
print("pi number is : " + str(pi)) # concatenate two string
```


ورودی/خروجی: input:

- دریافت ورودی از کاربر از طریق کنسول

```
text = input("enter what you want\n")  
print(5*text)
```

- مقداری که تابع input بر می گرداند یک رشته است
- در صورتی که مقدار ورودی یک عدد باشد باید رشته **تبدیل به عدد** شود

```
number=int(input("enter an integer\n"))  
print(5*number)
```

عملگرهای مقایسه‌ای

int , float, string

- $i=5$, $j=6$

- نتیجه‌ای که یک عملگر مقایسه‌ای برمی‌گرداند یک مقدار True / False است

- $i < j \rightarrow \text{True}$
- $i > j \rightarrow \text{False}$
- $i \leq j \rightarrow \text{True}$
- $i \geq j \rightarrow \text{False}$
- $i == j \rightarrow \text{False}$
- $i \neq j \rightarrow \text{True}$

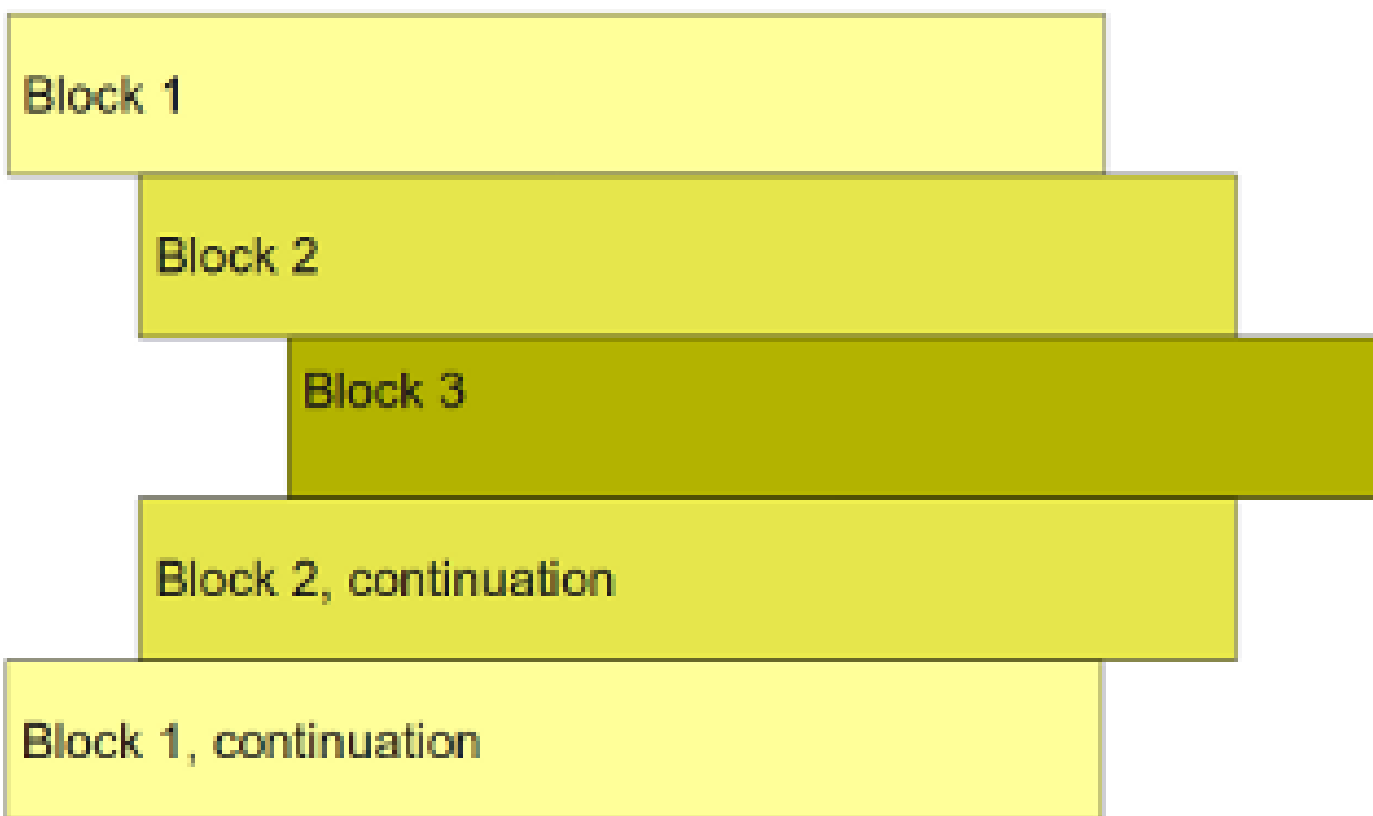
عملگرهای منطقی بر روی متغیرهای Boolean

• a و b نام متغیرهایی هستند از جنس `boolean`

- $\text{not } a \rightarrow$ True if a is False
False if a is True
- $a \text{ and } b \rightarrow$ True if both are True
- $a \text{ or } b \rightarrow$ True if one or both are True

بلاک بندی کدها

- در زبان پایتون برای مشخص کردن بلاک از تورفتگی استفاده می‌شود
- نحوه مشخص کردن یک بلاک در زبان پایتون



دستورات شرطی



انواع دستورات شرطی

```
if <شرط>:  
    <دستور ۱>  
    <دستور ۲>  
    ...
```

```
if <شرط>:  
    <دستور ۱>  
    <دستور ۲>  
    ...  
else:  
    <دستور ۱>  
    <دستور ۲>  
    ...
```

```
if <شرط>:  
    <دستور ۱>  
    <دستور ۲>  
    ...  
elif <شرط>:  
    <دستور ۱>  
    <دستور ۲>  
    ...  
else:  
    <دستور ۱>  
    <دستور ۲>  
    ...
```

مثال دستورات شرطی

```
a = 5
b = 6
if ( a == b ):
    print( " a == b " )
if ( a > b ):
    print( " a > b " )
elif ( a < b ):
    print( " a < b " )
else:
    pass # do nothing
```

حلقه تکرار while

- هنگامی استفاده می‌شود که ندانیم دقیقاً دستور مورد نظر ما چند بار باید تکرار شود

- ساختار حلقه While

```
year=int(input("please enter year\n"))
while year < 2029:
    year += 1
    print(year)
```


حلقه تکرار for

For (اندازه هر گام, حد بالا, مقدار اولیه) in range شمارنده:

<دستور ۱>

<دستور ۲>

...

```
year=int(input("please enter year\n"))  
for x in range(temp_year,2029):  
    print(x)
```

```
year=int(input("please enter year\n"))  
for x in range(temp_year,2029,2):  
    print(x)
```