

Rechnernetze POP3 in Python

Grundlegende Syntax von Python

Python ist eine Skriptsprache, die vor der Ausführung nicht zuerst übersetzt werden muss. Blöcke werden in ihr durch Einrückungen gekennzeichnet. Methodenköpfe, Schleifen und Bedingte Anweisungen werden mit einem Doppelpunkt abgeschlossen.

Java:

```
if( a > 5 ) {
    while( a > 0 ) {
        System.out.println(a);
        a += 1;
    }
}

public void hallo( String pName ) {
    System.out
        .printf("Hallo %s", pName);
}
```

Python:

```
if a > 5:
    while a > 0:
        print(a)
        a += 1

def hallo( pName ):
    print('Hallo {}'.format(pName))
```

Funktionen werden in Python mit dem `def` Schlüsselwort erstellt:

```
def meine_methode( pEingabe ):
```

Einige wichtige Python-Befehle

<code>print('Hallo Welt')</code>	Gibt „Hallo Welt“ auf der Konsole aus.
<code>'string'.startswith('prefix')</code>	prüft, ob der String „string“ mit „prefix“ beginnt.
<code>'Hallo {}'.format('Welt')</code>	Ergibt den String „Hallo Welt“.
<code>'{} ist {}'.format(6, 'sechs')</code>	Ergibt den String „6 ist sechs“.
<code>'Hallo Welt'[2:-2]</code>	Ergibt den String „llo We“. Die erste oder zweite Zahl kann auch weggelassen werden oder jeweils positiv oder negativ sein.

Aufgabe 1: Kopiere die Datei `connection.py` und die Skript-Vorlage `pop3-client.py` aus dem Tauschordner in einen Ordner deiner Wahl. Starte dann das Programm `MU-Editor` und öffne das Skript `pop3-client.py`.

Analysiere die gegebene Programmvorlage und vergleiche sie mit Deinem Wissen über Java-Programme.

Aufgabe 2: Kopiert und startet den `BabyPOP3` Server. (Denkt daran erst `babypop3_config.reg` auszuführen.) Führt dann das Python-Skript im `MU-Editor` aus.

Aufgabe 3: Das Programm benutzt die Klasse `Connection`, um eine TCP-Verbindung herzustellen. Die Klasse bekommt im Konstruktor die IP und den Port des Server übergeben und verbindet dann automatisch. Sie besitzt drei Methoden:

<code>receive()</code>	Wartet auf eine Zeile Text und gibt diese nach Erhalt als String zurück.
<code>send(str)</code>	Sendet einen String an den Server.
<code>close()</code>	Schließt die Verbindung zum Server.

Aufgabe 4: Implementiert den vollständigen Abruf einer Mail vom Server.

Aufgabe 5: Implementiert die Funktion `deleteNextMail`.