**1. Meet Python**

* EMPFOHLENE KURSE & ONLINE ENVIRONMENTS  
  Nutze zum Einstieg in Python (einen der) folgende E-Learning-Kurse:
  + [**Scientific Computing with Python Certification | freeCodeCamp.org**](https://www.freecodecamp.org/learn/scientific-computing-with-python/#python-for-everybody)
  + [**Python Playground | online-python**](https://www.online-python.com/)
  + [**Python Basic Tutorial | w3schools**](https://www.w3schools.com/python/)
  + [**Learn Python from basic to advance | learnpython.org**](https://www.learnpython.org/)

THEORETISCHE FRAGESTELLUNGEN

Setze dich zu Beginn mit folgenden Fragestellungen auseinander um einen guten Überblick zu bekommen.

**Basic Fragen:**

* + - Was bedeuten die folgende Begriffe: Programmiersprache, Scripting-Sprache, Kompilierte Sprachen und Interpretierte Sprachen?
    - Was ist Python?
    - Wieso ist Python so populär?
    - Was sind Typen in Python und was ist eine Typed Sprache?
    - Was ist eine Variable?
    - Wie organisiert man Code im Python?
    - Was sind Funktionsparameter und Return-Werts?
    - Welche Kontroll-Strukturen gibt es im Python?
    - Wann verwendet man welche Kontroll-Strukturen?
    - Welche Thumb-Rules gibt es bei der Verwendung vom Kontroll-Strukturen?
    - Was sind die gängigsten Daten Strukturen in Python? Was sind ihre Merkmale?
    - Was sind Python Modules? Welche Module nutzt man für Syscalls?
    - Wie funktioniert File I/O im Python?
    - Was ist eine „Regular Expression - Regex“, wann verwendet man die ?
    - Was ist und wozu dient „Error und Exception handling“?

**Advanced Fragen:**

* + - Was ist objektorientiertes Programmieren?
    - Wozu braucht man einen Konstruktor?
    - Was sind die Attribute?
    - Was sind die Methoden?
    - Was ist ein Iterator?

SELBSTKONTROLLLE

Führe folgende Übungsbeispiele zur Selbstkontrolle des bisher Gelernten durch.  
Die Übungen sind **freiwillig** und du musst sie **nicht** im Confluence abgeben/dokumentieren :

**Basic:**

* + Exercise 1: Calculate the multiplication and sum of two numbers

Given two integer numbers return their product only if the product is equal to or lower than 1000, else return their sum.

* + Exercise 2: Print the sum of the current number and the previous number

Write a program to iterate the first 10 numbers and in each iteration, print the sum of the current and previous number.

* + Exercise 3: Remove first n characters from a string

Write a program to remove characters from a string starting from zero up to n and return a new string.

* + Exercise 4: Display numbers divisible by 5 from a list

Iterate the given list of numbers and print only those numbers which are divisible by 5.

* + Exercise 5: Return the count of a given substring from a string

Write a program to find how many times substring x appears in the given string y.

* + Exercise 6: Check Palindrome

Write a program to check if the given string is a palindrome.

* + Exercise 7: Calculate income tax for the given income by adhering to the below rules

| Taxable income | Rate(in %) |
| --- | --- |
| EUR 0 - 10 999.99 | 0 |
| EUR 11 000 - 22 999.99 | 15 |
| EUR 23 000 - 49 999.99 | 25 |
| EUR 50 000 - 99 999.99 | 35 |
| EUR 100 000 - | 45 |

* + Exercise 8: Store list of students

Write a program to ask for a list of Students with their Names, Age and Grade and store it into the file names students.txt.

* + Exercise 9: Edit students.txt

Write a program which gives option to editing the existing list of students: Adding new Student, Deleting Existing student or Changing the Grade of the existing student. Store the changes into students.txt.

**Advanced:**

* + Exercise 10: Make a Program with 3 Types of Objects: Superheroes, Marvel Heroes, DC Heroe

Write a program in which the user will be offered to create different super heros, choosing they type (Marvel, DC), Name, Attack Strength and Defense Strength (as integer Number). The heroes can fight each other. In the fight the one wins whose Defense Strength is stronger after the attack. (Defense will be reduced by the number of the Attack Strength of the other.

VERPFLICHTENDES PRAXISBEISPIEL

Im folgenden Beispiel wirst du deine bereits gelernten Python Fähigkeiten einsetzen, um ein Script zu erstellen und den Syslog deines Ubuntu Servers auszulesen. Dokumentiere dabei dein Vorgehen in [**Confluence**](https://confluence.infonova.at/pages/viewpage.action?pageId=184647891).

**Szenario:**

In unserer Firma haben wir sehr viele Ubuntu Servers. Sehr oft müssen unsere Mitarbeiter in die Syslogs schauen um Probleme zu identifizieren. Um diese Aufgabe für alle zu erleichtern, hat sich unser Management überlegt die Logs täglich leserlich zu machen und nach dem Datum in eine eigene Datei im Homeverzeichnis zu speichern. Da du gerade frischer Python-Profi geworden bist, sollst du diese Aufgabe übernehmen. Beachte hierbei, dass dein Code so „leserlich“ wie möglich geschrieben ist. Deine Code-Blöcke und einzelne Werte sollten sich also nicht wiederholen, damit deine Kollegen:innen diesen auch gut nachvollziehen und benutzen können.

Deine Aufgaben:

* + Stelle sicher dass auf deiner Machine Python 3 und pip3 installiert sind. Installiere dir ein [**IDE**](https://www.codecademy.com/article/what-is-an-ide)deiner Wahl.

Vorschläge für IDE:

[**Visual Studio Code**](https://code.visualstudio.com/)**,**

[**PyCharm**](https://www.jetbrains.com/pycharm/)**,**

[**Thonny**](https://thonny.org/)

* + Erstelle ein Python Script, das Folgendes tut:  
    - Syslog in dein Homeverzeichnis kopieren.
    - Kopierten Syslog einlesen.
    - Bei jeder Log-Zeile nach dem Timestamp und vor der Error-Message alles mit einem „-“ ersetzen.
      * Beispiel:

Before -> Mar 03 2023 21:20:10 ubuntu01 %ubuntu01-5-1119080: ERROR: …

After -> Mar 03 2023 21:20:10 - ERROR: …

* + - Die bearbeitete Zeile in eine neue Datei mit dem heutigem Datum im Namen ins Homeverzeichnis schreiben.
    - Du kannst gerne auch andere Features in das Skript einbauen, wenn du sie für sinnvoll hältst :)
  + Das Script soll automatisch um 00:00 Uhr von Mo-Fr. ausgeführt werden (du sollst alles über Python einrichten können!)

ANMERKUNGEN  
👉 Deine Dokumentation soll eine prägnante:

* + - * Beschreibung deiner Vorgehensweise (welche Probleme hast du definiert? welches Problem hast du auf welche Art und Weise gelöst?)
      * Begründung deiner wichtigsten Entscheidungen (Welche Alternativen hattest du und wieso hast du dich für genau die entschieden) sein
      * How-To-Anleitung (für deine Kolleg:innen wenn sie deine Scripts verwenden wollen) darstellen.

Hilfreiche links zum Doku-Schreiben:

[**Beginners Guide to Docs**](https://www.writethedocs.org/guide/writing/beginners-guide-to-docs/)

[**How to write good documentation**](https://www.freecodecamp.org/news/how-to-write-good-documentation/)

[**Create engaging confluence pages: 4 easy steps**](https://www.atlassian.com/blog/confluence/create-engaging-confluence-pages-4-easy-steps)