Steuerung des Programmablaufs durch Kontrollstrukturen

Verzweigungen

- Zur Verzweigung des Programmablaufs stellen Programmiersprachen Mittel zur Verfügung.
- Wir kennen bereits das ***if***


```
zahl = int(input('Ganze Zahl eingeben:'))

if zahl>0:                          erste Bedingung
    print ('Zahl ist positiv')
elif zahl<0:                        alternative Bedingung
    print ('Zahl ist negativ')
else:                               alle sonstigen Fälle
    print ('Zahl ist null')
```

Verzweigungen

- Auf die Verzweigung durch ***bedingte Ausdrücke*** gehe ich nicht ein.

```
zahl = int(input('Ganze Zahl eingeben:'))

text = ('Zahl ist negativ' if zahl<0 else 'Zahl ist nicht negativ')
```

Wiederholungen

- Zur wiederholten Ausführung von Programm-blöcken stellen Programmiersprachen Mittel zur Verfügung.
- Python
  - for – Schleife
  - while – Schleife
  - Rekursion

Wiederholungen

for – Schleife

- Die for – Schleife setzt man ein, wenn man vorher weiß,
  - ***wie oft***

  oder
  - ***für welche***

  sie ausgeführt werden soll.
- *[Sie lässt sich mit break beenden und man kann mit continue den Durchlauf verlassen.]*

Wiederholungen

- for – Schleife als Zählschleife

```
for i in range(5):
    print i
```

- for als "for each"

```
for wort in text:
    if wort=='Hund':
        print ('gefunden')
```

text muss ein Sammlungsobjekt sein

Wiederholungen

while – Schleife

- Die Wiederholung wird so lange ausgeführt, wie eine Bedingung gilt.

```
while zahl>0:
    print (zahl)
    zahl = zahl - 1
```

Wiederholungen

Rekursion

- Durch Rekursion lassen sich Wiederholungen umsetzen.
- Sie sind bei Python nicht „das Werkzeug” wie bei funktionalen Sprachen.

```
def summeBis(zahl):
    if zahl<=0:          ## Abbruchbedingung
        return 0
    else:
        return zahl + summeBis(zahl-1)    ## rekursiver Aufruf
```