Loops

- while-loops
- Conditionals
- Operators
- Comparison sequences

Arithmetic operators

Assume variable **a** holds the value 10 and variable **b** holds the value 20, then

	Operator	Description	Example	Shortcut
+	Addition	Adds values on either side of the operator.	a + b = 30	x += y
-	Subtraction	Subtracts right hand operand from left hand operand.	a – b = -10	x -= y
*	Multiplication	Multiplies values on either side of the operator	a * b = 200	x *= y
/	Division	Divides left hand operand by right hand operand	b / a = 2.0	x /= y
%	Modulus	Divides left hand operand by right hand operan d and returns **remainder**	b % a = 1	x %= y
**	Exponent	Performs exponential (power) calculation on operators	a**b =10 to t he power 20	x **= y
//	Floor Division	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., r ounded away from zero (towards negative infini ty):	9//2 = 4 9.0//2.0 = 4.0 -11//3 = -4 -11.0//3 = -4.0	x //= y

https://www.tutorialspoint.com/python3/arithmetic_operators_example.htm

NAZARBAYEV UNIVERSITY

Comparison operators

- Comparison operators are used to compare values.
  It either returns **True** or **False** according to the condition.

Operator	Meaning	Example
>	Greater that - True if left operand is greater than the right	x > y
<	Less that - True if left operand is less than the right	x < y
==	Equal to - True if both operands are equal	x == y
!=	Not equal to - True if operands are not equal	x != y
>=	Greater than or equal to - True if left operand is greater than or equal to the right	x >= y
<=	Less than or equal to - True if left op erand is less than or equal to the right	x <= y

NAZARBAYEV UNIVERSITY

Examples

```
x = 10
y = 12

>>> print('x > y  is', x > y)

>>> print('x < y  is', x < y)

>>> print('x == y is', x == y)

>>> print('x != y is', x != y)

>>> print('x >= y is', x >= y)

>>> print('x <= y is', x <= y)
```

NAZARBAYEV UNIVERSITY

Examples

```
x = 10
y = 12

>>> print('x > y  is', x > y)
False
>>> print('x < y  is', x < y)
True
>>> print('x == y is', x == y)
False
>>> print('x != y is', x != y)
True
>>> print('x >= y is', x >= y)
False
>>> print('x <= y is', x <= y)
True
```

NAZARBAYEV
UNIVERSITY

Logical operators

Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not

https://www.programiz.com/python-programming/operators

Special operators

- **Identity operators**
  - *is* and *is not* are the identity operators in Python.
  - They are used to check if two values (or variables) are located on the **same part of the memory**.
  - Two variables that are equal does not imply that they are identical.

Operator	Meaning	Example
is	True if the operands are identical (refer to the same object)	x is True
is not	True if the operands are not identic al (do not refer to the same object)	x is not True

Special operators

- **Membership operators**
  - *in* and *not in* are the membership operators in Python.
  - They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

Operator	Meaning	Example
In	True if value/variable is found in the sequence	5 in x
not in	True if value/variable is not found in the s equence	5 not in x

https://www.programiz.com/python-programming/operators

NAZARBAYEV UNIVERSITY

Examples

```
x = True
y = False

>>> x and y

>>> x or y

>>> not x
```

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]

>>> x1 is not y1

>>> x2 is y2

>>> x3 is y3
```

```
x = 'Hello world'
y = {1:'a',2:'b'}

>>> 'H' in x

>>> 'hello' not in x

>>> 1 in y

>>> 'a' in y
```

Examples

```
x = True
y = False

>>> x and y
False
>>> x or y
True
>>> not x
False
```

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]

>>> x1 is not y1

>>> x2 is y2

>>> x3 is y3
```

```
x = 'Hello world'
y = {1:'a',2:'b'}

>>> 'H' in x

>>> 'hello' not in x

>>> 1 in y

>>> 'a' in y
```

Examples

```
x = True
y = False

>>> x and y
False
>>> x or y
True
>>> not x
False
```

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]

>>> x1 is not y1
False
>>> x2 is y2
True
>>> x3 is y3
False
```

```
x = 'Hello world'
y = {1:'a',2:'b'}

>>> 'H' in x

>>> 'hello' not in x

>>> 1 in y

>>> 'a' in y
```

Examples

```
x = True
y = False

>>> x and y
False
>>> x or y
True
>>> not x
False
```

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]

>>> x1 is not y1
False
>>> x2 is y2
True
>>> x3 is y3
False
```

```
x = 'Hello world'
y = {1:'a',2:'b'}

>>> 'H' in x
True
>>> 'hello' not in x
True
>>> 1 in y
True
>>> 'a' in y
False
```

NAZARBAYEV UNIVERSITY

A note on (in)equality operators

- The equality == and inequality != operators ignore the types of data to a certain extent. For example,

```
>>> False == 0
>>> False != 0
>>> False == 3
>>> True == -1
>>> Ture != 1
```

NAZARBAYEV UNIVERSITY

while-loop

- While-loop syntax has the following elements
- As with the for-loop, **do not forget the colon**!

1. start with the keyword "while",
2. followed by the condition that has to be checked
3. then a colon ":".

```
while conditions:
    codes (loop body)
```

for-loop

- The most common type is a for-loop.
- It executes some part of the code for predetermined number of times.

1. start with the keyword "`for`",
2. followed by the name of the variable that will be assigned

   all the values through which we want to loop
3. then the keyword "`in`",
4. then a list or something that acts like it
5. then a colon "`:`".

```
for varaible in items:
    codes (loop body)
```

while-loop

- While-loop syntax has the following elements
- As with the for-loop, **do not forget the colon**!

1. start with the keyword "while",
2. followed by the condition that has to be checked
3. then a colon ":".

While i is less than 5, print the text repetitively

```
i = 0
while i < 5:
    print(‘hello world’)
    i = i+1
```

NAZARBAYEV UNIVERSITY

Break the loops

- We can break loop earlier than prefixed # of iterations

```
i = 0
while i < 500:
    print(i)
    if i == 5:
        break
```

```
For i in range(500):
    print(i)
    if i == 5:
        break
```

NAZARBAYEV UNIVERSITY

for-loop

```
for i in [0, 1, 2, 3, 4]:
    print(i)
```

```
a = [0, 1, 2, 3, 4]
for i in a:
    print(i)
```

```
for i in range(5):
    print(i)
```

```
for i in range(4, -1, -1):
    print(i)
```

Output:

```
0
1
2
3
4
```

Output:

```
4
3
2
1
0
```

NAZARBAYEV UNIVERSITY

for loop -> while loop

- **Left-shifting a string**
- All letters are concatenated in opposite direction

```
a = 'abcdef'
t2 = '';
for i in range(len(a)-1, -1, -1):
    t2 = t2 + a[i]
    print(t2)
```

```
a = 'abcdef'
t = ''
for i in a:
    t = i + t
    print(t)
```

for loop -> while loop

- Iterating through a list

```
s = ['kor', 'kz', 'USA', 'jp']
s2 = ‘’.join(s)
for c in range(len(s2)):
    print(s2[ c ])
```

```
s = ['kor', 'kz', 'USA', 'jp']
for c in s:
    for j in range(len(c)):
        print(c[ j ])
```

Conditional - True or False

```
a = 'aDbAc1d5ef'
count = 0;
for i in a:
    if i.isdigit():
        print(i)
        count = count +1
print(count)

count = 0;
for i in a:
    if i.isupper():
        print(i)
        count = count +1
print(count)
```

```
a = input('Enter any integer number')

if int(a) % 2 == 0:

    print(a, 'is even number')

else:

    print(a, 'is odd number')
```

```
odd_n = list()
even_n = list()
for i in range(100):
    if i % 2 != 0:
        odd_n.append(i)
    else:
        even_n.append(i)
```

NAZARBAYEV
UNIVERSITY

Conditionals - True or False

```
a = 'aDbAc1d5ef'
count = 0;
for i in a:
    if i.isdigit():
        print(i)
        count = count +1
print(count)

count = 0;
for i in a:
    if i.isupper():
        print(i)
        count = count +1
print(count)
```

```
a = input('Enter any integer number')

if int(a) % 2 == 0:

    print(a, 'is even number')

else:

    print(a, 'is odd number')
```

```
odd_n = list()
even_n = list()
for i in range(100):
    if i % 2 != 0:
        odd_n.append(i)
    else:
        even_n.append(i)
```

NAZARBAYEV
UNIVERSITY