In [ ]:
```
"""
Operators :
    Arithmetic : +,-,*,/,%,**,//
    Assignment : =,+=,-=,*= ,/= ,%= ,//= ,**=
    Comparison : ==,!=,>,<,>=,<=
    Logical    : and, or, not
    Membership : in , not in
    Bitwise    :  &, |, ^, ~, <<, >>
"""

print('Arithmetic Operators')

#Addition
print(1 + 3)       # 4

#Subtraction
print(5 - 3)       # 2

#Multiplication
print(2 * 3)       # 6

#Float Division
print(3 / 2)       # 1.5

#Integer Division
print(3 // 2)      # 1

#Remainder
print(17 % 5)      # 2

# Exponentiation
print(2 ** 3)      # 8
print(0 ** 0)      # 1
print(6 ** 0)      # 1
```

In [ ]:
```
print('Operator Precedence')

print(8 - 2 * 3)                # 2
print(1 + 3 * 4 / 2)            # 7.0
print(16 / 2 ** 3)              # 2.0
print(2**2**3)                  # 256
```

In [ ]:
```
print('Augmented Assignment Operator')

x = 4
x += 2    # x = x + 2
print(x)  # 6

y = 8
y //= 2   # y = y // 2
print(y)  # 4
```

عملگرهای مقایسه‌ای

◻ عملگرهای مقایسه‌ای. عملوندهایی از یک نوع را دریافت و یک عملوند از نوع بولی تولید می‌کنند.

op	meaning	true	false
==	*equal*	2 == 2	2 == 3
!=	*not equal*	3 != 2	2 != 2
<	*less than*	2 < 13	2 < 2
<=	*less than or equal*	2 <= 2	3 <= 2
>	*greater than*	13 > 2	2 > 13
>=	*greater than or equal*	3 >= 2	2 >= 3

In [ ]:
```
print('Comparison Operators')

print(2 == 3)                        # False
print(2 != 3)                        # True
print(2 < 3)                         # True

print('Logical Operators')

print(1<3  or 4>5)                   # True
print(1<3 and 4>5)                   # False
print(not 1<3)                       # False

# 'Short-circuit'
print(1 >= 2 and (5/0) > 2)          # False

#print(3 >= 2 and (5/0) > 2)         # division by zero
```

In [ ]:
```
print('Membership Operators')

x = [1,2,3,4,5]
print(3 in x)                        # True
print(24 not in x)                   # True
```

In [ ]:

```
print('Bitwise Operators')

a = 13
print(bin(a))                               # 1101

b = 14
print(bin(b))                               # 1110

###

c = a | b
print(bin(c))                              # 1111

###

c = a & b
print(bin(c))                              # 1100

###

c= a ^ b
print(bin(c))                              # 0011

###

a = 13
print(a << 1)                              # 26

###

a = 20
print(a >> 1)                             # 10

###

a = 18
print(a >> 2)                             # 4

###

a = 20
print(~a)                                # -21 # -(a+1)
```

In [ ]:
```
print('--- Operations on string ---')

s1 = 'Amin'
s2 = ' Golzari Oskouei'
s3 = s1 + s2            # Amin Golzari Oskouei
print(s3)

###

s = 'sara'
print(3* (s + ' '))     # sara sara sara
```

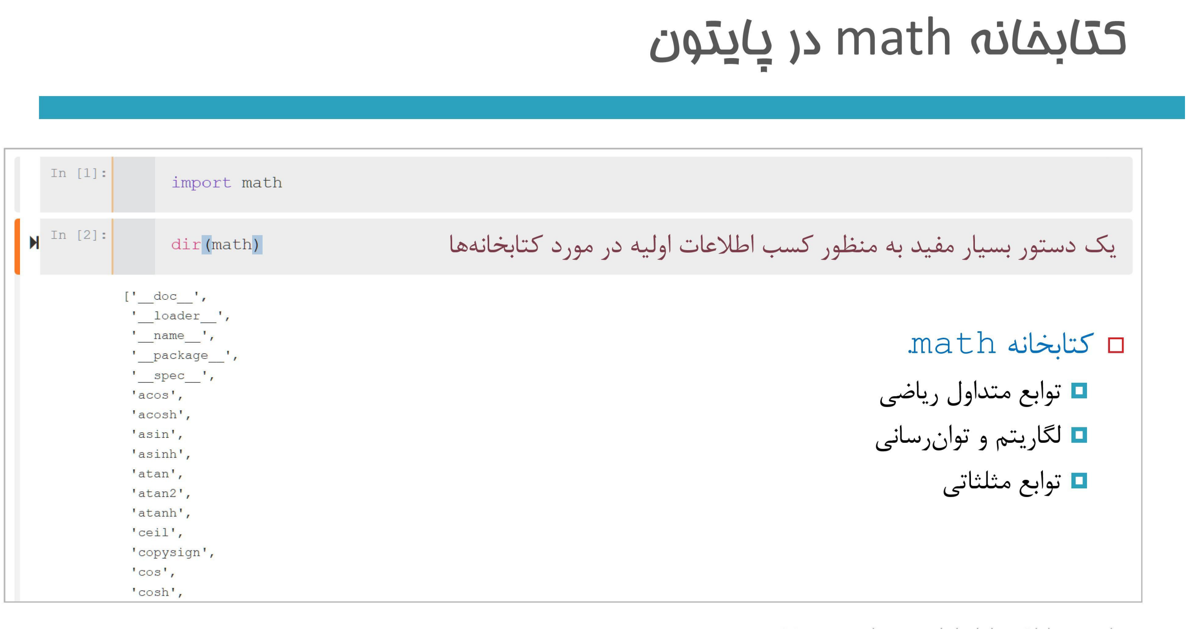
In [ ]:
```
#Every object in python is stored somewhere in memory.
#We can use id() to get that memory address.

s1 = 'amin'
s2 = 'amin'
print(id(s1)==id(s2))            # True

s1 += ' amin'

print(id(s1)==id(s2))            # False
```

In [ ]:
```
print(abs(-4))             # 4
print(pow(2,3))            # 8
print(divmod(8,4))         # (2,0)
print(round(2.6))          # 3
print(abs.__doc__)         # 'Return the absolute value of the argument.'
```



کتابخانه math در پایتون

In [1]:
```
import math
```

In [2]:
```
dir(math)
```

یک دستور بسیار مفید به منظور کسب اطلاعات اولیه در مورد کتابخانه‌ها

```
['__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
 'atan2',
 'atanh',
 'ceil',
 'copysign',
 'cos',
 'cosh',
```

- کتابخانه math
  - توابع متداول ریاضی
  - لگاریتم و توان‌رسانی
  - توابع مثلثاتی

In [ ]:
```
import math
dir(math)
```

In [ ]:

```
print('# math #')

import math
print( math.sqrt(4))       #2.0
print( math.trunc(2.7))    #2
print( math.floor(2.3))    #2
print( math.ceil(2.3))     #3
print( math.factorial(4))  #24
print( math.log2(32))      #5.0
print( math.log10(100))    #2.0
print( math.e)             #2.7
print( math.log(32))       #3.46
print( math.sin(5))        #-0.9
print( math.fmod(9,4))     #1.0
print( math.gcd(30,4))     #2
print( math.fabs(-4))      #4.0
print( abs(-4))            #4
print( math.pow(2,3))      #8.0
print( pow(2,3))           #8
print( math.pi)            # 3.141592653589793
print(f'{math.pi :.2f}')   # 3.14
```

In [ ]:

```
print('# random #')
import random
print( random.randint(1, 5))
print( random.choice([1,5]))
a = [1,2,3,4]
random.shuffle(a)
print(a)
```

In [ ]:

```
print('# datetime #')

import datetime
now = datetime.datetime.now()
print(now)                     # 2020-05-16
print( now.year)               # 2020
print( now.month)              # 2020
print( now.day)                # 16
```

In [ ]:

```
print('# sys , platform ,os #')
import sys
print( sys.version)          # 3.7.3
print( sys.platform)         # win32

import platform
platform.release()           # 10

import os
print(os.getcwd())           #'C:\Users\amin\Desktop\Python'
```

دانشگاه شهید مدنی آذربایجان
برنامه نویسی مقدماتی با پایتون

امین گلزاری اسکوئی
۱۴۰۰-۱۴۰۱

Codes and Projects (click here) (https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021) slides and videos (click here) (https://drive.google.com/drive/folders/1ZsQjBJJ4UAAp9zrGxm3c4qrhnvGBUYHw)