Empfohlene Unterrichtsinhalte als Vorbereitung für dieses Thema: 02-01/2/3/4/5 (Variablen, Strings, Listen, Integers und Floats, Bool)

Einführung in die computerlinguistische Programmierung mit Python

03-02: Bedingungen 👀: `if`, `elif`, `else`

Wir haben bisher verschiedene Datentypen und Befehle kennengelernt mit denen wir einfache Programme schreiben können, in denen alle Befehle Zeile für Zeile abgearbeitet werden. Bedingungen sind ein neues Werkzeug mit dem wir komplexere Programme gestalten können. Mit Hilfe von Wahrheitswerten und Aussagen, die wir in der letzten Woche kennengelernt haben, können wir Bedingungen formulieren, die erfüllt sein müssen, damit bestimmte Teile eines Programms ausgeführt werden.

Bedingte Code-Ausführung mit `if`

Stellen wir uns ein Programm vor, das Wörter verarbeitet und eine bestimmte Ausgabe zurück gibt, falls ein Wort mehr als 10 Zeichen enthält. Diese Programmfunktionalität können wir in folgende Teilaufgaben aufteilen:

- Es muss ein Eingabewort bereitgestellt werden
- Dann muss die Länge des Eingabeworts berechnet werden
- Danach müssen wir überprüfen, ob die Wortlänge größer als 10 ist.
- Falls dies der Fall ist, wird erfolgt eine Ausgabe.

Um die letzten beiden Punkte in Python umzusetzen benötigen wir eine erste einfache Bedingung, wie die im folgenden Code-Block.

In [1]:
```
eingabewort = "Blumenkasten"
wortlaenge = len(eingabewort)

#Hier wird's nun wirklich spannend; es folgtg eine Bedingung:
if wortlaenge > 10:  # Dieser Teil formuliert die Bedingung die überprüft w
    print("Das Wort hat mehr als 10 Zeichen.")
```

Der spannende Part im obigen Code sind die beiden Zeilen, die auf die Initialisierung der Zähler folgen:

Zeile 9 beginnt mit Signalwort `if`, welches in Python markiert, dass nun eine Bedingung beginnt. Man nennt die Zeile, die mit dem `if` beginnt auch den **Kopf/die Kopfzeile des `if` - Blocks**. Gleichzeitig können wir das `if` auch als ein konditionales *wenn* interpretieren, wie es aus Konditionalsätzen kennen.

Dem `if` folgt eine zu überprüfende Bedingung in Form einer Aussage **oder** eines Ausdrucks der auf einen Bool-Wert abgebildet werden kann. Falls diese Aussage auf den Bool-Wert `True` abgebildet werden kann, also falls die Aussage wahr bzw. “truthy” ist, dann wird der Code ausgeführt, der eingerückt unterhalb der Kopfzeile steht. Den eingerückten Code

bezeichnet man als den **Körper des `if`-Blocks**.

In [1]:
```
n = 3

aussage_1 = n < 5
aussage_2 = n > 5

print("Die Aussagen werden durch print() automatisch ausgewertet:")
print(aussage_1)
print(aussage_2)

# print("\n---\n")

# if aussage_1:
#     print("Hurra!")

# if aussage_2:
#     print("Hurra?")

# if "Hallo Welt":
#   print("Hä?")
```

Die Bedingungen in einem `if`-Block können einfach sein ...

In [1]:
```
if 1:
    print("Passiert hier was? Warum (nicht)?")

print("\n---\n")

if 0:
    print("Passiert hier was? Warum (nicht)?")
```

...oder auch komplex.

In [1]:
```
w = "Blumenkasten"

if not w.isupper and len(w) > 10 or len(w) < 15:
    print("Tolles Wort")
```

Alternativbedingungen mit `elif`

Wir können in einem `if`-Block auch weitere Bedingungen prüfen und alternativen Code angeben, der abgearbeitet wird falls die erste Bedingung in Kopf des Blocks nicht erfüllt wird. Diese Alternativbedingungen können mit `elif` eigeführt werden.

In [1]:
```
w = "Hui"
wortlaenge = len(w)

if wortlaenge > 10:
    print("Oh, langes Wort!")
# elif wortlaenge <= 5:
#     print("Oh, ein Wort")
elif wortlaenge >= 7:
    print("Oh, ein mittellanges Wort!")
elif wortlaenge >= 3 and wortlaenge < 7:
    print("Oh, ein kurzes Wort!")
```

Bei der Formulierung der Bedingungen und deren Anordnung in einem komplexeren `if`-Block müssen wir darauf achten, dass nicht eine leichter zu erfüllende Bedingung zu früh im

Block auftaucht. **Sobald eine der Bedingungen erfüllt ist, werden alle folgenden Bedingungen ignoriert. Es wird nur der jeweilige zur Bedinngung gehörige Code ausgeführt.**

Wenn alle Bedingungen gerissen werden: `else`

Mit Hilfe von `else` können wir für den Fall der Nichterfüllung aller vorherigen Bedingungen, stattdessen einen Satz von alternativen Befehlen ausführen.

In [1]:
```
w = "Hui"
wortlaenge = len(w)

if wortlaenge > 10:
    print("Oh, langes Wort!")
elif wortlaenge >= 5:
    print("Oh, ein mittellanges Wort!")
else:
    print("Hm, ein Wort dass mich nicht interessiert.")
```

Im Unterschied zum letzten Code-Block wird hier nun in jedem Fall ein String ausgegeben.

In short: Die Syntax eines `if`-Blocks

Mit `if`, `elif`, `else` können wir prüfen, ob die angegebenen Bedingungen erfüllt sind. Ein `if`-Block hat die folgende Struktur:

```
if <bedingung>:
    <befehl1>
    <befehl2 usw.>  # optional
elif <bedingung2>:  # optional
    <befehl3>
    <befehl4 usw.>  # optional
else:               # optional
    <befehl5>
    <befehl6 usw.>  # optional
```

Eine Bedingung kann alles sein, was sich zu einem Wahrheitswert umwandeln lässt - also praktisch alles. Wenn die Bedingung zum Wahrheitswert `False` führt, ist das `if` nicht zutreffend.

Wenn wir nur den `if`-Block verwenden und die beiden anderen weglassen, wird bei Nichterfüllung der Bedingung einfach der ganze Block übersprungen.

Wenn wir `if` in Kombination mit `elif` verwenden, wird bei Nichterfüllung der ersten Bedingung als nächstes die zweite Bedingung geprüft. Wir können beliebig viele `elif`-Blöcke aneinanderreihen. **Achtung:** Sobald ein `if`- oder `elif`-Block ausgeführt wurde, werden die Bedingungen der verbleibenden `elif`-Blöcke nicht mehr geprüft und die Befehle dort übersprungen.

Wenn wir `else` verwenden, werden bei Nichterfüllung der Bedingungen in den Blöcken davor stattdessen die Befehle im `else`-Block ausgeführt.

Zusammenfassung

- Bedingungen ermöglichen uns das selektive Ausführen von Programm-Code unter bestimmten Bedingungen.
- `if`, `elif`, and `else` erlauben uns verschiedene Bedingungen zu überprüfen und abhängig davon verschiedene Code-Bestandteile auszuführen.
- Innerhalb eines `if`-Blocks können beliebig viele `elif`-Bedingungen auftauchen.

Weitere Themen dieser Woche

- Indexing und Slicing 🍰
- Schleifen 🔃
- Zufallszahlen 🔢