

24 Ausgabe mit print()

24.1 Beschreibung

- `print()` wird zur Ausgabe auf die Konsole (Bildschirm) verwendet
- die Funktion `print()` erwartet in den runden Klammern den Inhalt, der ausgegeben werden soll
- in den runden Klammern können stehen Literale, Variablenname oder Ausdrücke von Operationen und Funktionsaufrufen
- es sind beliebig viele Werte und Ausdrücke erlaubt, sie werden durch Komata getrennt
- bei der Ausgabe werden alle Werte, durch Leerzeichen voneinander getrennt, dargestellt
- nach jeder Ausgabe erfolgt einen Zeilenumbruch
- Escape-Sequenzen werden zur Steuerung der Ausgabe verwendet
 - `\n` erzwingt Zeilenvorschub
 - `\"` einfügen Anführungszeichen in Zeichenkette
 - `\t` erzwingt Tabulatorvorschub
- Trennzeichen zwischen den Werten kann mit **sep()** geändert werden
- Zeilenumbruch nach der Ausgabe kann mit **end()** angepasst werden

- Beispiele:

```
# Ganzzahlen und Gleitpunktzahl ausgeben
```

```
x = 1.5
```

```
y = 10
```

```
print(x, y)
```

```
1.5 10
```

```
# Ganzzahlen, Gleitpunktzahl mit Text ausgeben
```

```
x = 1.5
```

```
y = 10
```

```
print("x:", x, "y:", y)
```

```
x: 1.5 y: 10
```

```
# Ausdruck mit Text ausgeben
```

```
x = 1.5
```

```
y = 10
```

```
print("Summe von x und y", x+y)
```

```
Summe von x und y 11.5
```

```
# Ausdruck mit Text ausgeben - Trennzeichen geändert
x = 1.5
y = 10

print("Summe von x und y", x+y, sep=" : ")
```

```
Summe von x und y : 11.5
```

```
# Ausdruck mit Text ausgeben - kein Zeilenumbruch nach der Ausgabe
x = 1.5
y = 10

print("Summe von x und y", x+y, end=" ")
print("alles steht in einer Zeile")
```

```
Summe von x und y 11.5 alles steht in einer Zeile
```

24.2 Formatierte Ausgabe mit print()

- für tabellarische Druckaufbereitung
- `f"normaler Text {Ausdruck} mehr normaler Text {weiterer Ausdruck:Format} ..."`
 - Ausdruck steht für Variable, Zahl, Berechnungsanweisung usw.
 - der Ausdruck wird zuerst evaluiert (ausgerechnet) und dann aufbereitet
 - die geschweiften Klammern werden nicht in den druckaufbereiteten Text übernommen
 - Angabe der Ausgabelänge z.B. 20 `f"...{variable:20}"`
 - Angabe der Ausrichtung
 - links `f"...{variable:<20}"`
 - rechts `f"...{variable:>20}"`
 - zentriert `f"...{variable:^20}"`
- Formatierungszeichen für Ganzzahlen (int)
 - vorzeichenbehaftet ausgeben `f"|{var:+15}"` oder `f"|{var:-15}"`
 - mit führenden Nullen `f"|{var:+015}"` oder `f"|{var:-015}"`
 - dezimale Ausgabe (Standard) keine weitere Angabe erforderlich
 - binäre Ausgabe `f"|{var:+08b}"`
 - oktale Ausgabe `f"|{var:+08o}"`
 - hexadezimale Ausgabe `f"|{var:+08x}"`
- Formatierungszeichen für Gleitpunktzahlen (float)
 - Anzahl der Nachkommastellen `f"|{var:+6.2f}"`
 - wissenschaftliche Schreibweise `f"|{var:+6.2e}"`

- Beispiele:

```
# Ausdruck formatiert ausgeben
x = 1.5
y = 10
```

```
print(f"Summe von x und y = {x+y}")
```

```
Summe von x und y = 11.5
```

```
# tabellarische Ausgabe mit Längenangabe
name1 = "Ben"
score1 = 9001
name2 = "Joe"
score2 = 666
print(f"{name1:20} : {score1:5} ")
print(f"{name2:20} : {score2:5} ")
```

```
Ben           : 9001
Joe           : 666
```

```
# tabellarische Ausgabe mit Längenangabe und Ausrichtung
text = "sample"
value = 123
print(f" |{{text:15}} | |{{value:15}} |" )
print(f" |{{text:<15}} | |{{value:<15}} |" )
print(f" |{{text:>15}} | |{{value:>15}} |" )
print(f" |{{text:^15}} | |{{value:^15}} |" )
```

```
|sample          | |          123 |
|sample         | |123         |
|          sample | |          123 |
|          sample | |          123 |
```

```
# tabellarische Ausgabe mit Längenangabe und Vorzeichen
pos = 10
neg = -10
print(f" |{{pos: 015}} | |{{neg: 015}} |" )
print(f" |{{pos:+015}} | |{{neg:+015}} |" )
print(f" |{{pos:+015}} | |{{neg:+015}} |" )
```

```
| 000000000000010 | |-000000000000010 |
|+000000000000010 | |-000000000000010 |
|+000000000000010 | |-000000000000010 |
```

```
# 8-stellige Ausgabe mit führenden Nullen der Zahl 123
# im dezimalen, oktalen und hexadezimalen Format
print(f" |123:08d|" )
print(f" |123:08o|" )
print(f" |123:08x|" )
```

```
00000123
00000173
0000007b
```

```
# unterschiedlich formatierte Ausgaben einer Gleitpunktzahl
num = 12.345
print(f" |{{num}} |" )
print(f" |{{num:f}} |" )
print(f" |{{num:6.2f}} |" )
print(f" |{{num:<6.1f}} |" )
print(f" |{{num:06.2f}} |" )
print(f" |{{num:+06.2f}} |" )
print(f" |{{num: 06.2f}} |" )
```

```
|12.345 |
|12.345000 |
| 12.35 |
|12.3 |
|012.35 |
|+12.35 |
| 12.35 |
```

```

# unterschiedlich formatierte Ausgaben einer Gleitpunktzahl
# in der wissenschaftlichen Schreibweise
num = 12.345
print(f"|{num:e} |")
print(f"|{num:6.2e} |")
print(f"|{num:<6.1e} |")
print(f"|{num:06.2e} |")
print(f"|{num:+06.2e} |")
print(f"|{num: 06.2e} |")

```

```

|1.234500e+01 |
|1.23e+01 |
|1.2e+01 |
|1.23e+01 |
|+1.23e+01 |
| 1.23e+01 |

```

Aufgabe

1. Im Programm „TicTacToe“ sollen die Ausgabe des Spielfeldes mit Hilfe von Formatierungsangaben erfolgen.
2. Das Programm zur Personenverwaltung mit dem Datentyp Assoziatives Array (Dictionary) ist mit einer tabellarischen Ausgabe zu erweitern:
Vornamen, Geburtsjahr und Telefonnr sind spaltenweise ausgerichtet auszugeben.