

## ▼ Модуль 1: Python талаар

Python бол өндөр түвшний програмчлалын хэл бөгөөд өгөгдөлд дүн шинжилгээ хийх янз бүрийн даалгаврыг гүйцэтгэх өргөн цар хүрээний сангуудтай. Дараах заавар нь стандарт Python номын санд байдаг төрөл бүрийн өгөгдлийн төрөл, функц, сан доторх модулиудыг ашиглах жишээг агуулсан. Дараах тэмдэглэлийг

<http://www.cse.msu.edu/~ptan/dmbook/tutorials/tutorial1/tutorial1.ipynb> энэ хаягаар татаж авах боломжтой. Доорх зааврыг алхам алхамаар анхааралтай уншина уу. Кодыг ажиллуулахын тулд тухайн ажиллуулах хэсгээ сонгоод SHIFT-ENTER товчлууруудыг зэрэг дарна уу.

Python-ны үндсэн мэдээлэл:

1. Python C, C++ зэрэг өндөр түвшний хэлнүүдээс ялгаатай интерпретатор(interpreter) программын хэл юм. Кодыг эхлээд шууд хөрвүүлэх, холбох шаардлагагүйгээр зөвхөн Python програмыг интерпретатор дээр ажиллуулахаар өгөх хэрэгтэй.
2. Python нь динамик хэл бөгөөд энэ нь хувьсагчийн нэрс програм ажиллах явцад өөр өөр төрөлд холбогддог гэсэн үг юм. Java, C ++ болон бусад статикаар бичигдсэн хэлнүүдээс ялгаатай нь хувсагч ашиглахаасаа өмнө хувьсагчийн төрлийг тодорхой зарлах шаардлагагүй.
3. Python нь '{' ба '}' хаалтын оронд хоосон догол зайг ашиглан давталт болон нөхцөлүүд дотор байрлах бүлэг үйлдлийг хийхэд ашигладаг.
4. Python нь ганц мөрт комментын өмнө('#') тэмдэгийг ашигладаг. Гурваласан давхар дуслыг('') олон мөрт комментууд(энэ нь стандарт Python хэлний хэсэг биш) болон функцийн доктринийг тэмдэглэхэд ашигладаг.
5. Python нь хувьсагчийг нөгөөд оногдуулахдаа (жишээлбэл, a = b) эсвэл функцийн оролтын аргумент болгон объектыг дамжуулахдаа дамжуулах дамжуулалтыг ашигладаг. Тиймээс хуваарилагдсан хувьсагч эсвэл функц доторх оролтын аргументыг өөрчлөх нь анхны объектод нөлөөлнө.
6. Python нь хоосон объектыг тэмдэглэхдээ `None`-ийг ашигладаг (жишээлбэл, a = None). Тайлбар бүрийг бусад хэлнээс ялгаатай төгсгөлийн тэмдэгтээр (цэг таслал гэх мэт) дуусгах шаардлагагүй.
7. Та Python програмын өөр файлд тодорхойлогдсон хувьсагч эсвэл функцуудад `import` командыг ашиглан хандаж болно. Энэ нь Java дахь `import` команд эсвэл C эсвэл C ++ хэл дээрх `# include` командтай адил юм.

### ▼ 1.1 Elementary Data Types (Үндсэн өгөгдлийн төрлүүд)

Стандарт Python-ны сан нь бүхэл тоо (integers), booleans, бутархай тоо (floating points), текст(string) зэрэг өгөгдлийн янз бүрийн төрлийг дэмжиж ажилладаг. Доорх хүснэгтээс өгөгдлийн төрлийн жишээнүүдийг харж болно.

Өгөгдлийн төрөл		Жишээ
Тоо	Integer	x = 4
	Long integer	x = 15L
	Floating point	x = 3.142
	Boolean	x = True
Текст	Character	x = 'c'
	String	x = "this" or x = 'this'

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

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

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

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

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

Доор бүхэл тоонууд ба бутархай тоонууд дээр хийгдэх зарим арифметик үйлдлүүдийг харуулав

```
x = 4                # integer (Бүхэл тоо)
x1 = x + 4           # addition (Нэмэх)
x2 = x * 3           # multiplication (Үржих)
x += 2               # equivalent to x = x + 2 (x = x + 2 бичиглэлтэй тэнцүү)
x3 = x
x *= 3               # equivalent to x = x * 3 (x = x * 3 бичиглэлтэй тэнцүү)
x4 = x
x5 = x % 4           # modulo (remainder) operator (модуль /үлдэгдэл/ оператор)

z = 3.7              # floating point number (Бутархай тоо)
z1 = z - 2           # subtraction (Хасах)
z2 = z / 3           # division (Хуваах)
z3 = z // 3          # integer division (Бүхлээр хуваах)
z4 = z ** 2          # square of z (z-ын квадрат)
z5 = z4 ** 0.5       # square root (квадрат язгуур)
z6 = pow(z,2)        # equivalent to square of z (z-ын квадраттай тэнцүү)
z7 = round(z)        # rounding z to its nearest integer (z-ийг хамгийн ойрын бүхэл тоогоор то
z8 = int(z)          # type casting float to int (int рүү төрөл хөрвүүлэх)
```

```
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 13.690000000000001
3.7 13.690000000000001 4 3
```

Математикийн модулийн бүхэл тоо ба бутархай тоотой ажиллах зарим функцууд

```
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))     # 2 суурьтай логарифм (default нь натурал логарифм)
print(math.fabs(-4))     # үнэмлэхүй утга
print(math.factorial(x)) # 4! = 4 x 3 x 2 x 1 = 24

z = 0.2
print(math.ceil(z))      # ceiling function
print(math.floor(z))     # floor function
print(math.trunc(z))     # truncate function

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

x = math.nan            # тоо биш
print(math.isnan(x))

x = math.inf            # хязгааргүй
print(math.isinf(x))
```

Boolean утгын логик үйлдлүүдийн заримыг доор харуулав

```
y1 = True
y2 = False

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

Тэмдэгт мөртэй ажиллах зарим үйлдлүүд ба функцуудыг дор харуулав

```
s1 = "This"

print(s1[1:])            # Сүүлийн 3 тэмдэгийг хэвлэх
print(len(s1))           # Тэмдэгт мөрийн уртыг авах
```

```

print("Length of string is " + str(len(s1))) # Бүхэл тоог тэмдэгт мөрлүү төрөл хөрвүүлэх
print(s1.upper())                          # том үсэгт хөрвүүлэх
print(s1.lower())                          # жижиг үсэгт хөрвүүлэх

s2 = "This is a string"
words = s2.split(' ')                     # Тэмдэгт мөрийг үг болгон 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"))                       # s2 дотроос "a"-н байрлалыг олох
print(s1 in s2)                           # s1 нь s2-ийн дэд мөр мөн эсэх

print(s1 == 'This')                       # Тэнцүү эсэхийг харьцуулах
print(s1 < 'That')                         # Тэнцэхгүй эсэхийг харьцуулах
print(s2 + " too")                        # тэмдэгт мөр залгах
print((s1 + " ") * 3)                     # тэмдэгт мөрийг 3 удаа давтана

```

```

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 Compound Data Types (Нийлмэл өгөгдлийн төрлүүд)

Жагсаалтын объектыг хэрхэн үүсгэх, удирдах талаар дараахь жишээнүүд харуулав

```

intlist = [1, 3, 5, 7, 9]
print(type(intlist))
print(intlist)
intlist2 = list(range(0,10,2))             # range[эхлэх_утга, сүүлийн_утга, шилжүүлэх_хэмжээ]
print(intlist2)

print(intlist[2])                          # Жагсаалтын гурав дахь элементийг авах
print(intlist[:2])                         # эхний хоёр элементийг авах
print(intlist[2:])                         # Жагсаалтын сүүлийн гурван элементийг авах
print(len(intlist))                       # Жагсаалтад байгаа элементүүдийн тоог авах
print(sum(intlist))                       # Жагсаалтад байгаа элементүүдийн нийлбэр авах

intlist.append(11)                         # Жагсаалтын төгсгөлд 11-г оруулах
print(intlist)
intlist.pop()                             # Жагсаалтын сүүлийн элементийг хасах
print(intlist)
print(intlist + [11,13,15])                # 2 жагсаалтыг нэгтгэх
print(intlist * 3)                         # жагсаалтыг хуулбарлах

```

```
intlist.insert(2,4)           # 2 дахь индекс дээр 4-г оруулах
print(intlist)
intlist.sort(reverse=True)    # элементүүдийг буурахаар эрэмбэлэх
print(intlist)
```

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

```
print("list" in mylist)      # mylist дотор "list" байгаа эсэхийг шалгах
print(mylist[2])             # Жагсаалтын гурав дахь элементийг харуулах
print(mylist[:2])            # Жагсаалтын эхний хоёр элементийг авах харуулах
print(mylist[2:])            # Жагсаалтын сүүлийн хоёр элементийг харуулах
mylist.append("too")         # Жагсаалтын төгсгөлд элемент оруулах
```

```
separator = " "
print(separator.join(mylist)) # Лист доторх тэмдэгт мөр элементүүдийг бүгдийг нь нэгтгэх
```

```
mylist.remove("is")          # листийн элементээс хасах
print(mylist)
```

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

Толь бичгийн объектыг хэрхэн үүсгэх, удирдах талаар дараахь жишээнүүд харуулав

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

print(abbrev)
print(abbrev.keys())          # толь бичгийн түлхүүрүүдийг авах
print(abbrev.values())        # толь бичгийн утгыг олж авах
print(len(abbrev))            # түлхүүр утгын хосын 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))         # толь бичгийн түлхүүрүүдийг эрэмбэлэх

from operator import itemgetter
```

```
print(sorted(fruits.items(), key=itemgetter(0)))    # толь бичгийн түлхүүрээр эрэмбэлэх
print(sorted(fruits.items(), key=itemgetter(1)))    # толь бичгийн утгын дагуу эрэмбэлэх
```

```
{'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)]
```

Дараахь жишээнүүд нь tuple объектыг хэрхэн үүсгэх, хэрхэн удирдахыг харуулдаг. Жагсаалтаас ялгаатай нь tuple объект өөрчлөгдөхгүй, өөрөөр хэлбэл тэдгээрийг үүсгэсний дараа өөрчлөх боломжгүй юм.

```
MITuple = ('MI', 'Michigan', 'Lansing')
CATuple = ('CA', 'California', 'Sacramento')
TXTuple = ('TX', 'Texas', 'Austin')

print(MITuple)
print(MITuple[1:])

states = [MITuple, CATuple, TXTuple]    # Энэ нь tuples жагсаалтыг үүсгэх болно
print(states)
print(states[2])
print(states[2][:])
print(states[2][1:])

states.sort(key=lambda state: state[2]) # мужуудыг нийслэл хотуудаар нь ангилах
print(states)
```

## ▼ 1.3 Удирдлагын урсгалын statement

Бусад програмчлалын хэлтэй адил Python дахь хяналтын урсгалын мэдэгдэлд if, for, while гэсэн өгүүлбэрүүд орно. Эдгээр мэдэгдлийг хэрхэн ашиглах талаархи жишээг доор харуулав.

```
# if-else ашиглах

x = 10

if x % 2 == 0:
    print("x =", x, "Тэгш")
else:
    print("x =", x, "Сондгой")

if x > 0:
```

```
    print("x =", x, "Эерэг")
elif x < 0:
    print("x =", x, "сөрөг")
else:
    print("x =", x, "эерэгч биш, сөрөгч биш")
```

```
# for давталтыг жагсаалт дээр ашиглах
```

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

```
mylist2 = [len(word) for word in mylist]    # үг бүрийн тэмдэгтийн тоо
print(mylist2)
```

```
# for давталтыг tuples жагсаалт дээр ашиглах
```

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

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

```
# for давталыг толь бичигт ашиглах
```

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

```
# while давталт ашиглах
```

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

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

```
print("Эхний сөрөг биш тоо:", mylist[i])
```

## ▼ 1.4 Хэрэглэгчийн тодорхойлсон функцууд

Та Python дээр өөрийн нэрлэсэн, нэрлээгүй функцуудыг бий болгож болно. Нэрлээгүй функцуудыг жишээний жагсаалтыг ялгахад өмнөх жишээнд үзүүлсэн шиг lambda түлхүүр үгийг ашиглан тодорхойлсон болно.

```
myfunc = lambda x: 3*x**2 - 2*x + 3    # нэрлээгүй квадрат функцийн жишээ
print(myfunc(2))
```

```
import math

# Дараах функц нь алга болсон утгуудыг жагсаалтаас хасах болно
def discard(inlist, sortFlag=False):    # sortFlag-ийн default утга нь худлаа
    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))
```

## ▼ 1.5 File I/O (Файл оролт/гаралт)

Та объект файлууд эсвэл листрүү өгөгдлийг бичиж мөн уншиж чадна.

```
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=',')    # мөр бүрийг өөрийн талбарт хуваана
        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
```



