



Dateien

Dateien



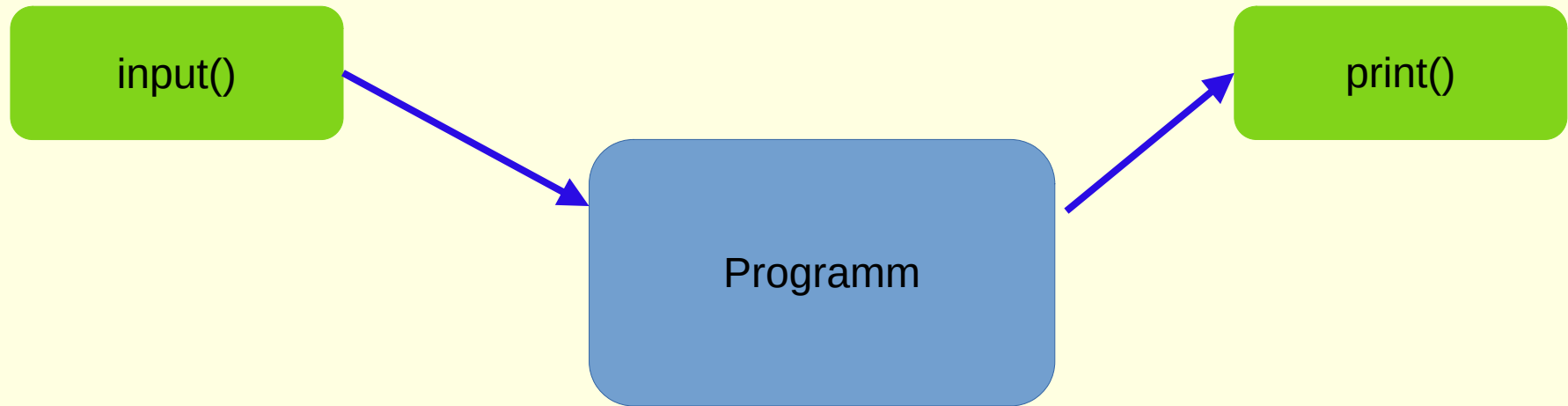
- Ergebnisse unserer Programme verschwinden
- Eingaben im Programmtext oder via Kommandozeile
- Ausgaben nur auf Bildschirm

Dateien

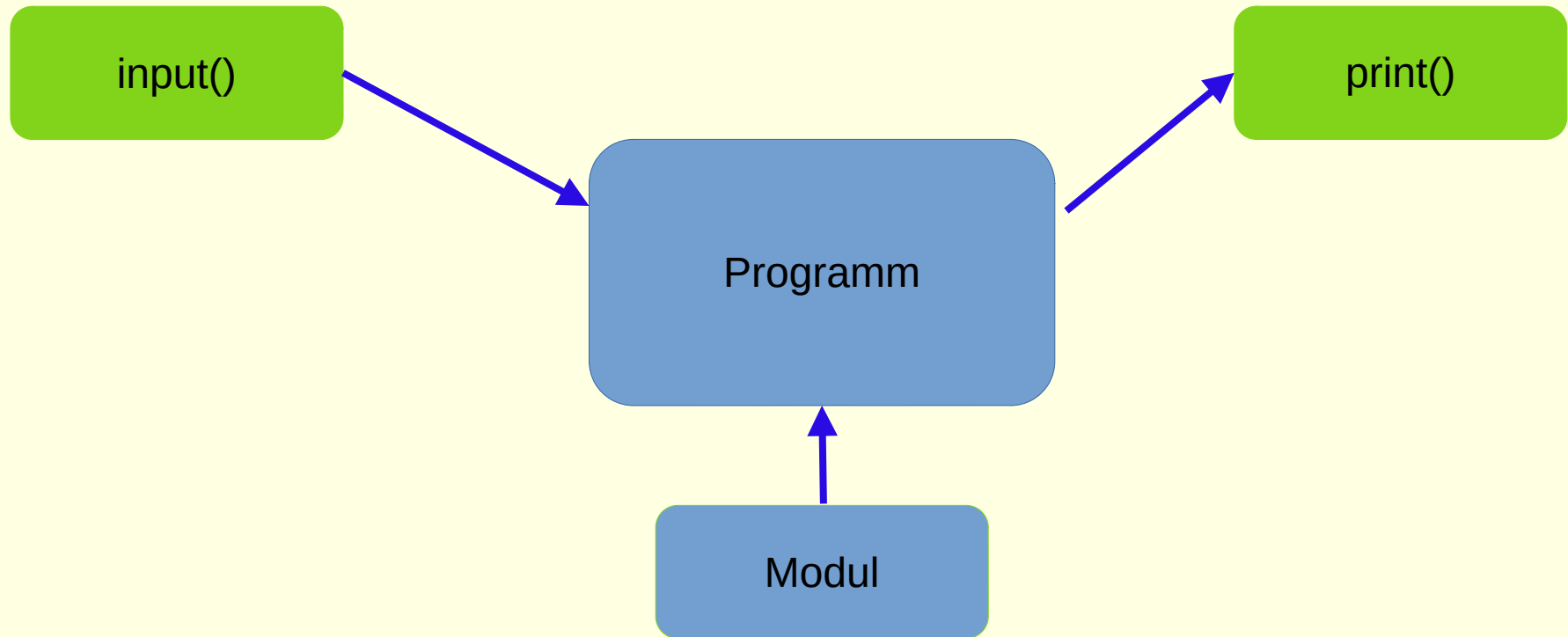


Programm

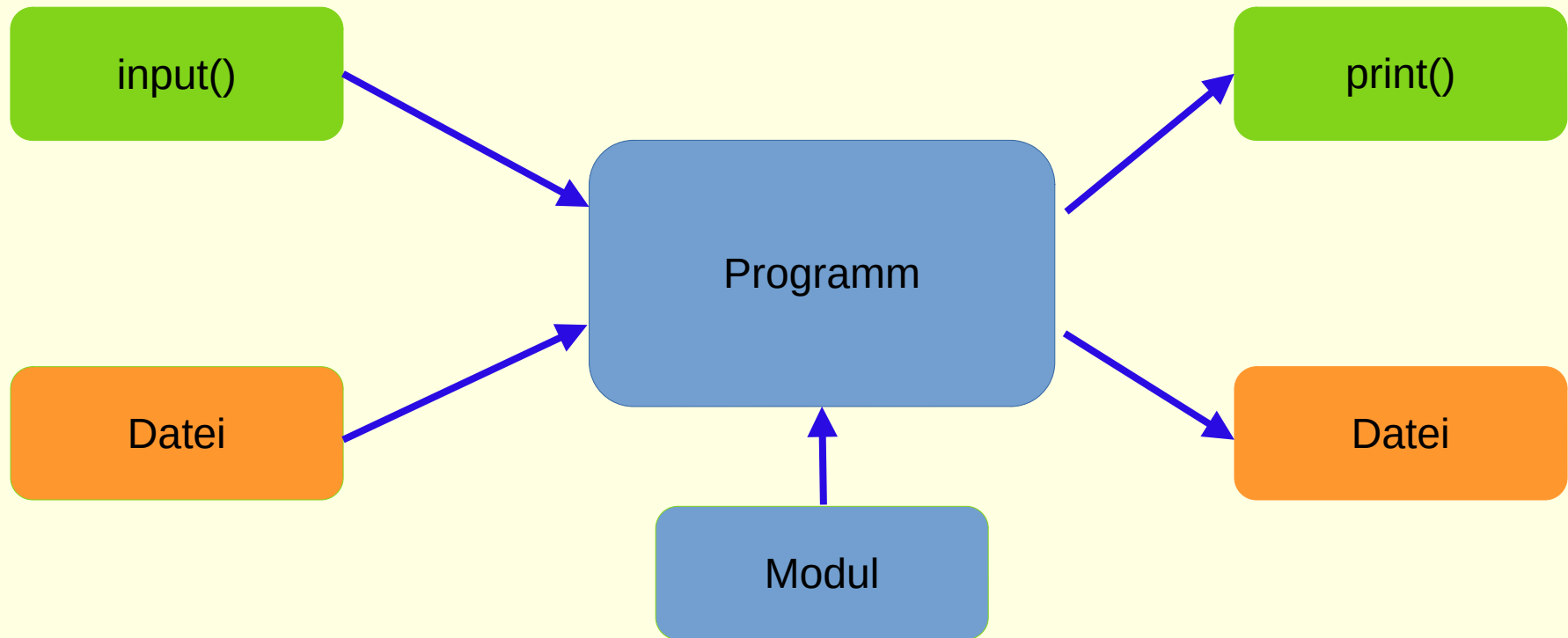
Dateien



Dateien



Dateien



Dateien



Dateien



Mit der "Außenwelt" kommunizieren Programme über
Ströme / Streams

- kontinuierliche Folge von Daten
- sowohl eingehende als auch ausgehende

Dateien



Zwei Standarddatenströme:

- stdin für Eingabe
- stdout für Ausgabe
- Interaktion indirekt über `input()`, `print()`

Dateien

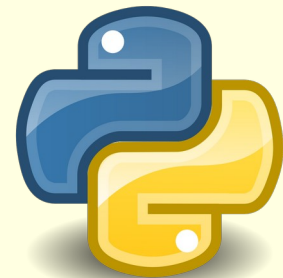


Ströme können in verschiedenen Modi arbeiten

Modus:

- Format des Zugriffs
- Art des Zugriffs

Dateien



Lesezugriff

1) Öffnen

```
1 eingabedatei = open("wetterdaten.txt", "r")
```

Dateien



Lesezugriff

1) Öffnen

```
1 eingabedatei = open("wetterdaten.txt", "r")
```

2)...

3) Schließen

```
1 eingabedatei.close()
```

Dateien



Die Datei wetterdaten.csv:

```
Augsburg;18,4;0  
Bremen;15,2;27  
Duisburg;16,9;18  
Stralsund;14,5;123  
Zwickau;9,7;0
```

- unter Dateien/src/

Dateien



Die Datei wetterdaten.txt:

```
eingabe_datei = open("wetterdaten.txt", "r")  
  
eingabe_datei.close()
```

Dateien



Die Datei wetterdaten.txt:

```
1 eingabe_datei = open("wetterdaten.txt", "r")  
2  
3 print(type(eingabe_datei))  
4  
5 eingabe_datei.close()
```

<class '_io.BufferedReader'>

Dateien



Die Datei wetterdaten.txt:

```
1 eingabe_datei = open("wetterdaten.txt", "r")
2
3 print(type(eingabe_datei))
4
5 eingabe_datei.close()
```

<class '_io.BufferedReader'>

```
class io.TextIOWrapper(buffer, encoding=None, errors=None, newline=None,
line_buffering=False, write_through=False)
```

A buffered text stream providing higher-level access to a [BufferedIOBase](#) buffered binary stream. It inherits from [TextIOBase](#).

Dateien



io.BufferedReader ist iterable

```
1 eingabe_datei = open("wetterdaten.txt", "r")
2
3 print(type(eingabe_datei))
4
5 eingabe_datei.close()
```

<class '_io.BufferedReader'>

Dateien



wollen: `split(";")` der eingelesenen Zeilen

```
1 eingabe_datei = open("wetterdaten.txt", "r")
2
3 for zeile in eingabe_datei:
4     print(zeile.split(";"))
5
6 eingabe_datei.close()
```

Dateien



wollen: `split(";")` der eingelesenen Zeilen

```
['Augsburg', '18,4', '0\n']  
['Bremen', '15,2', '27\n']  
['Duisburg', '16,9', '18\n']  
['Stralsund', '14,5', '123\n']  
['Zwickau', '9,7', '0\n']
```

Dateien



`string.strip(characters)`

- entfernt alle Vorkommen der Zeichen im String characters
- am Beginn und Ende des Strings
- default für characters sind whitespaces

Dateien



erst `strip()`, dann
`split(";")` der eingelesenen Zeilen

```
1 eingabe_datei = open("wetterdaten.txt", "r")
2
3 for zeile in eingabe_datei:
4     zeile = zeile.strip()
5     print(zeile.split(";"))
6
7 eingabe_datei.close()
```

Dateien



erst `strip()`, dann
`split(";")` der eingelesenen Zeilen

```
['Augsburg', '18,4', '0']  
['Bremen', '15,2', '27']  
['Duisburg', '16,9', '18']  
['Stralsund', '14,5', '123']  
['Zwickau', '9,7', '0']
```

Dateien



Sicherer als `open() ... close()`

- `with open() as ... :`

```
1 with open("wetterdaten.txt", "r") as eingabe_datei:
2     for zeile in eingabe_datei:
3         zeile = zeile.strip()
4         print(zeile.split(";"))
```

- schließt Datei nach Ende der Einrückung

Dateien



Aufgabe

- Eingabe von Tastatur einlesen
 - falls in Liste der Städte, Wetterdaten für die Stadt ausgeben
 - falls nicht in Liste, weiter
- Schleife, bis Eingabe "Ende"

Dateien



Modi für open()

"r" - Read – Default, zum Lesen

"a" - Append – zum Anhängen

"w" - Write – zum Schreiben, ggf. Erzeugen

"x" - Create – zum Erzeugen

zusätzlich möglich:

"t" - Text (Default)

"b" - Binary

"+" - Lesen und Schreiben

Dateien



Schreiben in eine Datei

```
1 with open("neueDatei.txt", "w") as ausgabe_datei:
```

Dateien



Schreiben in eine Datei

```
1 with open("neueDatei.txt", "w") as ausgabe_datei:  
2     for i in range(5):  
3         ausgabe_datei.write(f"Zeile {i}\n")
```

Dateien



Schreiben in eine Datei

```
1 with open("neueDatei.txt", "w") as ausgabe_datei:  
2     for i in range(5):  
3         ausgabe_datei.write(f"Zeile {i}\n")
```

- analog zu write(): read() und readline()
- read(size): nur size bytes, sonst alles
- s. Tabelle 6.2

Dateien



Schreiben in eine Datei

```
1 with open("neueDatei.txt", "w") as ausgabe_datei:  
2     for i in range(5):  
3         ausgabe_datei.write(f"Zeile {i}\n")
```

- `readlines()` liest alle Zeilen in Liste

Dateien



Steuern der Schreib-/Leseposition

- `tell()` liefert Position
- `seek(pos)` setzt Position

Dateien



What is the output of the code?

```
1 datei = open("txt.txt", "r")  
2 print(datei.read(2))
```

given txt.txt

```
This is LINE 1  
This is LINE 2  
This is LINE 3
```

- a) The first 2 characters *Th* is printed
- b) The first 2 lines *This is LINE 1* and *This is LINE 2* is printed
- c) The first 2 characters *Th* is skipped *is is LINE 1* is printed
- d) The 2nd line *This is LINE 2* is printed

Dateien



You want to access the test.txt file and print each line in it.
Which option can you use? (Select all that apply)

```
1 file = open(test.txt)
2 # insert code here
3 file.close()
```

- a) print(file.readlines())
- b) print(read.file(test.txt))
- c) print(file.readlines())
- d) for l in file: print(l)
- e) print(file.lines())
- f) print(file.read())