```
ÃM-^Dnderungen Python 2.x --> 3.x              (C) 2014-2021 T.Birnthaler OSTC GmbH
============================

Aktivieren von Eigenschaften neuerer Python-Versionen (z.B. Python 3) in
Ã¤lteren Python-Versionen (z.B. Python2):

  import __future__                          # Alle Eigenschaften
  from __future__ import *                   # FEHLER!
                                             # STD  PEP (Standard ab Version N.M)
  from __future__ import annotations         # 4.0  563 Type annotation
  from __future__ import generator_stop      # 3.7  479 Generator stop
  from __future__ import unicode_literals    # 3.0 3112 Unicode literals in Python 3000
  from __future__ import print_function      # 3.0 3105 Make "print" a function
  from __future__ import with_statement      # 2.6  343 The "with" statement
  from __future__ import absolute_import     # 3.0  328 Import: multi-line + absolute/Relative
  from __future__ import division            # 3.0  238 Changing division operator
  from __future__ import generators          # 2.3  255 Simple generators
  from __future__ import nested_scopes       # 2.2  227 Statically nested scopes
  from __future__ import all_feature_names   # ?.? TODO
  from __future__ import barry_as_FLUFL      # 3.9 TODO Easteregg (<> statt !=)

Die __future__-Imports MÃM-^\SSEN am Anfang des Quellcodes als 1. Anweisung stehen
(da sie die Verhaltensweise des Python-Interpreters verÃ¤ndern).

Ein kleines "Easteregg" ist auch versteckt (Blockbildung per geschweifte
Klammern {...} wird sicher nie implementiert):

  from __future__ import braces              # --> SyntaxError: not a chance !!!

Unterschiede zwischen Python 2.x und Python 3.x:

  +--------------------------------+--------------------------------+
  |          Python 2.x            |          Python 3.x            |
  +--------------------------------+--------------------------------+
  | "Ã¤Ã¶Ã¼"            (Bytestring) | b"abc"           (Bytestring) | String
  | u"Ã¤Ã¶Ã¼"    (Unicode erzwingen) | "Ã¤Ã¶Ã¼"           (Unicode Std) | String
  | unicode(VAR)         (Unicode) | str(VAR)        (Unicode Std) | Typ
  | unicode                       | str                            | Typ
  | basestring                    | str                            | Typ
  +--------------------------------+--------------------------------+
  | VAR = raw_input(PROMPT)        | VAR = input(PROMPT)            | Direkt lesen
  | VAR = input(PROMPT)            | VAR = eval(input(PROMPT))      | Interpretiert
  +--------------------------------+--------------------------------+
  | print ..., ...    (Statement)  | print(..., ...)    (Funktion)  | Ausgabe
  | print >> sys.stderr ...        | print(..., file=sys.stderr)    | Std: sys.stdout
  | print ...,                     | print(..., end="")             | Std: "\n" (newline)
  | print ... + ...                | print(..., ..., sep="")        | Std: " " (separator)
  | ?                              | print(..., flush="False")      | Nicht puffern
  | "...%d..." % (V1, ...)         | "...{:d}...".format(V1, ...)   | Formatieren
  | `...`              (Backticks) | repr(...)                      | DatenreprÃ¤sentation
  +--------------------------------+--------------------------------+
  | long        2147483648L        | int       2147483648           | Typ
  | 0123                           | 0o123     (oktal)              | Num. Konstante
  |                                | 0b123     (binÃ¤r)              |   "       "
  | sys.maxint                     | sys.maxsize                    |  "       "
  | 12345L      (long-Zahl)        | 12345     (beliebig lang)      | Typ
  | 5   / 4     (Ganzahldivision)  | 5  // 4   (Ganzzahldivision)   | Division
  | ---                            | 5   / 4   (Fliesskommadiv.)    |    "
  | 5.0 / 4     (Fliesskommadiv.)  | 5.0 / 4   (Fliesskommadiv.)    |    "
  | 5   / 4.0   (Fliesskommadiv.)  | 5   / 4.0 (Fliesskommadiv.)    |    "
  | 5.0 / 4.0   (Fliesskommadiv.)  | 5.0 / 4.0 (Fliesskommadiv.)    |    "
  | !=  <>                         | !=                             | Ungleich
  +--------------------------------+--------------------------------+
  | int                            | bool                           | Typ Boolean
  | 1                              | True                           | Wert "wahr"
  | 0                              | False                          | Wert "falsch"
  +--------------------------------+--------------------------------+
  | xrange()                       | range()                        | Umbenennung
  | os.getcwdu()                   | os.getcwd()                    |      "
  | import Tkinter                 | import tkinter                 |      "
  | intern()                       | sys.intern()                   |      "
  | for X in FILE.xreadlines():    | for X in FILE:                 |      "
  +--------------------------------+--------------------------------+
  | L = list(SEQ); L.sort()        | L = sorted(SEQ)                | Sortieren
  +--------------------------------+--------------------------------+
  | DICT.has_key(KEY)              | KEY in DICT                    | Dictionary
  | DICT.iteritems()               | DICT.items()                   |      "
  |     .iterkeys()                |     .keys()                    |      "
  |     .itervalues()              |     .values()                  |      "
  |     .viewitems()               |     .items()                   |      "
  |     .viewkeys()                |     .keys()                    |      "
  |     .viewvalues()              |     .values()                  |      "
```

type(X) == CLASS	isinstance(X, CLASS)	Typvergleich
type(X) is CLASS	isinstance(X, CLASS)	"
except X, T	except X as T	Ausnahmebehandlung
raise Exception, "String"	raise Exception("String")	"
raise Exc, "Str", Traceback	raise E(S).with_traceback(T)	"
StandardError	Exception	"
exec CODE (Statement)	exec(CODE) (Funktion)	
execfile(FILE)	with open(FILE) as fh: exec(fh.read())	
sys.exc_value/type/traceback	sys.exc_info() (-> Tupel)	exc=Exception

Konvertierungs-Programm 2to3 --help

```
Usage: 2to3 [options] file|dir ...

Options:
  -h, --help            Show this help message and exit
  -d, --doctests_only   Fix up doctests only
  -f FIX, --fix=FIX     Each FIX specifies a transformation; default: all
  -j PROCESSES, --processes=PROCESSES
                        Run 2to3 concurrently
  -x NOFIX, --nofix=NOFIX
                        Prevent a transformation from being run
  -l, --list-fixes      List available transformations
  -p, --print-function  Modify the grammar so that print() is a function
  -v, --verbose         More verbose logging
  --no-diffs            Don't show diffs of the refactoring
  -w, --write           Write back modified files
  -n, --nobackups       Don't write backups for modified files
  -o OUTPUT_DIR, --output-dir=OUTPUT_DIR
                        Put output files in this directory instead of
                        overwriting the input files.  Requires -n.
  -W, --write-unchanged-files
                        Also write files even if no changes were required
                        (useful with --output-dir); implies -w.
  --add-suffix=ADD_SUFFIX
                        Append this string to all output filenames. Requires
                        -n if non-empty.  ex: --add-suffix='3' will generate
                        .py3 files.
```

Korrekturen von 2to3 --list-fixes

```
Available transformations for the -f/--fix option:

  apply
  basestring
  buffer
  callable
  dict
  except
  exec
  execfile
  exitfunc
  filter
  funcattrs
  future
  getcwdu
  has_key
  idioms
  import
  imports
  imports2
  input
  intern
  isinstance
  itertools
  itertools_imports
  long
  map
  metaclass
  methodattrs
  ne
  next
  nonzero
  numliterals
  operator
```

```
  paren
  print
  raise
  raw_input
  reduce
  renames
  repr
  set_literal
  standarderror
  sys_exc
  throw
  tuple_params
  types
  unicode
  urllib
  ws_comma
  xrange
  xreadlines
  zip


Konvertierungstool "2to3" und Dateien
-------------------------------------
/usr/bin/2to3
/usr/bin/2to3-2.7
/usr/bin/2to3-3.2
/usr/share/doc/python2.7/examples/Tools/scripts/2to3
/usr/share/doc/python2.7/html/_sources/library/2to3.txt
/usr/share/doc/python2.7/html/library/2to3.html
/usr/share/doc/python3.2/examples/scripts/2to3
/usr/share/doc/python3.2/html/_sources/library/2to3.txt
/usr/share/doc/python3.2/html/library/2to3.html
/usr/share/man/man1/2to3-2.7.1.gz
/usr/share/man/man1/2to3-3.2.1.gz
/usr/share/man/man1/2to3.1.gz
```