

15 Primitive Datentypen und Operationen

15.1 Ganzzahl (Integer)

- Ganzzahlige Werte werden in Python durch den Datentyp **int** repräsentiert
- sie haben keine Nachkommastellen
- Integer-Datentypen werden genau dargestellt
- es treten keine Rundungsfehler auf
- in Python kann man theoretisch beliebig große ganzzahlige Werte speichern
- neben den Werten im Dezimalsystem sind auch Darstellungen im Oktal-, Hexadezimal und Dualsystem möglich

- Beispiel:

```
# Dezimalsystem, Basis 10)
ganzzahl = 12345

# Dualsystem, Basis 2)
dualzahl = 0b10101

# Oktalsystem, Basis 8)
oktalzahl = 0o11753

# Hexadezimalsystem, Basis 16)
hexazahl = 0xAB11F

print (ganzzahl, dualzahl, oktalzahl, hexazahl)
```

Ausgabe

```
12345 21 5099 700703
```

15.2 Gleitkommazahl (Float)

- Gleitkommazahlen oder Fließkommazahlen werden in Python durch den Datentyp **float** repräsentiert
- die Nachkommastellen werden durch einen Punkt vom ganzzahligen Anteil getrennt
- Float-Datentypen können ggf. nicht genau dargestellt werden
- es kann zu Rundungsfehlern kommen
- es ist möglich sehr kleine oder sehr große Werte anzugeben

- Beispiel 1:

```
float_zahl1 = 12345.0
float_zahl2 = 0.00000000000000012345
float_zahl3 = 12345678901234567890.1

print (float_zahl1, float_zahl2, float_zahl3)
```

Ausgabe

```
12345.0 1.2345e-15 1.2345678901234567e+19
```

- Beispiel 2:

```
a = 7.07
b = 5.05
print (a+b)
```

Ausgabe

```
12.1200000000000001
```

15.3 Zeichenketten (String)

- Zeichenketten werden in Python durch den Datentyp **str** repräsentiert
- Werte vom Zeichenketten-Typ enthalten beliebig viele Zeichen
- Zeichenketten oder Strings sind eine Folge von Zeichen, die wahlweise in einfachen oder doppelten Anführungszeichen geschrieben werden

- Beispiel:

```
str1 = "Hallo"  
str2 = 'Welt'  
str3 = "2%&/(@'+##`*'"
```

```
print (str1,str2,str3)
```

Ausgabe

```
Hallo Welt 2%&/(@'+##`*'"
```

15.4 Einfache Operationen

- mit den arithmetischen Operatoren (+, -, /, *) kann in Python addiert, subtrahiert, dividiert und multipliziert werden
- mit Hilfe des Zuweisungsoperators = werden die Werte bzw. Ergebnisse einer Berechnung einem Namen zugewiesen
- der Operator + verkettet Strings (Zeichenketten)

- Beispiel:

```
# Ganzzahl (Integerwerte)
x = 10
y = 20
summe = x + y
produkt = x * y

# Gleitkommazahl (Floatwerte)
x = 2.4
y = 1.2
quotient = x / y

# Zeichenkette (String)
text1 = "Hallo"
text2 = " wie geht es?"
text = text1 + text2

print (summe, produkt, quotient, text)
```

Ausgabe

```
30 200 2.0 Hallo wie geht es?
```

15.5 Literale für primitive Datentypen

15.6 Variablen und Konstanten

15.7 Operatoren für Zeichenketten (Strings)

- Zeichenketten oder Strings sind eine Folge von Zeichen
- mit der Index-Schreibweise kann auf einzelne Zeichen oder einen Bereich von Zeichen zugegriffen werden, jedoch lassen sich einzelne Zeichen so nicht ändern
- Verketteten von Zeichenketten mit **+**
- Vervielfachen von Zeichenketten mit *****
- Zugriff auf ein Element per Index (Zählweise beginnt bei 0) mit **[i]**
- Zugriff auf einen Bereich von Zeichen per Index (Zählweise beginnt bei 0) mit **[a:b]**
- Prüfen, ob Zeichen in einer Zeichenkette enthalten sind mit **in**
- Prüfen, ob Zeichen in einer Zeichenkette NICHT enthalten sind mit **not in**

- Beispiel:

```
wort = "Rasen"
print(wort)
for b in wort:
    print(b, type(b), type(wort))
print(wort[2:5])
# wort[1] = "o" TypeError: 'str' object does not support item assignment
wort_neu = wort[0] + "o" + wort[2:5]
print(wort_neu)
```

```
Rasen
R <class 'str'> <class 'str'>
a <class 'str'> <class 'str'>
s <class 'str'> <class 'str'>
e <class 'str'> <class 'str'>
n <class 'str'> <class 'str'>
sen
Rosen
```

15.8 Länge einer Zeichenkette (Anzahl der Zeichen)

- Länge einer Zeichenkette (Anzahl der Elemente)

len(str)

- Beispiele:

```
# Länge einer Zeichenkette ausgeben
str = "Python"
print("Länge der Zeichenkette 'str': ", len(str))

# Index-Bereich einer Zeichenkette ausgeben
print(str[2:])

# Zeichen und Index einer Zeichenkette ausgeben
for i in range(0, len(str)):
    print("Index", i, "Wert", str[i])
```

```
[Länge der Zeichenkette 'str': 6
Thon
Index 0 Wert P
Index 1 Wert y
Index 2 Wert t
Index 3 Wert h
Index 4 Wert o
Index 5 Wert n]
```

15.9 Methoden (Funktionen) für Zeichenketten (Strings)

- Erstes Zeichen in Großbuchstaben umwandeln `.capitalize()`
- Vorkommen eines oder mehrere Zeichen in der Zeichenkette zählen
optional mit Index bzw. Index-Bereich `.count("xy")`
- Prüft, ob die Zeichenkette mit bestimmten Zeichen endet
optional mit Index bzw. Index-Bereich `.endswith("abc")`
- Sucht nach dem ersten Vorkommen von Zeichen in einer Zeichenkette
optional mit Index bzw. Index-Bereich `.find("jkl")`
- Prüft, ob eine Zeichenkette nur aus Buchstaben und Ziffer besteht `.isalnum()`
- Prüft, ob eine Zeichenkette nur aus Buchstaben besteht `.isalpha()`
- Prüft, ob eine Zeichenkette nur aus Ziffer besteht `.isdigit()`
- Prüft, ob eine Zeichenkette nur aus Kleinbuchstaben besteht `.islower()`
- Prüft, ob eine Zeichenkette nur aus Großbuchstaben besteht `.isupper()`
- Setzt ein Trennzeichen zwischen alle Zeichen `.join("-")`
- Wandelt die Zeichenkette in Kleinbuchstaben `.lower()`
- Wandelt die Zeichenkette in Großbuchstaben `.upper()`
- Ersetzt Zeichen durch andere Zeichen `.replace("abc", "xyz")`
- Teilt eine Zeichenkette in Bestandteile auf `.split("abc", "xyz")`

- Beispiel:

```
str = "python-Programm"
print(str)
# Erstes Zeichen in Großbuchstaben umwandeln
print(str.capitalize())

python-Programm
Python-programm

# Vorkommen eines oder mehrere Zeichen in der Zeichenkette zählen
# optional mit Index bzw. Index-Bereich
print("Anzahl p in ", str, ":", str.count("p"))

Anzahl p in  python-Programm : 1

# Prüft, ob die Zeichenkette mit bestimmten Zeichen endet
# optional mit Index bzw. Index-Bereich
if str.endswith("mm"):
    print(str, "endet mit mm")

python-Programm endet mit mm

# Sucht nach dem ersten Vorkommen von Zeichen in einer Zeichenkette
# optional mit Index bzw. Index-Bereich
print("Position von gramm in ", str, ":", str.find("gramm"))

Position von gramm in  python-Programm : 10
```



```

# Prüft, ob eine Zeichenkette nur aus Buchstaben oder Ziffern besteht
# Prüft, ob eine Zeichenkette nur aus Buchstaben besteht
# Prüft, ob eine Zeichenkette nur aus Ziffern besteht
# Prüft, ob eine Zeichenkette nur aus Kleinbuchstaben besteht
# Prüft, ob eine Zeichenkette nur aus Großbuchstaben besteht
str = "Internet"
print(str)
if str.isalnum():
    print(str, "besteht nur aus Buchstaben und Ziffern")
if str.isalpha():
    print(str, "besteht nur aus Buchstaben")
if str.isdigit():
    print(str, "besteht nur aus Ziffern")
if str.islower():
    print(str, "besteht nur aus Kleinbuchstaben")
if str.isupper():
    print(str, "besteht nur aus Großbuchstaben")

Internet
Internet besteht nur aus Buchstaben und Ziffern
Internet besteht nur aus Buchstaben-

# Setzt ein Trennzeichen zwischen alle Zeichen
str = "-"
print(str.join("Python-Programm"))

P-y-t-h-o-n---P-r-o-g-r-a-m-m

# Wandelt die Zeichenkette in Kleinbuchstaben
print("in Kleinbuchstaben: ", str.lower())

in Kleinbuchstaben: python-programm

# Wandelt die Zeichenkette in Großbuchstaben
print("in Großbuchstaben: ", str.upper())

in Großbuchstaben: PYTHON-PROGRAMM

# Ersetzt Zeichen durch andere Zeichen
str = str.replace("Python", "C#")
print(str)

C#-Programm

# Teilt eine Zeichenkette in Bestandteile auf
str = "Das ist ein Python-Programm"
print(str.split())
print(str.split("-"))

['Das', 'ist', 'ein', 'Python-Programm']
['Das ist ein Python', 'Programm']

```

Aufgabe

- Das Spiel Tic Tac Toe (Drei-gewinnt) ist mit Python zu programmieren.
- Es gibt 2 Spieler, die abwechselnd in einem 3x3 großen Spielfeld ein **x** bzw. ein **o** setzen.
- Wer zuerst 3 Felder in einer Reihe, einer Spalte oder in der Diagonalen belegt, hat gewonnen.
- Das Spielfeld sollte im Programm über den Datentyp Liste abgebildet werden
- Bei der Programmerstellung sollten folgende Funktionalitäten beachtet werden
 - Hauptroutine festlegen
 - Spielfeld erstellen und ausgeben
 - Spielzug erfassen
 - Spielzug in das Spielfeld einarbeiten
 - Sieger ermitteln
 - Spiel vorzeitig beenden