

Conditions

عبارات بولی

- عبارتی که مقدار آن درست (True) و یا غلط (False) باشد.
- در پایتون به ترتیب برای مقدار درست و غلط مقادیر ۱ و ۰ هم در نظر گرفته می شود.

```
>>> 5 == 5
```

Output: True

- چند عملگر مقایسه ای دیگر:

```
>>> x != y
```

```
>>> x > y
```

```
>>> x < y
```

```
>>> x >= y
```

```
>>> x <= y
```

عملگرهای منطقی

- در پایتون سه عملگر منطقی وجود دارد؛ and، or و not
- شبیه مفهومی که در زبان انگلیسی

```
>>> n = 16
```

```
>>> n%2 == 0 or n%3 == 0
```

```
Output: True
```

- عملگر % برای محاسبه باقی مانده بکار می رود.

دستور if

- برای ایجاد دستورات شرطی استفاده می شود.

- ساختار یک if ساده

```
>>> if (condition(s)):  
    statements
```

- دستور if مثل تابع جز دستورات مرکب در پایتون است. یعنی شامل یک عنوان و بلاکی از دستورات

Header:

First Statement

...

Last Statement

انتخاب دوگانه - else

- دومین فرم دستور if اجرای انتخاب دوگانه می باشد با کمک else می باشد.

```
>>> n = 16
>>> if x%2 == 0:
    print(x , ' is even')
else:
    print(x , ' is odd')
```

Output: 16 is even

شرط تودرتو

• به مثال زیر توجه کنید:

```
word = input('Enter something: ')
if word == 'hi':
    print('hi to you too!')
else:
    if word == 'hello':
        print('hello hello!')
    else:
        if word == 'howdy':
            print('Howdyyyyyyy')
        else:
            if word == 'hey':
                print('Hey hey hey!')
            else:
                if word == 'gday m8':
                    print('gday 4 u 2!')
                else:
                    print("I don't know what ", word, "means.")
```

کد ناخوانا می باشد

دستور elif

```
word = input('Enter something: ')
if word == 'hi':
    print('hi to you too!')
elif word == 'hello':
    print('hello hello!')
elif word == 'howdy':
    print('Howdyyyyyyy')
elif word == 'hey':
    print('Hey hey hey!')
elif word == 'gday m8':
    print('gday 4 u 2!')
else:
    print("I don't know what ", word, "means.")
```

- برای حل مشکل قبل در پایتون از elif استفاده می کنیم.
- این دستور برای ایجاد دستورات شرطی زنجیره ای بکار می رود.

دستور return

- امکان توقف اجرا قبل از رسیدن به انتهای تابع را فراهم می کند.
- خیلی وقت ها برای نمایش یک وضعیت خطا در برنامه نویسی مورد استفاده قرار می گیرد.

```
import math
def printLogarithm(x):
    if x <= 0:
        print('positive number onliy!')
        return
    result = math.log10(x)
    print(result)
```


تمرین

- تمرینات می بایست در تاریخ مشخص به TA در قالب یک فایل PDF ایمیل شود.
- در فایل ارسالی حتما نام و نام خانوادگی خود را بنویسید.

تمرین ۱

1. This program contains several problems. Copy-paste it to a file, then try to run it, fix the errors you got, try to run it again and keep going until it works.

```
print(Hello!)  
something == input('Enter something: )  
print('You entered:' something)
```

تمرین ۲

2. Fix this program the same way:

```
print('Hello!')  
something = input("Enter something: ")  
if something = 'hello':  
    print("Hello for you too!")  
elif something = 'hi'  
    print('Hi there!')  
else:  
    print(' don't know what, something, "means.")
```

تمرین ۳

3. Write a program into a file that asks the user to write a word and then prints that word 1000 times. For example, if the user enters hi the program would reply *hihihihihihihihi ...*

تمرین ۴

4. Add spaces between the words, so the output is like *hi hi hi hi ...*

تمرین ۵

5. Make something that asks the user to enter two words, and prints 1000 of each with spaces in between. For example, if the user enters hello and hi the program would print *hello hi hello hi hello hi hello hi ...*

تمرین ۶

6. Make a program that asks for a password and prints *Welcome!*, *Access denied* or *You didn't enter anything* depending on whether the user entered the correct password, a wrong password, or nothing at all by pressing Enter without typing anything.