



STATWORX TRAINING

PYTHON BASICS – CHEAT SHEET

Basics

IN & OUTPUT

```

1 print(object) - # converts any object to
                  string and prints it to
                  screen
2 #this is a comment
3 '''This is a multiline comment'''
4 x = 1.674 - # Variable
5 y = „Hello“ - # Assignment
6 del x - # Object Deletion
7 input(prompt) - # gets input from the user
8 raw input(prompt) - # gets input as a string
9 help() - # offers interactive help

```

IMPORT PACKAGES

```

1 import module
2 import module as name
3 from module, submodule import function

```

Datatypes

```

1 int(), float(), str() - # casts an object to a desired
                          datatype

```

```

1 int: 1 - # Integer
2 float: 1.674 - # Float
3 bool: True/False - # Boolean
4 str: „Hello Statworx“ - # String

```

Structures

```

1 list: [1, 1.674, True, „Hello“] - # allows multiple items in
                                   one object
2 tuple: (1, 1.674, True, „Hello“) - # cannot be changed, only
                                   limited use
3 dict: {„A“: 1, „B“: 1.674, „C“: - # Dictionary
         True, „D“: „Hello Statworx“}

```

```

1 type(object) - # returns class type of object

```

Find help online: www.statworx.com/de/blog

Operations

NUMBERS

Basic Arithmetics: $+-*/$
 Exponentiation: $**$
 Floor Division: $//$
 Moduls: $%$

Logical And: **and**
 Logical Or: **or**
 Logical Not: **not**

Assignment Operators: Combine one of the above operators with: $=$, f.e. $x += 3$

Equal to: $==$
 Less than: $<$
 Less than or equal to: $<=$

Not Equal to: $!=$
 Greater than: $>$
 Greater than or equal to: $>=$

TEXT

```

1 substr in str - # checks if substring is in string
2 substr not in str - # checks if substring is not in string
3 str + str - # joins two strings together
4 \ - # to insert characters that are illegal in a string,
   followed by the illegal character
4 f"Hello {object}" - # embed expressions inside a string
5 len(str) - # get the length of a string (number of characters)
6 .strip() - # removes any whitespace at beginning or end of string
7 .upper() - # returns string in upper case
8 .lower() - # returns string in lower case
9 .replace(old, new) - # replaces a string with another string
10 .split(separator) - # splits the str at specified separator, returns list
11 .count(str) - # returns the number of times a specified value occurs
   in a string
12 .find(str) - # searches the string for a specified value and returns
   the position of where it was found

```

LIST

```

1 object in list - # checks if object exists in list
2 object not in list - # checks if object does not exists in list
3 list + list - # joins two lists together
4 len(list) - # determines how many items a list has
5 .append(object) - # adds object to the end of list
6 .insert(index, object) - # adds object at a specified index
7 .remove(object) - # removes the specified object
8 .pop(index) - # removes specified index (or last item if index is
   not specified)
9 .copy() - # returns a copy of the list
10 .count(object) - # returns the number of items with specified value
11 .index(object) - # returns the position at the first occurrence of the
   specified value
12 .sort() - # sorts the list ascending by default
13 .reverse() - # reverses the sorting order of items

```



PYTHON BASICS CHEAT SHEET

STATWORX ACADEMY FURTHER PYTHON TRAININGS

INDEXING

STRING, LIST OR TUPLE

```

1 a[0] - # get the first element
2 a[-1] - # get the last element
3 a[3:6] - # elements from position 3 to position 6 (not included)
4 a[:6] - # all elements up to position 6 (not included)
5 a[6:] - # all elements from position 6 on (included)
6 a[-4:-1] - # elements from the 4th from last to the last position (not included)
7 a[-4:] - # elements from the 4th from last to the last position (included)
8 a[:-4] - # all elements up to the fourth last (not included)

```

CONTROL STRUCTURES

CONDITION, LOOPS, LOOP CONTROL

IF-ELSE-CONDITION

```

1 if condition:
2     statement (s)
3 elif condition_2:
4     statement (s)
5 else condition_3:
6     statement (s)

```

WHILE-LOOP

```

1 while condition:
2     statement (s)

```

FOR-LOOP

```

1 for var in iterable:
2     statement (s)

```

LOOP CONTROL STATEMENTS

```

1 break - # terminates the current loop and resumes execution at the next
          statement
2 continue - # rejects all the remaining statements in current iteration of the
              loop and moves the control back to the top of the loop
3 pass - # is used when a statement is required syntactically but no command
          or code should be executed

```

OOP

FUNCTIONS, METHODS, ATTRIBUTES

FUNCTIONS

```

1 function(arg1, arg2, ...) - # called by its name and gets input
                             by its arguments

```

METHODS

```

1 object.method(arg1, arg2, ...) - # functions that belong to an object
                                   and called through the dot syntax

```

ATTRIBUTES

```

1 object.attribute - # returns the value of the attribute;
                    attributes are properties of an
                    object

```

STATWORX ACADEMY FURTHER PYTHON TRAININGS



Python Basics



**BEGINNER
LEVEL**



Machine Learning



Data Manipulation



Programming



Data Visualization



Statistics

ADVANCED LEVEL



Advanced Data Manipulation

Using panda



Time Series Analysis

Time series knowledge



Interactive Data Visualization

With the interactive library plotly



Advanced ML

Advanced algorithms



Deep Learning

Using TensorFlow API Keras



Advanced Programming

OOP in python

EXPERT LEVEL



Distributed Systems

In Spark



Advanced Statistics

e.g. multilevel models



Dashboarding

Using Dash or Streamlit



DevOps

Versioning and debugging in python



Advanced Deep Learning

Topics regarding neural networks



NLP

Introduction to sequential data and embeddings for language



ML Ops

Application of DevOps in ML context