```
Spezielle Bezeichner (Identifier)          (C) 2017-2021 T.Birnthaler OSTC GmbH
================================

Doku --> http://docs.python.org/dev/peps/pep-0008
         http://docs.python.org/3/reference/lexical_analysis.html#identifiers-and-keywords
         http://www.pep8.org

Folgende Namenskonventionen fÃ¼r Bezeichner (Identifier) gelten in Python:

+----------+-------------------------------------------------------------------+
| Name     | Bedeutung                                                         |
+----------+-------------------------------------------------------------------+
| __*__    | INTERNER Name (reserviert fÃ¼r Python selbst)                      |
| __*      | Privater Name in Klasse (mangled --> _CLASS__*)                   |
| _*       | Protected Name in Klasse (nur fÃ¼r Vererbung)                      |
|          | Interner Name in Modul (fÃ¼r modulinterne Zwecke)                  |
|          | (von "from MODULE import *" nicht importiert)                     |
|          | (Hilfsklasse, Hilfsfunktion, ...)                                 |
+----------+-------------------------------------------------------------------+
| *_       | SchlÃ¼sselwort als Bezeichner (z.B. "if_")                         |
| _        | Ergebnis der letzten Auswertung im interaktiven Interpreter       |
|          | TemporÃ¤re Variable in Schleife (for _ in ...)                     |
|          | "Wegwerfvariable" (syntaktisch notwendig, inhaltlich irrelevant)  |
|          | Internationalisierung (i18n, l10n, gettext)                       |
+----------+-------------------------------------------------------------------+
| self     | Objekt in normaler Methode einer Klasse                           |
| other    | 2. Objekt in normaler 2-wertiger Methode einer Klasse             |
| cls      | Klasse in Klassen-Methode einer Klasse                            |
| *args    | "Sammler" fÃ¼r Positions-Parameter in Funktion (--> Tupel)         |
| **kwargs | "Sammler" fÃ¼r Keyword-Parameter in Funktion (--> Dictionary)      |
+----------+-------------------------------------------------------+-----------+
| NAME     | Konstante       (GROSSB., Unterstrich, Ziffern)       | Subst/Adj |
|          | (Ausnahmen: True, False, None, math.inf, math.nan)    |           |
| Name     | Klassenname     (CamelCase)                           | Substantiv|
| Name     | Exceptionklasse (CamelCase)                           | Substantiv|
| name     | Modulname       (kleinb., KEIN Unterstrich, Ziffern)  | Substantiv|
| name     | Variablename    (kleinb., Unterstrich, Ziffern)       | Subst/Adj |
| name     | Funktionsname   (kleinb., Unterstrich, Ziffern)       | Verb      |
| name     | Objektname      (kleinb., Unterstrich, Ziffern)       | Subst/Adj |
+----------+-------------------------------------------------------+-----------+

Qualifizierte Namen werden durch "." getrennt (der "." wird beim Zugriff auf
das Dateisystem in den jeweiligen Verz.-Trenner "\" oder "/" umgesetzt).

  MODUL.VARIABLE               # Zugriff auf Variable in Modul
  MODUL.FUNKTION()             # Aufruf von Funktion in Modul
  PAKET.SUBMODUL.VARIABLE      # Zugriff auf Variable in Paket-Untermodul
  PAKET.SUBMODUL.FUNKTION()    # Aufruf von Funktion in Paket-Untermodul
  PFAD.ZU.MODUL.VARIABLE       # Zugriff auf Variable in Modul in Unterverzeichnis
  PFAD.ZU.MODUL.FUNKTION()     # Aufruf von Funktion in Modul in Unterverzeichnis

Komponenten von Klassen und Objekten werden ebenfalls durch "." im Namen
angesprochen:

  OBJEKT.ATTRIBUT              # Zugriff auf Attribut von Objekt
  OBJEKT.METHODE()             # Aufruf von Methode fÃ¼r Objekt
  KLASSE.ATTRIBUT              # Zugriff auf Klassen-Attribut
  KLASSE.METHODE()             # Aufruf von Klassen-Methode oder statischer Methode
  FUNKTION.ATTRIBUT            # Zugriff auf Funktions-Attribut

Spezielle Attribute
-------------------
Spezielle Attribute, werden teilweise von der built-in Funktion "dict()" nicht
aufgelistet, wohl aber von der Funktion "dir()".

+------------------+-------------------------------------------------------+
| Attribut         | Beschreibung                                          |
+------------------+-------------------------------------------------------+
| __doc__          | Dokumentationsstring ("Docstring")                    |
| __name__         | Name Klasse/Funktion/Methode/Descriptor/Generator-Instanz |
| __qualname__     | Qualifizierter Name Klasse/.../Generator-Instanz      |
| __module__       | Modul in dem Name definiert ist                       |
| __all__          | In __init__.py autom. zu lad. Subm. bei from ... import * |
+------------------+-------------------------------------------------------+
| __call__         | Codeobjekt einer Funktion                             |
| __defaults__     | Tuple mit Defaultwerten fÃ¼r Position-Parameter von Funk. |
| __kwdefaults__   | Dictionary mit Defaultwerten fÃ¼r Keyword-Parameter von F. |
| __annotations__  | Dictionary mit "Annotations" fÃ¼r Param. + RÃ¼ckgabewert |
| __closure__      | Bindungen freier Variablen an Werte                   |
+------------------+-------------------------------------------------------+
| __mro__          | Tupel der Basisklassen fÃ¼r MethodenauflÃ¶sung (linear) |
| mro()            | Liefert Ergebnis fÃ¼r __mro__ bei Klassen-Instanziierung |
```

__subclasses__()	Liste schwacher Referenzen zu direkten Unterklassen
__class__	Klasse zu der eine Instanz gehÃ¶rt
__bases__	Tupel der Basisklassen eines Klassenobjekts
__dict__	Speicher fÃ¼r dynamische Objektattribute (Name + Wert)
__slots__	Namen der statische Objektattribute
__weakref__	"Schwache" Referenzen
__globals__	Referenz zu Dictionary mit globalen Variablen
__self__	Objekt auf dem eine Methode arbeitet
__func__	Funktion die hinter einer Methode steckt (implementiert)

Spezielle Methoden
