



# ШУТИС, Мэдээлэл Холбооны Технологийн Сургууль

F.CS214 Пайтон програмчлал

Лекц - 2

Х.Хулан

# STRINGS



- Үсэг, тусгай тэмдэгт, хоосон зай, тоо
- 'эсвэл' хашилтанд тэмдэгт мөрийг хавсаргадаг

```
hi = "hello there"
```

- Тэмдэгт мөрүүдийг залгах

```
name = "ana"
```

```
greet = hi + name
```

```
greeting = hi + " " + name
```

- Зарим үйлдэл хийх

```
silly = hi + " " + name * 3
```

# INPUT/OUTPUT: print



- Console руугаргахад **OUTPUT** ашигладаг
- Түлхүүр үг нь print

```
x = 1
```

```
print(x)
```

```
x_str = str(x)
```

```
print("my fav num is", x, ".", "x =", x)
```

```
print("my fav num is " + x_str + ". " + "x = " + x_str)
```

# INPUT/OUTPUT: input("")



- Хашилтанд байгаа зүйлийг хэвлэнэ
- Хэрэглэгч ямар нэгэн юм бичээд **enter** дарна
- Тухайн утга нь хувьсагчтай холбогдоно

```
text = input("Type anything... ")  
  
print(5*text)
```

- Хэрэв тоотой ажиллах бол төрөл хувиргалт хийнэ.  
Учир нь input **танд string утга олгоно**

```
num = int(input("Type a number... "))  
  
print(5*num)
```

# COMPARISON OPERATORS ON int, float, string



- $i$  болон  $j$  нь хувьсагчын нэр
- Доорх харьцуулалт нь Boolean утга буцаана

$i > j$

$i \geq j$

$i < j$

$i \leq j$

$i == j \rightarrow$  **equality** test, True if  $i$  is the same as  $j$

$i != j \rightarrow$  **inequality** test, True if  $i$  not the same as  $j$

# LOGIC OPERATORS ON bools



- `a` болон `b` хувьсагчдын нэр (boolean утгууд)

`not a`  $\rightarrow$  True if `a` is False  
False if `a` is True

`a and b`  $\rightarrow$  True if both are True

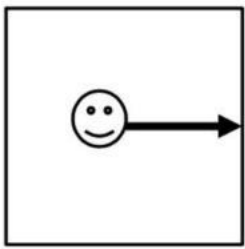
`a or b`  $\rightarrow$  True if either or both are True

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

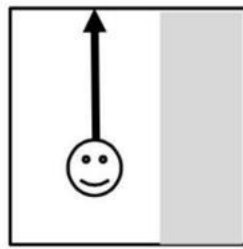
# COMPARISON EXAMPLE



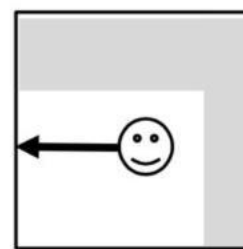
```
pset_time = 15
sleep_time = 8
print(sleep_time > pset_time)
derive = True
drink = False
both = drink and derive
print(both)
```



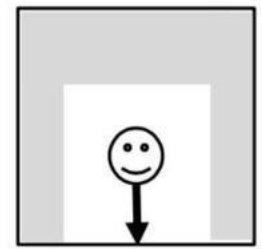
If right clear,  
go right



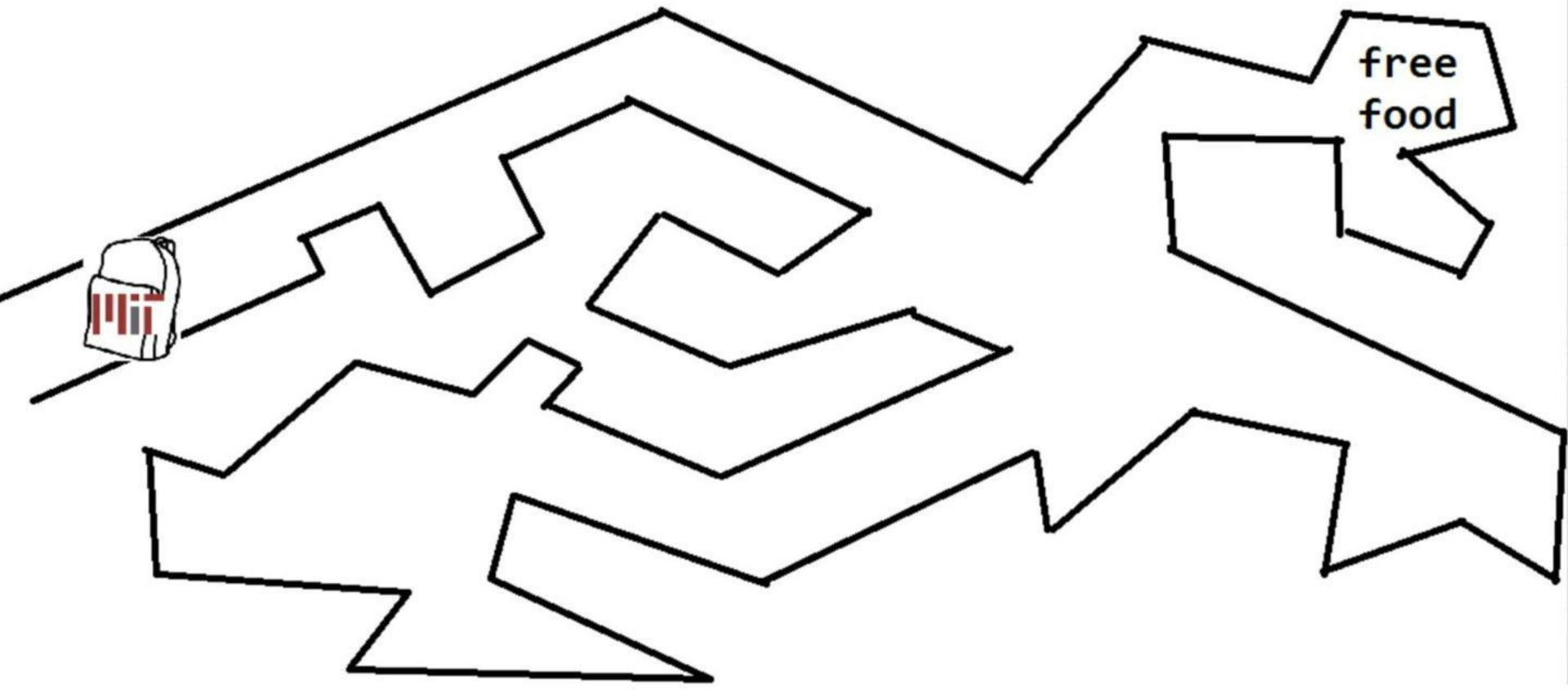
If right blocked,  
go forward



If right and  
front blocked,  
go left



If right , front,  
left blocked,  
go back





# CONTROL FLOW - BRANCHING



```
if <condition>:  
    <expression>  
    <expression>  
    ...
```

```
if <condition>:  
    <expression>  
    <expression>  
    ...  
else:  
    <expression>  
    <expression>  
    ...
```

```
if <condition>:  
    <expression>  
    <expression>  
    ...  
elif <condition>:  
    <expression>  
    <expression>  
    ...  
else:  
    <expression>  
    <expression>  
    ...
```

- <conditions> нь True эсвэл False гэсэн утгатай
- Хэрэв <conditions> нь True бол харгалзах <expression> ажиллана

# INDENTATION



- Паятонд чухал
- Кодын блокуудыг хэрхэн тэмдэглэсэн байна

```
x = float(input("Enter a number for x: "))
y = float(input("Enter a number for y: "))
if x == y:
    print("x and y are equal")
    if y != 0:
        print("therefore, x / y is", x/y)
elif x < y:
    print("x is smaller")
else:
    print("y is smaller")
print("thanks!")
```

## Python доголын дүрэм

- Бид урвуу зураас ашиглан догол мөрийг олон мөрөнд хувааж чадахгүй.
- Python кодын эхний мөрөнд доголтой байж болохгүй, энэ нь `IndentationError` шиднэ.
- Догол үүсгэхийн тулд таб болон хоосон зайг холихоос зайлсхийх хэрэгтэй. Учир нь Unix бус систем дэх текст засварлагчид өөр өөр ажилладаг бөгөөд тэдгээрийг холих нь буруу догол үүсгэх шалтгаан болдог.
- Доголын хувьд табын тэмдэгтээс илүү хоосон зайг ашиглахыг илүүд үздэг.
- Хамгийн сайн туршлага бол эхний доголд 4 хоосон зайг ашиглах ба дараа нь доголтыг нэмэгдүүлэхийн тулд нэмэлт 4 хоосон зайг үргэлжлүүлэн нэмэх явдал юм.

```
def foo():  
    print("Hi")  
    if True:  
        print("true")  
    else:  
        print("false")  
  
print("Done")
```

1st level indentation  
foo() method statements

2nd level indentation  
if and else block code

Code without indentation  
Belongs to the source file

## Python хэл дээрх доголын ашиг тус

- Ихэнх програмчлалын хэлэнд кодыг зөв бүтэцжүүлэхийн тулд догол мөрийг ашигладаг. Python-д үүнийг бүлэглэхэд ашигладаг бөгөөд кодыг автоматаар гоё болгодог.
- Python доголын дүрэм маш энгийн. Ихэнх Python IDE нь кодыг автоматаар доголдуулдаг тул зөв доголтой код бичихэд маш хялбар байдаг.

## Python дахь доголын сул талууд

- Хоосон зайг догол хийхэд ашигладаг тул хэрэв код том бөгөөд догол нь эвдэрсэн бол үүнийг засах нь үнэхээр залхмаар байдаг. Энэ нь ихэвчлэн онлайн эх сурвалж, Word документ эсвэл PDF файлаас код хуулах үед тохиолддог.

## IndentationError Examples

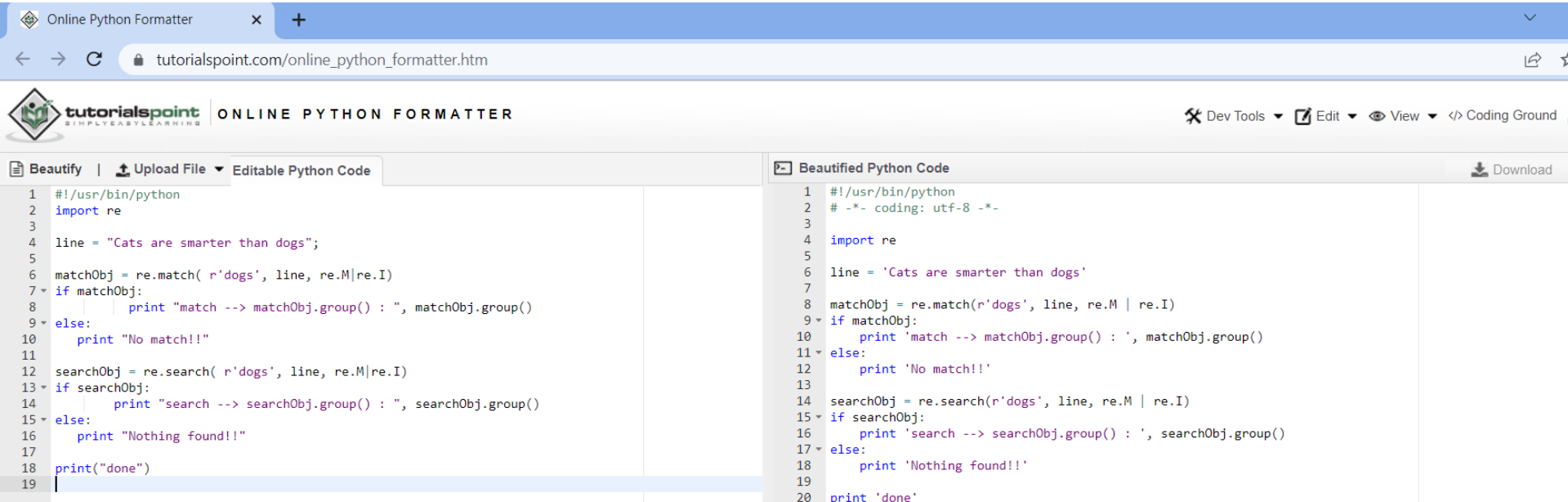
```
if True:
    print("true")
    print("Hi")
else:
    print("false")
```

```
if True:
    print("true")
else:
    print("false")

print("Done")
```

```
if True:
print("true")
else:
print("false")
```

# ONLINE PYTHON FORMATTER



The screenshot displays the 'Online Python Formatter' web application. The browser's address bar shows the URL `tutorialspoint.com/online_python_formatter.htm`. The page header includes the 'tutorialspoint' logo and the title 'ONLINE PYTHON FORMATTER'. On the right side of the header, there are links for 'Dev Tools', 'Edit', 'View', and 'Coding Ground'. The main interface is divided into two panels. The left panel, titled 'Editable Python Code', contains the following Python code:

```
1 #!/usr/bin/python
2 import re
3
4 line = "Cats are smarter than dogs";
5
6 matchObj = re.match( r'dogs', line, re.M|re.I)
7 if matchObj:
8     print "match --> matchObj.group() : ", matchObj.group()
9 else:
10    print "No match!!"
11
12 searchObj = re.search( r'dogs', line, re.M|re.I)
13 if searchObj:
14     print "search --> searchObj.group() : ", searchObj.group()
15 else:
16     print "Nothing found!!"
17
18 print("done")
19
```

The right panel, titled 'Beautified Python Code', shows the same code after formatting, with proper indentation and single quotes for string literals:

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3
4 import re
5
6 line = 'Cats are smarter than dogs'
7
8 matchObj = re.match(r'dogs', line, re.M | re.I)
9 if matchObj:
10     print 'match --> matchObj.group() : ', matchObj.group()
11 else:
12     print 'No match!!'
13
14 searchObj = re.search(r'dogs', line, re.M | re.I)
15 if searchObj:
16     print 'search --> searchObj.group() : ', searchObj.group()
17 else:
18     print 'Nothing found!!'
19
20 print 'done'
```

Buttons for 'Beautify', 'Upload File', and 'Download' are visible at the top of the code panels.

[https://www.tutorialspoint.com/online\\_python\\_formatter.htm](https://www.tutorialspoint.com/online_python_formatter.htm)

**= VS ==**



```
x = float(input("Enter a number for x: "))
```

```
y = float(input("Enter a number for y: "))
```

```
if x == y:
```

```
    print("x and y are equal")
```

```
    if y != 0:
```

```
        print("therefore, x / y is", x/y)
```

```
elif x < y:
```

```
    print("x is smaller")
```

```
else:
```

```
    print("y is smaller")
```

```
print("thanks!")
```

хэрэв  $x=y$  байвал яах вэ?  
SyntaxError үүснэ



- Legend of Zelda
- Баруун тийш яв, яг ижилхэн дэлгэц үзэгдэнэ

```
if <exit right>:
```

```
    <set background to woods_background>
```

```
    if <exit right>:
```

```
        <set background to woods_background>
```

```
        if <exit right>:
```

```
            <set background to woods_background>
```

```
            and so on and on and on...
```

```
        else:
```

```
            <set background to exit_background>
```

```
    else:
```

```
        <set background to exit_background>
```

```
else:
```

```
    <set background to exit_background>
```






- Legend of Zelda
- Баруун тийш яв, яг ижилхэндэлгэц үзэгдэнэ

```
while <exit right>:
```

```
<set background to woods_background>
```

```
<set background to exit_background>
```

# CONTROL FLOW: while LOOPS



```
while <condition>:  
    <expression>  
    <expression>  
    ...
```

- <condition> boolean утгаар үнэлнэ
- Хэрэв <condition> нь True бол while давталт дахь бүх блокыг гүйцэтгэнэ
- дахиад <condition> шалгана
- <condition> нь False болтол давтана

# While loop EXAMPLE



```
You are in the Lost Forest.  
*****  
*****  
  😊  
*****  
*****  
Go left or right?
```

## PROGRAM:

```
n = input("You're in the Lost Forest. Go left or right? ")  
while n == "right":  
    n = input("You're in the Lost Forest. Go left or right? ")  
print("You got out of the Lost Forest!")
```

# CONTROL FLOW: while and for LOOPS



- Тоонуудыг дарааллаар нь бичнэ

```
# more complicated with while loop  
n = 0
```

```
while n < 5:  
    print(n)  
    n = n+1
```

```
# shortcut with for loop  
for n in range(5):  
    print(n)
```

# CONTROL FLOW: for LOOPS



```
for <variable> in range(<some_num>) :  
    <expression>  
    <expression>  
    ...
```

- Далталтын тохиолдол бүрт, <variable> утга авна
- Эхний удаа, <variable> хамгийн бага утгаас эхлэнэ
- Дараагийн удаа, <variable> нь өмнөх утга дээр + 1
- Гэх мэт ...

## range (start, stop, step)



- Анхны утга нь start=0, step=1 бөгөөд заавал биш
- Хүртэл давтах утга нь stop-1 байна

```
mysum = 0
for i in range(7, 10):
    mysum += i
print(mysum)
```

```
mysum = 0
for i in range(5, 11, 2):
    mysum += i
print(mysum)
```

# Break STATEMENT



- Тухайн давталтнаас нэн даруй гарна
- Кодын блок дахь үлдсэн илэрхийллийг алгасна
- Зөвхөн дотоод давталтнаас гарна

```
while <condition_1>:  
    while <condition_2>:  
        <expression_a>  
        break  
        <expression_b>  
    <expression_c>
```

# Break STATEMENT



```
mysum = 0
for i in range(5, 11, 2):
    mysum += i
    if mysum == 5:
        break
    mysum += 1
print(mysum)
```

- Үр дүн юу гарах вэ?



# for

# VS

# while LOOPS



## for давталт

- Давталтын тоог **мэднэ**
- Break-р **эрт зогсоож** болно
- **Тоолуур** хэрэглэдэг
- While давталтаар for давталыг **хийж болно**

## while давталт

- Давталын тоо **мэдэгдэхгүй**
- Break-р **эрт зогсоож** болно
- Тоолуур хэрэглэж болно, гэхдээ **эхлээд заавал анхны утга олгоно**
- **Бараг боломжгүй**, for давталтаар while давталт хийх

