

# Python

...für Anfänger

## Unit 15

by

Dr. Günter Kolousek

# Kommandozeilenverarbeitung - 1

```
# cmd.py  
import sys
```

```
print(sys.argv)
```

```
$ python cmd.py  
['cmd.py']  
$ python cmd.py abc def  
['cmd.py', 'abc', 'def']
```

# Kommandozeilenverarbeitung - 2

```
import sys
```

```
if __name__ == "__main__":  
    print(sys.argv, len(sys.argv))
```

```
    i = 1  
    while i < len(sys.argv):  
        print(sys.argv[i])  
        i += 1
```

```
$ python cmd.py abc def  
['cmd.py', 'abc', 'def'] 3  
abc  
def
```

# Kommandozeilenverarbeitung - 3

- ▶ Parameter bzw. Argumente
  - ▶ in spitzen Klammern oder zur Gänze in Großbuchstaben
- ▶ Optionen
  - ▶ mit -: Kurze Form, wie -a, -b, -c oder -abc
  - ▶ mit --: Lange Form, kann Wert haben, wie --input ARG oder --input=ARG muss jedoch nicht --jump
- ▶ in eckigen Klammern bedeutet optional, wie [`<number>`]
- ▶ | bedeutet Variante, wie in `<name> | <number>`
- ▶ . . . bedeutet "beliebig oft", wie in `<name> . . .`
- ▶ in runden Klammern zum Gruppieren, wie in `( <x> <y> ) . . .`

# Sortieren mittels BubbleSort - 1

1. Beginne vorne.
2. Vergleiche jeweils 2 benachbarte Zahlen und vertausche diese, wenn diese nicht in der richtigen Reihenfolge sind. Danach ist sicher die größte Zahl am rechten Ende, aber alle anderen Zahlen sind unsortiert.
3. Beginne deshalb wieder von vorne
4. Name: Jeweils größtes Element wie eine Blase an die Oberfläche
5. Beispiel
  - ▶ [15, 2, 43, 17, 4, 8, 47]
  - ▶ [2, 15, 43, 17, 4, 8, 47]
  - ▶ [2, 15, 43, 17, 4, 8, 47]
  - ▶ [2, 15, 17, 43, 4, 8, 47]
  - ▶ ...

# Sortieren mittels BubbleSort - 2

- ▶ Prinzip

  - für jedes Element der Liste:

    - für jedes y der Liste bis zum Vorletzten:

      - wenn y größer ist als dessen Nachfolger:

        - vertausche y mit seinem Nachfolger

- ▶ Algorithmus

```
def bubble(lst):  
    for x in lst[:-1]:  
        for i, y in enumerate(lst[:-1]):  
            if y > lst[i + 1]:  
                lst[i], lst[i+1] = lst[i+1], lst[i]  
    return lst
```