NIS Oskemen

Part of curriculum: **Unit 11.1A: Basic structures of the Python programming language**

Nazarbayev Intellectual Schools

Introduction to the Python programming language. Organizing data output

LO, which will be achieved in this lesson (link to Curriculum):

- ❑ organize data output
- ❑ use the escape sequences with data output

Nazarbayev Intellectual Schools

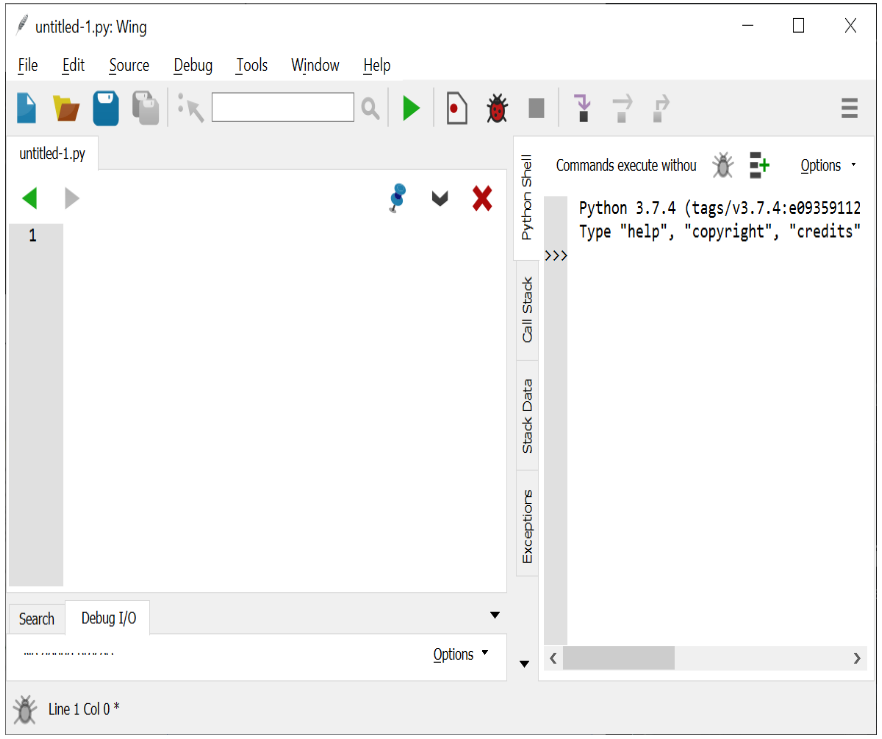
Lesson objectives:

~ to be able to use output commands

~ to be able use different type of output operations

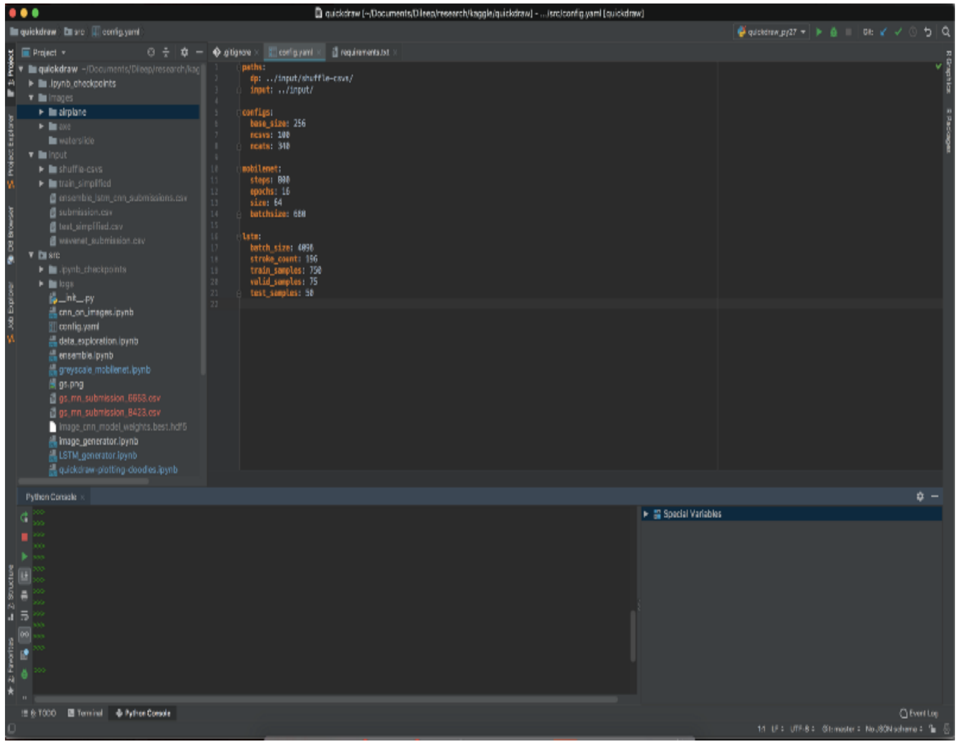
Also …

Wing IDE



```
Python 3.7.4 (tags/v3.7.4:e09359112
Type "help", "copyright", "credits"
>>>
```

PyCharm



```
paths:
  dp: ../input/shuffle-csvs/
  input: ../input/

configs:
  base_size: 256
  ncsvs: 100
  ncats: 340

mobilenet:
  steps: 800
  epochs: 16
  size: 64
  batchsize: 680

lstm:
  batch_size: 4096
  stroke_count: 196
  train_samples: 750
  valid_samples: 75
  test_samples: 50
```

For teaching a programming language

For an application developer

Resources

- https://ru.code-basics.com/languages/python/lessons/escape-characters
- https://www.w3schools.com/python/default.asp
- https://www.tutorialspoint.com/python/index.htm
- https://pyprog.pro/python/py/str/esqape_sec.html

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on **indentation**, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Creating a Comment

- *Comments can be used to explain Python code.*
- *Comments can be used to make the code more readable.*
- *Comments can be used to prevent execution when testing code.*

```
#This is a comment
print("Hello, World!")
```

```
"""
This is a comment
written in
more than just one line
"""
print("Hello, World!")
```

Output data

```
print("Hello, world!")      # the correct representation

print('Hello, world!')      # the correct representation

print("Hello, world!')      # erroneous representation
```

The output arguments are sep and end.

```
print("text1", "text2", "text3")          # the output has two arguments
```

Result :

```
text1text2text3
```

Using the sep (separator) argument

```
print("text1", "text2", "text3", sep="---")
```

Result :

```
text1---text2---text3
```

Using the end argument

```
print("text1", end=" ")
print("text2", end=" ")
print("text3")
```

Result :

```
text1 text2 text3
```

Escape characters

\\\\	Backslash
\\'	Single Quote
\\"	Quotation mark
\n	New Line
\t	Tab
\u …	16-bit Unicode character in 16-bit representation
\U …	A 32-bit Unicode character in a 32-bit representation
\x…	Hex value

```
print("text1\n text2")
```

Result :

```
text1

text2
```

****

```
print("Carrot\t150 tenge\n Beet \t180 тенге")
```

Result :

```
Carrot              150 tenge

Beet                180 tenge
```

****

Determine what happens when the next line is output

```
print("\u1D66")
```

Simple arithmetic.

a + b	Addition: adds two operands
a - b	Subtraction: subtracts two operands
a * b	Multiplication: multiplies two operands
a / b	Division (float): divides the first operand by the second
a // b	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed.
a % b	Modulus: Divides left hand operand by right hand operand and returns remainder
a ** b	Exponent: Performs exponential (power) calculation on operators
round(f, 2)	Rounding the number f to the second decimal place (to hundredths)
round(f)	Rounding the number f to integers

Composite assignment operator

x = x + y	x += y
x = x - y	x -= y
x = x * y	x *= y
x = x / y	x /= y
x = x // y	x //= y
x = x % y	x %= y
x = x ** y	x **= y

Multiple assignment

a, b = 5, 7   # variable a will store the number 5, and variable b will store the number 7

a, b = b, a   # variable a will store the number 7, and variable b will store the number 5

Determine the result :

```
a, b, c = 3, 2, 1
b, a, c, a, b
print(b, c, a) # Determine THE RESULT
```

Multiple assignment

a, b = 5, 7 # variable a will store the number 5, and variable b will store the number 7

a, b = b, a # variable a will store the number 7, and variable b will store the number 5

Determine the result :

```
a, b, c = 3, 2, 1
b, a, c, a, b
print(b, c, a) # 1 2 3
```

Data entry. Simple functions.

int("text") – converts a string to an integer

float("text") – converts a string to a real number

str(num) – converts a number to a string

abs(num) – absolute value of a number (modulus of a number)

len("text") – string length (number of characters per line)

Example. Determine the distance between two points on the coordinate line.

Data entry. Simple functions.

int("text") – converts a string to an integer

float("text") – converts a string to a real number

str(num) – converts a number to a string

abs(num) – absolute value of a number (modulus of a number)

len("text") – string length (number of characters per line)

Example. Determine the distance between two points on the coordinate line.

```
point1 = float(input(" Enter the coordinate of the first point :"))

point2 = float(input(" Enter the coordinates of the second point :"))

s = point2 - point1

print(" Distance between points =", abs(s))
```

1) Why is the float type used in this task, and not int?

2) Why use the abs() function when displaying the distance between two points?

Define the data output

```
print(str(10) + str(20))

print(int('10') + int('20'))

print(len('10') + len('20'))
```

Define the data output

```
print(str(10) + str(20))          # will output '1020'
print(int('10') + int('20’))      # will output 30
print(len('10') + len('20’)) # will output 4
```

Questions

Task 1

Assume variable a holds 21 and variable b holds 10, then –

```
a + b
a - b
a * b
a / b
a % b
a**b
a//b
```

Task 2

Write programming code, that will count

- 6 + 4
- ((27 * 2) + 46) ** 0.5
- Find the modulus of 35 / 6

Task 3

Write programming code, that will count

$$1)\ z1 = \frac{(x+y)^3}{x^2 - 5xy + y^3};$$

$$2)\ z2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a};$$

NIS Oskemen

Part of curriculum: **Unit 11.1A: Basic structures of the Python programming language**

Nazarbayev Intellectual Schools

Data types. Data input

LO, which will be achieved in this lesson (link to Curriculum):

- ❑ distinguish between data types in Python
- ❑ convert data types of variables
- ❑ organize keyboard inputs

Nazarbayev Intellectual Schools

Lesson objectives:

~ to be able to use input commands

~ to be able define data types

C++	Python	Pascal

#include <iostream>
using namespace std;
int main()
{
  string username;
  cout << "What is your name? \n";
  cin >> username;
  cout << "Hello, " << username <<
"!";
}
	print(‘What is your name?’)
	username = input()
	print(‘Hello, ’, username, ‘!’)
		var username: string;
		begin
		 writeln('What is your
		name?');
		 readln(username);
		 writeln('Hello, ', username,
		'!');
		end.

Which language is easier to understand the program in? …………………………………………………

In which language will the program be written faster than in other languages?

Repetition

You have a line of program code:

print ("15:00 movie Terminator 16:30 The game Field of Miracles ")

Using escaped characters, get the following output option:

```
15:00          movie "Terminator "
16:30          the game  "Field of Miracles"
```

USEFUL LINKS

- ❑ **V.N. Pilshchikov - Collection of exercises on the Pascal language, pp. 10-11.:**

  https://studizba.com/files/show/djvu/569-1-v-n-pil-schikov--sbornik-uprazhneniy-po.html

- ❑ **Converting Data Types in Python 3:**

  https://pythonist.ru/preobrazovanie-tipov-dannyh-v-python-3/

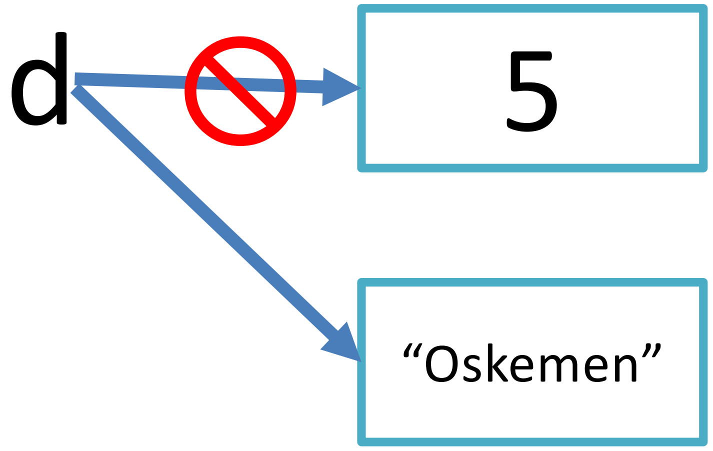
- ❑ **Read input as a float in Python:**

  https://www.includehelp.com/python/read-input-as-a-float.aspx

Dynamic typing

- a method used in programming languages in which a variable is associated with a type at the time of assigning a value, and not at the time of declaring a variable.

```
d = 5
print(type(d))              # <class 'int'>
d = ”Oskemen"
print(type(d))              # <class 'str'>
```



Dynamic typing

Define the type of variables

```
lang = "Python"
print( type(lang) )
amount = 15
print( type(amount) )
height = 1.78
print( type(height) )
```

Dynamic typing

Define the type of variables

```
lang = "Python"
print(type(lang))                  # <class 'str'>
amount = 15
print(type(amount))    # <class 'int'>
height = 1.78
print(type(height))      #<class 'float'>
```

Multi-purpose purpose. Inputting numbers in a single line.

```
a, b = int(input()), int(input()) # entering 5 7 in one line will result in an error
```

If you need to enter two numbers in the program, then you should do the following:

```
line = input()
num1, num2 = line.split()
num1 = int(num1)
num2 = int(num2)
```

also, these three lines can be replaced with one :

```
num1, num2 = map(int, input().split())
```

The *map* function, which applies another function (in our case, int) to each part obtained after splitting the entered string into parts and creates a "map" of numbers

The task "Speed of a cyclist"

- The cyclist drove the distance **S** km in **t** hours. Write a program that calculates the speed of a cyclist **v**.

The task "Speed of a cyclist"

- The cyclist drove the distance **S** km in **t** hours. Write a program that calculates the speed of a cyclist **v**.

```
S = float(input())
t = float(input())
v = S / t
print(round(v, 2))
```

Explain the purpose of each line.

What happens when t is equal to 0?

Determine the output results. If the action cannot be performed, then enter "error“ in the response

- print(12 + int("8"))
- print(str(2.4) + "29")
- print(int(4.5) + 5)
- print(float("1.5") + 5)
- print(int("4.5") + 1)

Task "Simple addition“

Write a simple program for adding two integers a and b.

Input data: Two integers per line.

Output: The result of the sum of two numbers.

Task " **Average value**“

At KVN competitions, the team receives five ratings from different judges. Write a program to calculate the average score of the team.

Input data: Five integers.

Output data: The average score of the team.

Define the results

- round(4.2) =
- round(4.6) =
- round(1.234, 1) =
- round(5.728, 2) =
- round(5 / 7, 2) =

- 12 // 7 =
- 21 % 8 =
- 156 % 10 =
- 238 % 100 =
- 876 // 10 =
- 907 // 100 =
- 439 % 1000 =
- 191 // 1000 =