

▼ Модуль 1: Python-ы танилцуулга

Python нь өгөгдөлд дүн шинжилгээ хийх олон төрлийн даалгавруудыг биелүүлэх боломжтой өргөн хүрээний сантай өндөр төвшний програмчлалын хэл юм. Доорх зааварт python-ы стандарт санд байдаг олон төрлийн өгөгдлийн төрлүүд , функц, модулиудыг ашиглах жишээг оруулсан. Тухайн тэмдэглэлийг

<http://www.cse.msu.edu/~ptan/dmbook/tutorials/tutorial1/tutorial1.ipynb> эндээс татаж авч болно. Доорх жишээ дасгалуудыг алхам алхмаар нь сайтар уншаарай. Кодыг ажиллуулахын тулд доорх нүд бүр дээр дарах эсвэл SHIFT-ENTER -г зэрэг дарж ажиллуулж болно.

Python-ы тухай үндсэн мэдээллүүд:

1. Python нь C болон C++ шиг бусад өндөр төвшний хэлнүүдээс ялгаатай нь interpreter хэл юм. Кодыг эхлээд хөрвүүлэх(compile) болон холбох шаардлагагүйгээр шууд ажиллуулахаар interpreter-рүү өгөхөд болно.
2. Python нь динамик хэл бөгөөд ажиллаж байх явцдаа хувьсагчууд нь өөр өөр төрөл-рүү хувирдаг. Java , C++ болон бусад статик хэлнүүд шиг хувьсагчийн төрлийг тодорхойлж өгөх шаардлага байдаггүй.
3. Python дээр '{' '}' эдгээр хаалтуудыг ашиглахын оронд хоосон догол мөрийг ашиглаж давталт болон нөхцөлүүдийг хооронд нь бүлэглэдэг.
4. Python дээр нэг мөрийг тайлбар болгохдоо чагтыг('#') ашигладаг. Гурвал-сан хашилтыг(""") олон мөр тайлбарт эсвэл функцийн docstring-д ашигладаг.
5. Python нь нэг хувьсагчийн утгыг өөр хувьсагчид оноохдоо эсвэл функцийг оролтын утгаар объект дамжуулахдаа заалтан дамжуулагч ашигладаг(Утгыг нь дамжуулахын оронд).Тиймээс оноож өгсөн хувьсагч эсвэл функц доторх оролтын утгыг өөрчлөх нь эх объектэд нөлөөлдөг.
6. Python дээр null объектыг тэмдэглэхдээ None -г ашигладаг(a= None). Бусад хэлнүүдээс ялгаатай нь зааврыг(statement) дуусгахдаа цэг таслалыг(semicolan) ардаас нь бичих шаардлагагүй.
7. Өөр python програм файл дээр бичигдсэн хувьсагч болон функц-руу import командыг ашиглан хандаж болно. Энэ нь Java дээрх import , C болон C++ дээрх #include командтай ижил юм.

▼ 1.1 Энгийн өгөгдлийн төрлүүд

Стандарт python-гийн сан нь integer, boolean, floating point, string гэсэн энгийн өгөгдлийн төрлүүдийг дэмждэг. Доорх хүснэгтэд өгөгдлийн төрлүүдийг үзүүлэв.

	Data Type	Example
Number	Integer	x = 4
	Long integer	x = 15L
	Floating point	x = 3.142
Text	Boolean	x = True
	Character	x = 'c'

```
x = 4                # integer
print(x, type(x))

y = True             # boolean (True, False)
print(y, type(y))

z = 3.7              # floating point
print(z, type(z))

s = "This is a string"  # string
print(s, type(s))

4 <class 'int'>
True <class 'bool'>
3.7 <class 'float'>
This is a string <class 'str'>
```

Integer болон float тоонуудын арифметик үйлдлүүдийн заримыг харуулав.

```
x = 4                # integer
x1 = x + 4           # нэмэх
x2 = x * 3           # үржих
x += 2               # x = x + 2 той тэнцүү
x3 = x
x *= 3               # x = x * 3 тай тэнцүү
x4 = x
x5 = x % 4           # модуль оператор

z = 3.7              # floating point number
z1 = z - 2           # хасах
z2 = z / 3           # хуваах
z3 = z // 3          # хуваагаад бүхэл тоог нь авах
z4 = z ** 2           # зэрэгт дэвшүүлэх
z5 = z4 ** 0.5        # квадрат язгуур
z6 = pow(z,2)         # z-н квадраттай тэнцүү
z7 = round(z)         # ойролцоох бүхэл тооруу хөрвүүлнэ.
z8 = int(z)           # int-рүү хөрвүүлнэ.

print(x,x1,x2,x3,x4,x5)
```

```
print(z,z1,z2,z3,z4)
print(z5,z6,z7,z8)
```

```
18 8 12 6 18 2
3.7 1.7000000000000002 1.2333333333333334 1.0 50.653000000000006
7.117092102818398 13.690000000000001 4 3
```

math модуль доторх функцуудыг integer болон floating point дээр ашигласан жишээ.

```
import math
```

```
x = 4
print(math.sqrt(x))      # sqrt(4) = 2
print(math.pow(x,2))     # 4**2 = 16
print(math.exp(x))       # exp(4) = 54.6
print(math.log(x,2))     # log based 2 (default is natural logarithm)
print(math.fabs(-4))     # absolute value
print(math.factorial(x)) # 4! = 4 x 3 x 2 x 1 = 24

z = 1.2
print(math.ceil(z))      # ceiling function(тухайн тооны баруун талд байх бүхэл тооруу хөрвүү
print(math.floor(z))     # floor function(зүүн талд орших бүхэл тооруу хөрвүүлнэ.)
print(math.trunc(z))     # truncate function == int()

z = 3*math.pi           # math.pi = 3.141592653589793
print(math.sin(z))       # sine function
print(math.tanh(z))      # arctan function

x = math.nan             # not a number
print(math.isnan())

x = math.inf             # infinity
print(math.isinf(x))

2.0
16.0
54.598150033144236
2.0
4.0
24
2
1
1
3.6739403974420594e-16
0.9999999869751758
nan
True
```

Логик үйлдлүүдийн жишээг boolean төрөл дээр хэрэгжүүлсэн нь.

```
v1 = True
```

```

y1 = True
y2 = False

print(y1 and y2)      # logical AND
print(y1 or y2)       # logical OR
print(y1 and not y2)  # logical NOT

False
True
True

```

String дээрх үйлдлүүд.

```

s1 = "This"

print(s1[1:])          # Заасан индексээс хойших бүх үсгүүдийг хэвлэнэ.
print(len(s1))         # тэмдэгтийн уртыг авна.
print("Length of string is " + str(len(s1))) # int -г string рүү хөрвүүлэх
print(s1.upper())      # Том үсэгрүү хөрвүүлэх
print(s1.lower())      # Жижиг үсэгрүү хөрвүүлэх

s2 = "This is a string"
words = s2.split(' ')  # Тэмдэгтийг зайгаар нь хуваах
print(words[0])
print(s2.replace('a','another')) # a-г another-р солих
print(s2.replace('is','at'))     # is-г at-р солих
print(s2.find("a"))              # a үсэгний байрлалын индексийг хайх
print(s1 in s2)                  # S2-т s1 байгаа эсэхийг шалгах

print(s1 == 'This')              # equality comparison
print(s1 < 'That')               # inequality comparison
print(s2 + " too")               # string concatenation
print((s1 + " ") * 3)            # string-г 3 удаа хэвлэнэ.

his
4
Length of string is 4
THIS
this
This
This is another string
That at a string
8
True
True
False
This is a string too
This This This

```

▼ 1.2 Нийлмэл өгөгдлийн төрөл

Лист объектэн дээрх үйлдлүүд.

```

intlist = [1, 3, 5, 7, 9]
print(type(intlist))
print(intlist)
intlist2 = list(range(0,10,2)) # range[startvalue, endvalue, stepsize]
print(intlist2)

print(intlist[2]) # листийн 3 дахь элементийг авч байна.
print(intlist[:2]) # 2 дох индекс хүртэлх элементүүдийг хэвлэнэ.
print(intlist[2:]) # 2 дох индексээс цааших бүх элементүүдийг хэвлэж байна.
print(len(intlist)) # листэн доторх элементүүдийн тоо
print(sum(intlist)) # листэн дэх элементүүдийн нийлбэр

intlist.append(11) # 11 гэсэн тоог листийн төгсгөлд хийж байна.
print(intlist)
print(intlist.pop()) # сүүлийн элементийг хасаж байна.
print(intlist)
print(intlist + [11,13,15]) # листүүдийг нэгтгэж хэвлэж байна.
print(intlist * 3) # листийг 3 удаа хэвлэнэ.
intlist.insert(2,4) # 2 дох индекс дээр 4-ийн тоог оруулна.
print(intlist)
intlist.sort(reverse=True) # листийг эсрэг талруу нь хөрвүүлнэ.
print(intlist)

<class 'list'>
[1, 3, 5, 7, 9]
[0, 2, 4, 6, 8]
5
[1, 3]
[5, 7, 9]
5
25
[1, 3, 5, 7, 9, 11]
11
[1, 3, 5, 7, 9]
[1, 3, 5, 7, 9, 11, 13, 15]
[1, 3, 5, 7, 9, 1, 3, 5, 7, 9, 1, 3, 5, 7, 9]
[1, 3, 4, 5, 7, 9]
[9, 7, 5, 4, 3, 1]

mylist = ['this', 'is', 'a', 'list']
print(mylist)
print(type(mylist))

print("list" in mylist) # листэн дотор "list" гэдэг тэмдэгт мөр байгаа
                        # эсэхийг шалгах
print(mylist[2]) # 2 дох индекс дээрх элементийг харуулах
print(mylist[:2]) # 2 дох индекс хүртэлх элементүүдийг харуулах
print(mylist[2:]) # 2 дох индексээс цааших элементүүдийг харуулах
mylist.append("too") # лимтийн төгсгөлл элемент нэмэх

```

```

separator = " "
print(separator.join(mylist))    # листийн элементүүдийг нийлүүлээд string болгох

mylist.remove("is")             # "is"-г листээс устгах
print(mylist)

['this', 'is', 'a', 'list']
<class 'list'>
True
a
['this', 'is']
['a', 'list']
this is a list too
['this', 'a', 'list', 'too']

```

dictionary object-г үүсгэх болон түүн дээрх үйлдлүүд.

```

abbrev = {}
abbrev['MI'] = "Michigan"
abbrev['MN'] = "Minnesota"
abbrev['TX'] = "Texas"
abbrev['CA'] = "California"

print(abbrev)
print(abbrev.keys())           # Түлхүүрүүдийг нь авна.
print(abbrev.values())         # Утгуудыг нь авна.
print(len(abbrev))             # Түлхүүр болон утгуудын хослолын нийт тоог авах

print(abbrev.get('MI'))
print("FL" in abbrev)
print("CA" in abbrev)

keys = ['apples', 'oranges', 'bananas', 'cherries']
values = [3, 4, 2, 10]
fruits = dict(zip(keys, values))
print(fruits)
print(sorted(fruits))          # Түлхүүрээр нь сортлох(лист буцаана)

from operator import itemgetter
print(sorted(fruits.items(), key=itemgetter(0)))    # түлхүүрээр нь сортлоно.
                                                    #(dictionary буцаана.)
print(sorted(fruits.items(), key=itemgetter(1)))    # утгуудаар нь сортлоно.
                                                    #(dictionary буцаана.)

{'MI': 'Michigan', 'MN': 'Minnesota', 'TX': 'Texas', 'CA': 'California'}
dict_keys(['MI', 'MN', 'TX', 'CA'])
dict_values(['Michigan', 'Minnesota', 'Texas', 'California'])
4
Michigan
False

```

```
True
{'apples': 3, 'oranges': 4, 'bananas': 2, 'cherries': 10}
['apples', 'bananas', 'cherries', 'oranges']
[('apples', 3), ('bananas', 2), ('cherries', 10), ('oranges', 4)]
[('bananas', 2), ('apples', 3), ('oranges', 4), ('cherries', 10)]
```

Дараах жишээн дээр tuple объектийг хэрхэн үүсгэх, үйлдэл хийхийг харуулав. Tuple объектийг үүсгэсний дараа нь өөрчилж болдоггүй.

```
MItuple = ('MI', 'Michigan', 'Lansing')
CAtuple = ('CA', 'California', 'Sacramento')
TXtuple = ('TX', 'Texas', 'Austin')

print(MItuple)
print(MItuple[1:])

states = [MItuple, CAtuple, TXtuple]    # tuple-уудын листийг үүсгэнэ.
print(states)
print(states[2])
print(states[2][:])
print(states[2][1:])

states.sort(key=lambda state: state[2]) # tuple-н 2 дох индекс буюу хотын
                                         #төвийн нэрээр сортлоно.
print(states)

[<img alt="copy icon" data-bbox="91 531 111 546"/> ('MI', 'Michigan', 'Lansing')
('Michigan', 'Lansing')
[('MI', 'Michigan', 'Lansing'), ('CA', 'California', 'Sacramento'), ('TX', 'Texas', 'Au
('TX', 'Texas', 'Austin')
('TX', 'Texas', 'Austin')
('Texas', 'Austin')
[('TX', 'Texas', 'Austin'), ('MI', 'Michigan', 'Lansing'), ('CA', 'California', 'Sacram
```

▼ 1.3 Control Flow Statements

Python дээрх if, for, while-г хэрхэн ашиглах жишээнүүд.

```
# using if-else statement

x = 10

if x % 2 == 0:
    print("x =", x, "is even")
else:
    print("x =", x, "is odd")
```

```
if x > 0:
```

```

x = 10
if x > 0:
    print("x =", x, "is positive")
elif x < 0:
    print("x =", x, "is negative")
else:
    print("x =", x, "is neither positive nor negative")

x = 10
print(x, "is even")
print(x, "is positive")

# using for loop with a list

mylist = ['this', 'is', 'a', 'list']
for word in mylist:
    print(word.replace("is", "at"))

mylist2 = [len(word) for word in mylist] # үг бүрийн үсгийн тоогоор лист үүсгэх
print(mylist2)

# using for loop with list of tuples

states = [('MI', 'Michigan', 'Lansing'), ('CA', 'California', 'Sacramento'),
          ('TX', 'Texas', 'Austin')]

sorted_capitals = [state[2] for state in states]
sorted_capitals.sort()
print(sorted_capitals)

# using for loop with dictionary

fruits = {'apples': 3, 'oranges': 4, 'bananas': 2, 'cherries': 10}
fruitnames = [k for (k,v) in fruits.items()]
print(fruitnames)

that
at
a
latt
[4, 2, 1, 4]
['Austin', 'Lansing', 'Sacramento']
['apples', 'oranges', 'bananas', 'cherries']

# using while loop

mylist = list(range(-10,10))
print(mylist)

i = 0
while (mylist[i] < 0):
    i = i + 1

print("First non-negative number:", mylist[i])

```



```
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
First non-negative number: 0
```

▼ 1.4 Хэрэглэгчийн тодорхойлсон функцууд

Python дээр дурын функцийг үүсгэж болох ба нэртэй болон нэргүй функц байж болно. Нэргүй функцийг lambda гэсэн түлхүүр үгийг ашиглан тодорхойлно.

```
myfunc = lambda x: 3*x**2 - 2*x + 3      # example of an unnamed quadratic function
print(myfunc(2))
```

```
11
```

```
import math
```

```
# The following function will discard missing values from a list
def discard(inlist, sortFlag=False):    # default value for sortFlag is False
    outlist = []
    for item in inlist:
        if not math.isnan(item):
            outlist.append(item)

    if sortFlag:
        outlist.sort()
    return outlist
```

```
mylist = [12, math.nan, 23, -11, 45, math.nan, 71]
```

```
print(discard(mylist, True))
```

```
[-11, 12, 23, 45, 71]
```

▼ 1.5 File I/O

Лист болон объектоос датаг файл руу унших, бичих

```
states = [('MI', 'Michigan', 'Lansing'), ('CA', 'California', 'Sacramento'),
          ('TX', 'Texas', 'Austin'), ('MN', 'Minnesota', 'St Paul')]
```

```
with open('states.txt', 'w') as f:
    f.write('\n'.join('%s,%s,%s' % state for state in states))
```

```
with open('states.txt', 'r') as f:
    for line in f:
        fields = line.split(sep=',')    # split each line into its respective fields
```

```
fields = line.split(sep=',' ) # split each line into its respective fields  
print('State=',fields[1], '(' ,fields[0], ')', 'Capital:', fields[2])
```

State= Michigan (MI) Capital: Lansing

State= California (CA) Capital: Sacramento

State= Texas (TX) Capital: Austin

State= Minnesota (MN) Capital: St Paul