

## Panta Rhei (Flusssteuerung)

### Gleichheit von Text und Zahlen

```
42 == '42'  
False  
42 == 42.0  
True  
42.0 == 0042.000  
True
```

### Boolsche Werte

```
spam = True  
spam  
True  
true
```

```
-----  
NameError                                Traceback (most recent call last)  
  
<ipython-input-20-74d9a83219ca> in <module>()  
----> 1 true
```

```
NameError: name 'true' is not defined  
  
spam = true  
spam
```

```
-----  
NameError                                Traceback (most recent call last)  
  
<ipython-input-21-8337e4d5075e> in <module>()  
----> 1 spam = true  
      2 spam
```

```
NameError: name 'true' is not defined  
  
True = 2 + 2
```

```
File "<ipython-input-22-5f87ef2f4024>", line 1
```

```
True = 2 + 2
      ^
```

SyntaxError: can't assign to keyword

**Arbeitsauftrag:** Welche Vergleichsoperatoren gibt es unter Python3?

## Boolsche Operatoren

```
True and True
```

```
True
```

```
True and False
```

```
False
```

```
False or True
```

```
True
```

```
False or False
```

```
False
```

```
not True
```

```
False
```

```
not not not not True
```

```
True
```

```
(4 < 5) and (5 < 6)
```

```
True
```

```
(4 < 5) and (9 < 6)
```

```
False
```

```
(1 == 2) or (2 == 2)
```

```
True
```

## Elemente zur Flusssteuerung

Fast alle Flusssteuerungselemente fangen mit einer **Bedingung** an, auf der ein **Codeblock (Klausel)** folgt.

## Codeblock

1. Ein Codeblock fängt mit einer Einrückung an.
2. Blöcke können weitere Blöcke beinhalten.
3. Ein Block endet, wenn die Einrückung auf null geht, oder auf die Einrückung des einschließenden Blocks zurückgeht.

Siehe auch: [http://www.python-kurs.eu/python3\\_bloেকে.php](http://www.python-kurs.eu/python3_bloেকে.php)

```
# Flusssteuerung mit Codeblöcken
name = input()
password = ''
if name == 'Marry':
    print('Hallo Marry')
    if password == 'swordfish':
        print('Access granted.')
    else:
        print('Wrong password')
```

Michl

## Flusssteuerungsanweisungen

### if-Anweisung

- Schlüsselwort if
- Einer Bedingung
- Einen Doppelpunkt
- Einen eingerückten Codeblock

### else-Anweisung

- Schlüsselwort else
- Einem Doppelpunkt
- Einen eingerückten Codeblock

### elif-Anweisung

- Schlüsselwort elif
- Einer Bedingung
- Einen Doppelpunkt
- Einen eingerückten Codeblock

## while-Schleifen

- Schlüsselwort while
- Einer Bedingung
- Einen Doppelpunkt
- Einen eingerückten Codeblock

```
spam = 0
while spam < 5:
    print('Hallo, Welt!')
    spam = spam + 1

Hallo, Welt!
Hallo, Welt!
Hallo, Welt!
Hallo, Welt!
Hallo, Welt!

name = ''
while name != 'deinen Namen':
    print('Bitte tippe deinen Namen')
    name = input()
print('Danke sehr')

Bitte tippe deinen Namen
michl
Bitte tippe deinen Namen
deinen Namen
Danke sehr
```

## break-Anweisung

Bei einer break-Anweisung wird eine while-Schleife sofort verlassen.

```
while True:
    print('Bitte tippe deinen Namen')
    name = input()
    if name == 'deinen Namen':
        break
print('Danke sehr')

Bitte tippe deinen Namen
klaus
Bitte tippe deinen Namen
deinen Namen
Danke sehr
```

## continue-Anweisung

Bei einer continue-Anweisung erfolgt ein Rücksprung zum Anfang der Schleife.

```
while True:
    print('Wer bist Du?')
    name = input()
    if name != 'Michael':
        continue
    print('Hallo Michael. Wie lautet das Passwort. Tipp: Schule')
    password = input()
    if password == "RDF":
        break
print('Zugang gewährt')

Wer bist Du?
Klaus
Wer bist Du?
Michael
Hallo Michael. Wie lautet das Passwort. Tipp: Schule
RDF
Zugang gewährt
```

## Truthy und Falsey

In Bedingungen gelten 0, 0.0 und "" (ein leerer String) als False, alle anderen Werte dagegen als True.

```
name = ''
while not name:
    print('Namen eingeben')
    name = input()
print('Wie viele Schüler sind da?')
numOfPupil = int(input())
if numOfPupil:
    print('Achte auf ausreichenden Platz')
print('Fertig')

Namen eingeben
Michael
Wie viele Schüler sind da?
5
Achte auf ausreichenden Platz
Fertig
```

## for-Schleife und range()

- Schlüsselwort for
- Einem Variablennamen

- Dem Schlüsselwort in
- Einen Aufruf von range()
- Einem Doppelpunkt
- Einen eingerückten Codeblock

Auch hier gibt es break und continue.

```
print('Mein Name ist')
for i in range(5):
    print('Foobar (' + str(i) + ')')
```

```
Mein Name ist
Foobar (0)
Foobar (1)
Foobar (2)
Foobar (3)
Foobar (4)
```

**Aufgabe** Addieren Sie mit Python alle Zahlen von 0 bis 100 (1000)

**Anfangs-, End- und Schrittwerte für range()**

```
for i in range(12,16):
    print(i)

12
13
14
15

for i in range(0, 10, 2):
    print(i)

0
2
4
6
8

for i in range(5, -1, -1):
    print(i)

5
4
3
2
1
0
```

## Module importieren

- Das Schlüsselwort import
- Der Name des Modules
- Optional weitere Modulnamen, durch Kommata getrennt

```
import random
for i in range(5):
    print(random.randint(1, 10))
```

```
6
9
1
1
9
```

```
import random, sys, os, math
```

```
import sys
```

```
while True:
    print('Bitte exit eingeben um aufzuhören')
    response = input()
    if response == 'exit':
        sys.exit()
    print('Deine Eingabe war ' + response + '.')
```

```
Bitte exit eingeben um aufzuhören
weiter
Deine Eingabe war weiter.
Bitte exit eingeben um aufzuhören
aufhören
Deine Eingabe war aufhören.
Bitte exit eingeben um aufzuhören
exit
```

An exception has occurred, use %tb to see the full traceback.

SystemExit

```
/home/michl/.local/lib/python3.5/site-packages/IPython/core/interactiveshell.py:2889: UserWarning:
warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

```

import platform, datetime, os, subprocess, socket
today = datetime.date.today()
print('Heut ist: ' + str(today))
system = platform.system()
print('System: ' + system)
node = platform.node()
print('Rechner heist: ' + node)
arch = platform.machine()
print('Die Platform ist: ' + arch)
#user = os.getlogin()
#space = os.statvfs('/home/'+user)
#freespace = (space.f_frsize * space.f_bavail)/1024/1024

get_uptime = subprocess.Popen('uptime', stdout=subprocess.PIPE)
uptime = get_uptime.stdout.read()
print('Uptime ist: ' + str(uptime))

myip = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
myip.connect(('8.8.8.8', 80))
getip = myip.getsockname()[0]
myip.close()
print('Ip ist: ' + getip)

Heut ist: 2016-11-03
System: Linux
Rechner heist: michl-laptop
Die Platform ist: x86_64
Uptime ist: b' 15:01:02 up 1 day,  8:34,  3 users,  load average: 0,53, 0,65, 0,76\n'
Ip ist: 192.168.56.200

```