



Unit 11.1 A Basic structures of the Python programming language.



Заголовок слайда

- 1 Introduction to Python. Output data
- 2 Data types. Input data
- 3 Linear and branching algorithms. Selection
- 4 Loop algorithms. Loop For
- 5 Loop While
- 6 Nested loops

Execute Python Syntax



- Python syntax can be executed by writing directly in the Command

```
>>> print("Hello, World!")  
Hello, World!
```

- Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```



Escaped sequences

Escaped sequences are sequences that start with the character `"\"` followed by one or more characters.

```
main.py x
1 print("Gregor Clegane\n Dunsen\n Polliver\n Chiswyck")
2 print(4, 5, 6, sep=', ', end='. ')
3
```

Console Shell

```
Gregor Clegane
Dunsen
Polliver
Chiswyck
4, 5, 6. ✎
```



Escaped sequences

<code>\\</code>	Allows you to write a backslash character.
<code>\'</code>	Allows you to write a single apostrophe character
<code>\"</code>	Allows you to write a single quote character.
<code>\n</code>	Line break (new line).
<code>\r</code>	Returns the cursor to the beginning of the line.
<code>\t</code>	Horizontal indentation to the left of the beginning of the line (horizontal tab).
<code>\v</code>	Vertical indentation at the top (vertical tab).

https://pyprog.pro/python/py/str/esqape_sec.html



WHAT IS A VARIABLE?

Variables are containers for storing data values.

How variables are declared in C++?

What is a variable type?

```
int a; // объявление переменной a целого типа.  
float b; // объявление переменной b типа данных с плавающей запятой.  
double c = 14.2; // инициализация переменной типа double.  
char d = 's'; // инициализация переменной типа char.  
bool k = true; // инициализация логической переменной k.
```

Python Variables



- Python has **no command** for declaring a variable.
- Variables **do not need to be declared** with any particular *type*

```
a = 4
A = "Sally"
#A will not overwrite a
```

main.py x

```
1 x = 4          # x is of type int
2 print(x)
3 x = "Sally"   # x is now of type str
4 print(x)
```

Console

Shell

```
4
Sally
> █
```



Built-in Data Types

Text Type: `str`

Numeric Types: `int`, `float`, `complex`

Sequence Types: `list`, `tuple`, `range`

Mapping Type: `dict`

Set Types: `set`, `frozenset`

Boolean Type: `bool`



Python Logical Operators

Operator	Description	Example
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>



Logical operators

- Operators are used to perform operations on values and variables. These are the special symbols that carry out arithmetic and logical computations. The value the operator operates on is known as **Operand**.
- In Python, Logical operators are used on conditional statements (either True or False). They perform **Logical AND**, **Logical OR** and **Logical NOT** operations.

OPERATOR	DESCRIPTION	SYNTAX
and	Logical AND: True if both the operands are true	x and y
or	Logical OR: True if either of the operands is true	x or y
not	Logical NOT: True if operand is false	not x



If you have only one statement to execute, you can put it on the same line as the if statement.

```
if a > b: print("a is greater than b")
```

If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:

```
a = 2  
b = 330  
print("A") if a > b else print("B")
```

You can also have multiple else statements on the same line:

```
a = 330  
b = 330  
print("A") if a > b else print("=") if a == b else print("B")
```

Short Hand If ... Else



Task1

1. Write a Python program that will: (3 pts)
 - Ask the user for seven numbers
 - Print the total sum of the numbers
 - Print the count of the positive entries, the number entries equal to zero, and the number of negative entries. Use an if, elif, else chain, not just three if statements.



And Or

The and keyword is a logical operator, and is used to combine conditional statements:

```
a = 200
b = 33
c = 500
if a > b and c > a:
    print("Both conditions are True")
```

The or keyword is a logical operator, and is used to combine conditional statements:

```
a = 200
b = 33
c = 500
if a > b or a > c:
    print("At least one of the conditions is True")
```



The pass Statement

if statements cannot be empty, but if you for some reason have an if statement with no content, put in the pass statement to avoid getting an error.

```
a = 33
b = 200

if b > a:
    pass
```



For loops

Example:

```
>>> #The list has four elements, indices start at 0 and end at 3
>>> color_list = ["Red", "Blue", "Green", "Black"]
>>> for c in color_list:
        print(c)
        Red
        Blue
        Green
        Black
>>>
```

In the above example `color_list` is a sequence contains a list of various color names. When the for loop executed the first item (i.e. Red) is assigned to the variable `c`. After this, the print statement will execute and the process will continue until we reach the end of the list.



Python for loop and range() function

Syntax:

```
for "variable" in range("start_number",  
"end_number"):
```

Example:

```
>>> for a in range(2,7):  
    print(a)
```

```
2  
3  
4  
5  
6  
>>>
```

Example:

```
>>> for a in range(2,19,5):  
    print(a)
```

```
2  
7  
12  
17  
>>>
```

range(a,b,c): Generates a sequence of numbers from a to b excluding b, incrementing by c.



The **range()** Function

To loop through a set of code a specified number of times, we can use the **range()** function, this function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.



The **while** Loop

With the while loop we can execute a set of statements as long as a condition is true.

Example

Print *i* as long as *i* is less than 6:

```
i = 1
while i < 6:
    print(i)
    i += 1
```

Note: remember to increment *i*, or else the loop will continue forever.

The while loop requires relevant variables to be ready, in this example we need to define an indexing variable, *i*, which we set to 1.

Task2



Solve the square equation. Find the roots of the square equation and display them on the screen if they exist. If there is no root, then send a message about it. The specific quadratic equation is determined by the coefficients a , b , c , which are entered by the user.





TASK 3-4

1. Create a program that finds the average value of a list of numbers using a for loop.
2. Write a program that finds the smallest element in a list of numbers using a for loop.



- Task 5 Write a program that asks for a password and keeps asking until the correct password, apple is entered and then says Accepted.
- Task 6 The sequence 1,4,9,16,25 is made up of square numbers (i.e. $1=1^2$, $4=2^2$, $9=3^2$ etc.). Write a program that writes out all the square numbers under 5000.

TASK 7



- Ruslan decided to start drawing with ASCII characters.
- Help Ruslan write a program for constructing a **height** x **width** rectangle consisting of **symbols**.
- The shape should only contain an outline.

Sample Input 1:

```
7
10
*
```

Sample Output 1:

```
*****
*           *
*           *
*           *
*           *
*           *
*           *
*****
```


8.25. Найти количество делителей каждого из целых чисел от 120 до 140.

8.26. Составить программу для графического изображения делимости чисел от 1 до n (значение n вводится с клавиатуры). В каждой строке надо напечатать очередное число и столько символов "+", сколько делителей у этого числа. Например, если $n=4$, то на экране должно быть напечатано:

1+

2++

8.27. Найти все целые числа из промежутка от 1 до 300, у которых ровно пять делителей.

8.28. Найти все целые числа из промежутка от 200 до 500, у которых ровно шесть делителей.

8.29. Найти все целые числа из промежутка от a до b , у которых количество делителей равно k .

8.30. Найти натуральное число из интервала от a до b , у которого количество делителей максимально. Если таких чисел несколько, то должно быть найдено:

а) максимальное из них;

б) минимальное из них.

8.31. Найти все трехзначные простые числа (*простым* называется натуральное число, большее 1, не имеющее других делителей, кроме единицы и самого себя).

8.32. Найти 100 первых простых чисел.

8.33. Найти сумму делителей каждого из целых чисел от 50 до 70.

8.34. Найти все целые числа из промежутка от 100 до 300, у которых сумма делителей равна 50.

8.35. Найти все целые числа из промежутка от 300 до 600, у которых сумма делителей кратна 10.

8.36. Натуральное число называется *совершенным*, если оно равно сумме своих делителей, включая 1 и, естественно, исключая это самое число. Например, совершенным является число 6 ($6=1+2+3$). Найти все совершенные числа, меньшие 100 000.

8.37. Найти натуральное число из интервала от a до b с максимальной суммой делителей.

8.38. Два натуральных числа называются дружественными, если каждое из них

ASK 4

39-i

Advanced tasks