```
Python String (str)                          (C) 2020-2021 T.Birnthaler OSTC GmbH
===================

Doku --> http://docs.python.org/3/library/stdtypes.html#str
     --> http://docs.python.org/3/library/stdtypes.html#string-methods

Strings in Python sind seit Python 3.0 Sequenzen von Unicode-Einzelzeichen
codiert im Format UTF-8, d.h. jedes Zeichen belegt 1-4 Bytes. In Python 2.x
wurden Strings noch durch Einzelbyte-Zeichen dargestellt (z.B. ISO-8859).

  "hallo welt"    # Unicode-String (in P3 automatisch)
  u"hallo welt"   # Unicode-String (in P2 notwendig, in P3 Standard)
  ""              # Leerer String

Folgende Operationen werden von Strings zusÃ¤tzlich zu den allgemeinen und
immutable Sequenz-Operationen unterstÃ¼tzt (R, S, T sind Strings, I, J, W sind
Ganzzahlen, C ist ein Zeichen, D ist ein Dictionary, L ist eine Liste von
Strings, ENC ist ein Encoding):

ACHTUNG: Die Sequenz S bleibt unverÃ¤ndert, neuer String kommt als Ergebnis zurÃ¼ck.

+----------------------+-------------------------------------------------------+
| Operation            | Bedeutung                                             |
+----------------------+-------------------------------------------------------+
| S.lower()            | Kleinschreibung                                       |
| S.upper()            | Grossschreibung                                       |
| S.swapcase()         | Gross/Kleinschreibung vertauschen                     |
| S.capitalize()       | 1. Buchstaben gross schreiben                         |
| S.title()            | Alle Worte beginnen mit Grossbuchstaben, Rest klein   |
+----------------------+-------------------------------------------------------+
| S.startswith(T,I,J)  | Beginnt mit Text T?                                   |
| S.endswith(T,I,J)    | Endet mit Text T?                                     |
+----------------------+-------------------------------------------------------+
| S.removeprefix(T)    | Falls S mit T beginnt, T entfernen              3.9   |
| S.removesuffix(T)    | Falls S mit T endet, T entfernen                3.9   |
+----------------------+-------------------------------------------------------+
| S.count(T,I,J)       | Anzahl Vorkommen von T                                |
| S.find(T,I,J)        | Von links Treffer-Index von T finden (-1 wenn nein)   |
| S.index(T,I,J)       | Wie S.find(), ValueError statt -1 falls kein Treffer  |
| S.rfind(T,I,J)       | Von rechts Treffer-Index von T finden (-1 wenn nein)  |
| S.rindex(T,I,J)      | Wie S.rfind(), ValueError statt -1 falls kein Treffer |
+----------------------+-------------------------------------------------------+
| S.isalnum()          | Nur Buchstaben oder Ziffern? (mind. 1)                |
| S.isalpha()          | Nur Buchstaben? (mind. 1)                             |
| S.isdigit()          | Nur Ziffern? (mind. 1)                                |
| S.islower()          | Nur Kleinbuchstaben? (mind. 1)                        |
| S.isspace()          | Nur Leerraum? (mind. 1)                               |
| S.istitle()          | Alle Worte starten mit Grossbuchstaben, Rest klein?   |
| S.isupper()          | Nur Grossbuchstaben? (mind. 1)                        |
+----------------------+-------------------------------------------------------+
| S.lstrip(C)          | Leerraum oder Zeichen C links entfernen               |
| S.rstrip(C)          | Leerraum oder Zeichen C rechts entfernen              |
| S.strip(C)           | Leerraum oder Zeichen C links + rechts entfernen      |
+----------------------+-------------------------------------------------------+
| S.center(W,C)        | Zentriert in Breite W mit FÃ¼llz. C (STD: Space)       |
| S.ljust(W,C)         | LinksbÃ¼ndig in Breite W mit FÃ¼llz. C (STD: Space)     |
| S.rjust(W,C)         | RechtsbÃ¼ndig in Breite W mit FÃ¼llz. C (STD: Space)    |
| S.zfill(W)           | Links in Breite W mit "0" auffÃ¼llen                   |
+----------------------+-------------------------------------------------------+
| S.join(L)            | Verketten der Elemente in L getrennt durch S          |
| S.split(C,MAX)       | Teile-Liste von links anhand Trennzeichen C           |
| S.rsplit(C,MAX)      | Teile-Liste von rechts anhand Trennzeichen C          |
| S.splitlines()       | Zeilen-Liste anhand Zeilentrenner "\n"                |
+----------------------+-------------------------------------------------------+
| S.partition(C)       | Zerlegen in (Head, C, Tail) am 1. C von links         |
| S.rpartition(C)      | Zerlegen in (Head, C, Tail) am 1. C von rechts        |
+----------------------+-------------------------------------------------------+
| S.replace(T,R,MAX)   | T durch R ersetzen (alle oder Anzahl MAX)             |
| S.translate(D)       | Zeichen-Key in D durch Value ersetzen oder lÃ¶schen    |
| S.expandtabs(W)      | Tabulatoren (Breite W) durch Leerzeichen ersetzen     |
+----------------------+-------------------------------------------------------+
| S.decode(ENC,ERR)    | Decodieren gemÃ¤Ã-^_ Codierung ENC                     |
| S.encode(ENC,ERR)    | Codieren gemÃ¤Ã-^_ Codierung ENC                       |
| S.format(...)        | Werte ... fÃ¼r Platzhalter {} in S einsetzen           |
+----------------------+-------------------------------------------------------+

* ERR kann "strict", "ignore", "replace" und "xmlcharrefreplace" sein.

* Leerraum (Whitespace) ist eine beliebig lange Folge folgender Zeichen:

    +-----+----+--------------------------+
    | \t  |  9 | (Horizontaler) Tabulator |
```

\n	10	Newline
\v	11	Vertikaler Tabulator
\f	12	Form Feed
\r	13	Carriage Return
" "	32	Leerzeichen