

Aufgabe 1 Analyse des Quelltextes

- a. Kennzeichne alle dir schon aus anderen Programmiersprachen bekannten Schlüsselworte im Quelltext und erkläre ihre Bedeutung.
- b. Prozeduren oder Funktionen markiert man in python genauso wie in anderen Programmiersprachen durch eine Parameterliste in runden Klammern. Notiere alle Funktionsnamen. Hast du eine Idee, wofür sie stehen?
- c. Finde bei der Untersuchung der restlichen Bezeichner heraus, welche für Zahlen, Zeichen oder Zeichenketten stehen.

Aufgabe 2 Verwendung des python-GUI IDLE

- a. Lege einen Ordner `Kryptographie` und darin ein neues Textdokument „`krypto.py`“ an.
- b. Öffne das Dokument durch einen Rechtsklick und die Auswahl von „Edit with IDLE“.
- c. Gib das Programm der nächsten Seite im Editor ein. Beachte dabei die Einrückungen. Speichere die Datei regelmäßig.

Aufgabe 3 Prüfen, Laden und Testen des Programms in IDLE

- a. Teste das Programm. Starte es dazu mit F5 und gib dann im Interpreter den Befehl `print caesar(„testtext“, 3)` ein.

Notiere die Ausgabe und gehe zum Lehrer um die Korrektheit zu überprüfen.

- b. Berichtige das Programm und teste es erneut. Achtung: Speichern und F5 drücken.

Aufgabe 4 Anwenden der Funktion

- a. Verschlüssele folgende Wörter mit der python-Funktion:

Klartext	hallo	informatik	kolibri	affe	enirsta
Schlüssel	5	3	13	11	33
Geheimtext					

Das Wort `enirsta` gibt es zwar gar nicht, aber es enthält die sieben häufigsten deutschen Buchstaben in der richtigen Reihenfolge...

- b. Finde drei weitere Schlüssel, die anstelle von 33 zum selben Ergebnis führen. Formuliere die Beziehung zwischen gleichwertigen Schlüsseln mathematisch.
- c. Ermittle für die Texte `hallo` und `kolibri` einen weiteren Schlüssel, mit dem die Verschlüsselung wieder rückgängig gemacht werden kann. Finde eine allgemeine Regel, diesen Umkehrschlüssel zu berechnen.

Hinweis Negative Zahlen sind als Schlüssel nicht zugelassen.

CAESAR-Chiffre in python

Inhalt der Datei `krypto.py`

```
# *****
# Programm zum Verschluesseln einer Nachricht
# nach dem Caesar-Verfahren
# *****

def caesar(klartext, schluessel):
    klartext = klartext.upper()
    geheimtext = ""
    for i in range(0, len(klartext)):
        buchstabe = klartext[i]
        ascii = ord(buchstabe)
        code = ascii + schluessel
        while (code > 90):
            code = code - 26
        geheimbuchstabe = chr(code)
        geheimtext = geheimtext + geheimbuchstabe
    return geheimtext
```

Aufgabe 5 Erweitern der Funktion

Entwirf und teste eine Funktion `decaesar(geheimtext, schluessel)`, welche den Geheimtext wieder in Klartext (Kleinbuchstaben!) verwandelt. Die Funktion soll aus nicht mehr als vier Programmzeilen bestehen.

Aufgabe 6 Struktogramme

- Kommentiere die einzelnen Quelltextzeilen.
- Kennzeichne die strukturierten Anweisungen (Schleifen, Verzweigungen, Sequenzen). Wodurch wird der Anweisungsteil innerhalb der strukturierten Anweisungen gekennzeichnet? (Zur Erinnerung: In C haben wir dazu die Klammern `{...}` verwendet.)
- Überführe das Programm in ein Struktogramm. Notiere unklare bzw. unbekannte Anweisungen bzw. Anweisungsteile.

Tafelbild zu **Aufgabe 1:**

Schlüsselworte	Funktionen	Variablennamen	Optional
<ul style="list-style-type: none"> def ... : --> while (...): --> for ... : --> return ... 	<ul style="list-style-type: none"> caesar() range() len() ord() chr() upper() 	<ul style="list-style-type: none"> Zeichenketten <ul style="list-style-type: none"> klartext geheimtext Zeichen <ul style="list-style-type: none"> klartext[...] buchstabe geheimbuchstabe Zahlen <ul style="list-style-type: none"> schluessel ascii code 	<p>ord() chr()</p>

Vorbereitung zu **Aufgabe 2:**

Am Lehrercomputer vorzeigen...

- Erstellung und Namensänderung des Textdokuments
- Dokument öffnen, Hallo-Welt-Programm schreiben und speichern des Dokuments
- Ausführen im Interpreter
- Verändern und ausführen ohne F5 (→ keine Veränderung im Interpreter)
- Ausführen mit F5

```
def ausgabe():
    print ("Hallo Welt")
    for i in range(0,5):
        print(i)
    return "fertig"
```

Lösung zu **Aufgabe 3:** print caesar("testtext",3) → **WHVWWHAW**

Lösung zu **Aufgabe 4:**

Klartext	hallo	informatik	kolibri	affe	enirsta
Schlüssel	5	3	13	11	33
Geheimtext	MFQQT	LQIRUPDWLN	XBYVOEV	LQQP	LUPYZAH

Tafelbild zu Aufgabe 4:**b.** $33 - 26 = 7$

Schlüssel sind gleichwertig, wenn ihre Differenz ein Vielfaches von 26 ist.

(Alternativ: gleicher Rest bei Division durch 26)

c. Hallo: $26 - 5 = 21$ UmkehrschlüsselKolibri: $26 - 13 = 13$ Umkehrschlüssel

Der Umkehrschlüssel ist die Differenz aus 26 und dem Schlüssel bzw. des gleichwertigen Partners, der kleiner 26 ist.

Lösung zu Aufgabe 5:

```
def decaesar(geheimtext, schluessel):  
    while (schluessel > 26):          #nur noetig fuer schluessel > 26  
        schluessel = schluessel - 26 #in Verbindung mit hinteren Buchstaben im  
                                     Alphabet  
    return caesar(geheimtext, 26 - schluessel).lower()
```