Kontrollstrukturen I

Python3 – Kontrollstrukturen I

Kontrollstrukturen erlauben Abweichung

von Sequentialität

Zwei Arten:

- Verzweigungen
- Schleifen

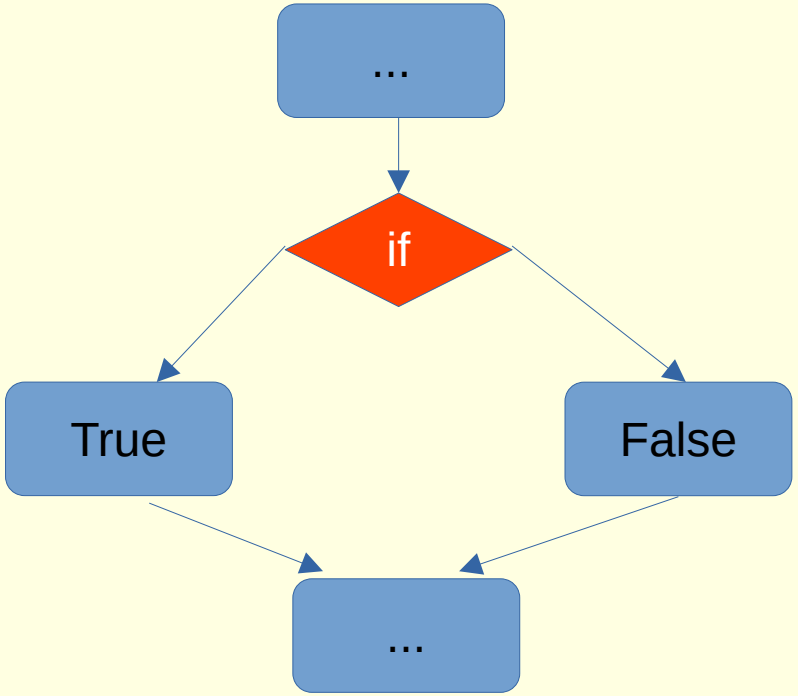
Python3 – Kontrollstrukturen I

Kontrollstrukturen erlauben Abweichung

von Sequentialität

Zwei Arten:

- Verzweigungen
- Schleifen



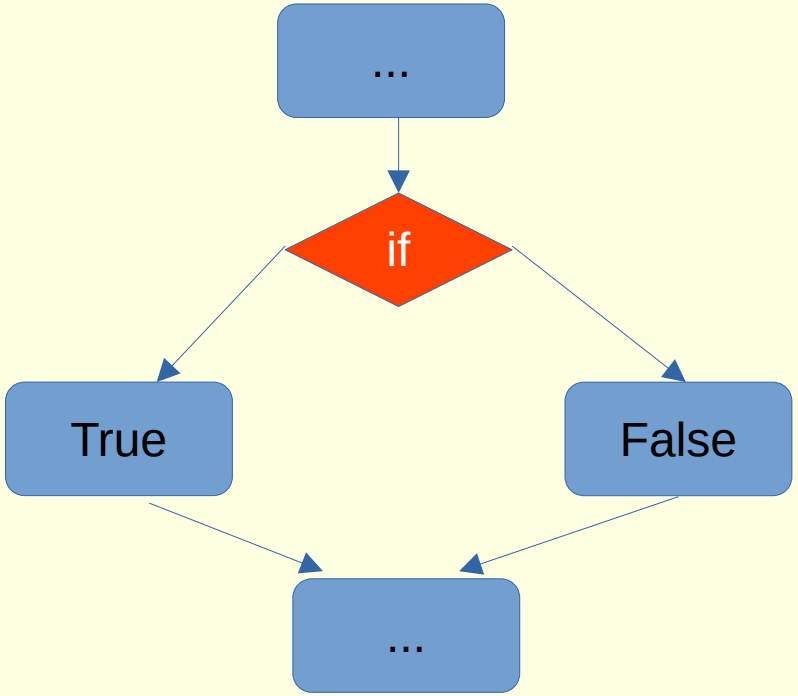
Python3 – Kontrollstrukturen I

if – Anweisungen

- prüfen eine Bedingung
- führen dann einen

  von zwei möglichen

  Blöcken aus



Python3 – Kontrollstrukturen I

if - Anweisung

- Bedingung boolesch
- Anweisungen eingerückt

```
if Bedingung:
    Anweisung
    …
    Anweisung
else:
    Anweisung
    …
    Anweisung
```

Python3 – Kontrollstrukturen I

if – Anweisung

- kein else
- beide `print` ausgeführt

```
1 x = 10
2 if x > 5:
3     print("grösser")
4 print("fertig")
```

Python3 – Kontrollstrukturen I

if – Anweisung

- else wenn x>5 False
- nur zwei `print` ausgeführt

```
1 x = 10
2 if x > 5:
3     print("grösser")
4 else:
5     print("nicht grössser")
6 print("fertig")
```

Python3 – Kontrollstrukturen I

if – Anweisung

- 3 Fälle
- Problem?

```
1 x = 10
2 if x > 5:
3     print("grösser")
4 if x == 5:
5     print("gleich")
6 else:
7     print("kleiner")
8 print("fertig")
```

Python3 – Kontrollstrukturen I

if – Anweisung

- 3 Fälle
- unübersichtlich

```
x = 10
if x > 5:
    print("grösser")
if x == 5:
    print("gleich")
else:
    if x < 5:
        print("kleiner")
print("fertig")
```

Python3 – Kontrollstrukturen I

if – Anweisung mit elif

- beliebig viele
- else wenn keine

  Bedingung zutrifft

```
1 x = 10
2 if x > 5:
3   print("grösser")
4 elif x == 5:
5   print("gleich")
6 else:
7   print("kleiner")
8 print("fertig")
```

Python3 – Kontrollstrukturen I

PCAP-Fragen

- Was ist Ausgabe folgenden Programmes?

```
a) 42

b) 7
   x

c) 7
   x
   2

d) x
   2
```

```
1  x = 8
2  if x > 5:
3      if x < 5:
4          if x > 7:
5              print("7")
6      else:
7          print("x")
8  else:
9      print("4", end="")
10 print("2")
```

Python3 – Kontrollstrukturen I

Aufgabe:

Schreiben Sie ein Programm das

- eine Zahl von der Tastatur einliest
- je nach dem Wert der Zahl "positiv", "negativ" oder "Null" ausgibt

Python3 – Kontrollstrukturen I

Bedingte Ausdrücke

- vier Zeilen
- nur ein Anweisung ausgeführt

```
1 if x == 1:
2     var = 20
3 else:
4     var = 30
```

Python3 – Kontrollstrukturen I

Bedingte Ausdrücke

- vier Zeilen
- nur ein Anweisung ausgeführt
- Abkürzung:

```
1 var = (20 if x == 1 else 30)
```

```
1 if x == 1:
2     var = 20
3 else:
4     var = 30
```

Python3 – Kontrollstrukturen I

Bedingte Ausdrücke

- keine Doppelpunkte
- keine Einrückungen
- kurz, aber schwer lesbar

```
1 var = (20 if x == 1 else 30)
```

Python3 – Kontrollstrukturen I

Bedingte Ausdrücke

- nur wenn in allen Zweigen Zuweisung an selbe Variable

```
1 var = (20 if x == 1 else 30)
```

Python3 – Kontrollstrukturen I

Bedingte Ausdrücke

- Bedingung kann komplex sein
- Werte müssen nicht elementar sein

```
1 var = (x * 2 if (x > 10 or y < 5) else y * 2)
```

Python3 – Kontrollstrukturen I

Was gibt folgendes Programm aus?

```
1 x = 8
2 y = 3
3 var = ( x - 2 * 3 if ( x + y > 10 and y < 5 ) else y - 3 * 2 )
4 print( var )
```

a) 18

b) 0

c) 2

d) -3

Python3 – Kontrollstrukturen I

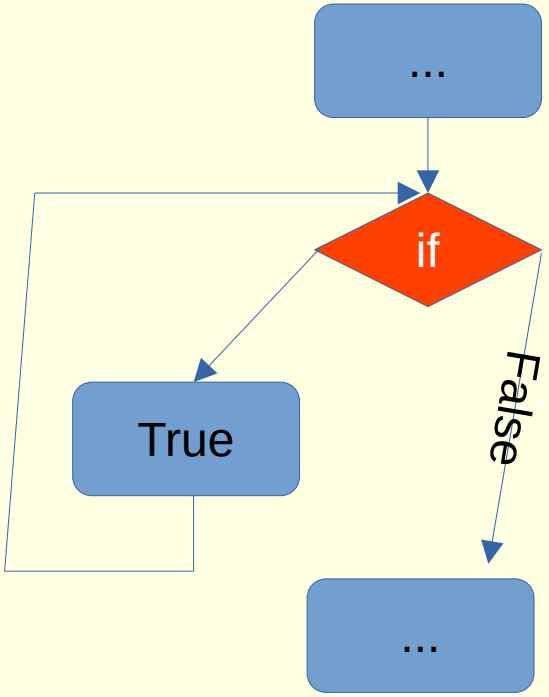
Aufgabe:

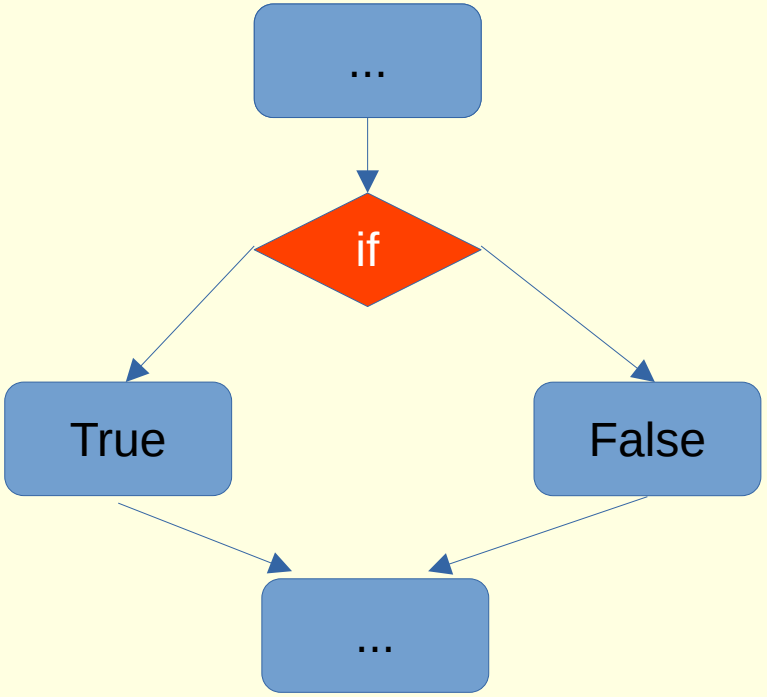
Schreiben Sie ein Programm das

- eine Zahl von der Tastatur einliest
- je nach dem Wert der Zahl einer Variable den String "nichtnegativ" oder "negativ" zuweist
- am Ende den String ausgibt
- verwenden Sie für die Zuweisung einen bedingten Ausdruck

Python3 – Kontrollstrukturen I

Zweite Kontrollstruktur: **Schleifen**





Python3 – Kontrollstrukturen I

Zweite Kontrollstruktur: **Schleifen**

- Bedingung wird wiederholt geprüft
- Anweisung wird wiederholt ausgeführt

Python3 – Kontrollstrukturen I

while – Schleife

```
while Bedingung:
    Anweisung
    …
    Anweisung
```

- Bedingung boolesch
- Anweisung ausgeführt

  bis Bedingung False

Python3 – Kontrollstrukturen I

while – Schleife

- Bedingung boolesch
- Anweisung ausgeführt

  bis Bedingung False

```
1 bedg = True
2 x = 0
3 while bedg:
4     x += 2
5     if x > 10:
6         bedg = False
7 print(bedg)
```

Python3 – Kontrollstrukturen I

while – Schleife

```
1 bedg = True
2 x = 0
3 while bedg:
4     x += 2
5     if x > 10:
6         bedg = False
7 print(bedg)
```

- Bedingung boolesch
- Anweisung ausgeführt

  bis Bedingung False
- Bedingung muss irgendwann False werden

Python3 – Kontrollstrukturen I

while – Schleife

```
1  bedg = True
2  x = 0
3
4  while x != 9 :
5      x += 2
6      print(x)
7
8  print(bedg)
```

- Bedingung boolesch
- Anweisung ausgeführt

  bis Bedingung False
- Bedingung muss irgendwann False werden

Python3 – Kontrollstrukturen I

while – Schleife

```
1  bedg = True
2  x = 0
3
4  while bedg:
5      x += 2
6      if x > 9 :
7          bedg = False
8      print(x)
9
10 print(bedg)
```

```
1  bedg = True
2  x = 0
3
4  while bedg:
5      x += 2
6      if x > 9 :
7          break
8      print(x)
9
10 print(bedg)
```

break bricht Schleife sofort ab

Python3 – Kontrollstrukturen I

while – Schleife mit else

```
while Bedingung:
    Anweisung
    …
    Anweisung
else :
    Anweisung
    …
    Anweisung
```

- Anweisungsblock bei else

  ausgeführt bei "natürlichem"

  Ende der Schleife

  (Bedingung zu False ausgewertet)

Python3 – Kontrollstrukturen I

Unterscheidung ob break oder normaler Abbruch

```
1  bedg = True
2  x = 0
3  while bedg:
4      x += 2
5      if x > 10:
6          break
7  else:
8      print(x)
```

```
1  bedg = True
2  x = 0
3  while bedg:
4      x += 2
5      if x > 10:
6          bedg = False
7  else:
8      print(x)
```

x wird nur rechts ausgegeben

Python3 – Kontrollstrukturen I

Weitere Möglichkeit Fluss in Schleife zu steuern:

continue

- bricht aktuellen Durchlauf ab
- kehrt zur Bedingung zurück
- beendet Schleife nicht

```
1  bedg = True
2  x = 0
3
4  while bedg:
5      x += 2
6      if x > 9 :
7          break
8      print( x, end = " - " )
9      if x > 5 :
10         print( "" )
11         continue
12     print(x)
```

Python3 – Kontrollstrukturen I

Was gibt folgendes Programm aus?

```
1  x = 1
2  while True:
3      x *= 2
4      if x < 10:
5          x += 1
6          continue
7      if x > 21 :
8              x += 2;
9              continue
10     x += 1
11     break
12 else:
13     x += 3
14 print(x)
```

a) 18

b) 15

c) 1

d) 2

Python3 – Kontrollstrukturen I

Aufgabe

Erstellen Sie ein Programm, das

- zwei Zahlen von der Tastatur einliest
- alle Zahlen ausgibt, die zwischen den beiden eingegebenen liegen

Python3 – Kontrollstrukturen I

Aufgabe

Erstellen Sie ein Programm, das

- zwei Zahlen von der Tastatur einliest
- alle **geraden** Zahlen ausgibt, die zwischen den beiden eingegebenen liegen

Python3 – Kontrollstrukturen I

Zählschleifen

```
1 x = 0
2 while True:
3     x += 1
4     print(x, ", ", end="")
5     if x == 7:
6         break


Ausgabe:
1 , 2 , 3 , 4 , 5 , 6 , 7 ,
```

Python3 – Kontrollstrukturen I

Zählschleifen

```
1 x = 0
2 while True:
3     x += 1
4     print(x, ", ", end="")
5     if x == 7:
6         break
```

Ausgabe:
1 , 2 , 3 , 4 , 5 , 6 , 7 ,

```
1 for x in range(7):
2     print(x+1, ", ", end="")
```

Ausgabe:
1 , 2 , 3 , 4 , 5 , 6 , 7 ,

Python3 – Kontrollstrukturen I

for – Schleifen

- durchlaufen iterierbares Objekt
- range(7): Liste der Zahlen 0-6
- x nimmt alle Werte an

```
1 for x in range(7):
2     print(x+1, ", ", end="")
```

```
Ausgabe:
1 , 2 , 3 , 4 , 5 , 6 , 7 ,
```

Python3 – Kontrollstrukturen I

for – Schleifen

- range() mit zwei Argumenten:
- Startwert (dabei) und
- Endwert (nicht dabei)

```
1 for i in range( 4, 9 ) :
2     print( i , end = ", " )
```

```
Ausgabe:
4, 5, 6, 7, 8,
```

Python3 – Kontrollstrukturen I

for – Schleifen

- range() mit drei Argumenten:
- drittes: Schrittweite

```
1 for i in range( 4, 15, 3 ) :
2     print( i , end = ", " )


Ausgabe:
4, 7, 10, 13,
```

Python3 – Kontrollstrukturen I

for – Schleifen

- range() mit drei Argumenten:
- auch negative Schrittweite

```
1 for i in range( 15, 4, -3 ) :
2     print( i , end = ", " )


Ausgabe:
15, 12, 9, 6,
```

Python3 – Kontrollstrukturen I

for – Schleifen

- durchlaufen iterierbares Objekt
- zwei Schlüsselwörter in Kopf
- Variable im Körper verfügbar

```
for Variable in Iterierbares :
    Anweisung
    …
    Anweisung
```

Python3 – Kontrollstrukturen I

for – Schleifen

- continue und break wie in while-Schleifen
- mehr Varianten,

  wenn wir mehr iterierbare Objekte kennen

Python3 – Kontrollstrukturen I

Aufgabe

Erstellen Sie ein Programm, das

- zwei Zahlen von der Tastatur einliest
- mittels einer for-Schleife alle Zahlen ausgibt, die zwischen den beiden eingegebenen liegen

Python3 – Kontrollstrukturen I

Aufgabe

Erstellen Sie ein Programm, das

- zwei Zahlen von der Tastatur einliest
- mittels einer for-Schleife alle **geraden** Zahlen ausgibt, die zwischen den beiden eingegebenen liegen