

Änderungen Python 2.x --> 3.x

(C) 2014-2020 T.Birnthaler OSTC GmbH

Aktivieren von Eigenschaften neuerer Python-Versionen (z.B. Python 3) in älteren 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 müssen am Anfang des Quellcodes als 1. Anweisung stehen (da sie die Verhaltensweise des Python-Interpreters verändern).

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

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