Python Namensräume (Namespace)   (C) 2017-2025 T.Birnthaler OSTC GmbH
==============================

--> python-scope.txt   Scope/Sichtbarkeitsbereich
--> python-namespace.txt   Namespace/Namensraum
--> python-lifetime.txt   Lebensdauer/Lifetime/Existenz/Gültigkeitsbereich

Der Begriff "Namespace/Namensraum" bezieht sich auf die Bereiche in Python, in
denen Namen EINDEUTIG sind. In GETRENNTEN Namensräumen kann der gleiche Name
parallel verwendet werden, ohne dass es zu Überschneidungen kommt. Namensräume
sind GLOBAL, d.h. wenn ein Namensraum ansprechbar ist, dann sind auch alle
Namen darin ansprechbar durch die Notation NAMESPACE.NAME.

Folgende Namensräume (Namespaces) gibt es in Python:

Namensraum	Bedeutung
Built-in	In Python-Interpreter direkt eingebaut (z.B. len, sum, ...)
	Zugriff per NAME oder builtins.NAME (nach "import builtins")
Global	Im Skript (Hauptmodul) außerhalb von Funktion, Klasse, Objekt
	Zugriff per NAME oder __main__.NAME (nach "import __main__")
Modul	Zugriff per MODUL.NAME nach "import MODUL"
Funktion	Zugriff per FUNKTION.NAME nach "def FUNKTION(...): ..."
	oder "lambda ...: ..."
Klasse	Zugriff per KLASSE.NAME nach "class KLASSE: ..." oder Datentyp
Objekt	Zugriff per OBJEKT.NAME nach OBJEKT = KLASSE()

Zugriff auf die in Python direkt eingebauten ("built-in") Namen ist über das
Modul "builtins" grundsätzlich immer möglich:

```
  import builtins               # Namensraum "builtins" importieren
  erg = builtins.len("hallo")   # Statt len("hallo")
```

Zugriff auf die im aktuellen Skript (Hauptmodul) vorhandenen Namen ist über das
Modul "__main__" grundsätzlich immer möglich:

```
  name = "skript"        # Name im Namensraum des Skriptes
  import __main__        # Namensraum des aktuellen Skriptes importieren
  print(__main__.name)   # Statt print(name)
```

Im folgenden Beispiel wird der gleiche Name "len" parallel in verschiedenen
Namensräumen verwendet und referenziert jeweils verschiedene Objekte/Werte:

```
  import builtins        # Namensraum "builtins" importieren (Built-in)
  import __main__        # Namensraum "__main__" importieren (Skript)
  import math            # Namensraum "math"     importieren (Modul)
  def f(): pass          # Name "f"   (im Skript-   Namensraum "__main__")
  class C(): pass        # Name "C"   (im Skript-   Namensraum "__main__")
  o = C()                # Name "o"   (im Skript-   Namensraum "__main__")
  len = "global"         # Name "len" (im Skript-   Namensraum "__main__")
  math.len = "modul"     # Name "len" (im Modul-    Namensraum "math")
  f.len = "funktion"     # Name "len" (im Funktions-Namensraum "f")
  C.len = "klasse"       # Name "len" (im Klassen-  Namensraum "C")
  o.len = "objekt"       # Name "len" (im Objekt-   Namensraum "o")

  for name in (builtins.len, __main__.len, len, math.len, f.len, C.len, o.len):
      print(id(name), name, type(name))

  # --> 4520416080 <built-in function len> <class 'builtin_function_or_method'>
  # --> 4521365296 global <class 'str'>
  # --> 4521365296 global <class 'str'>
  # --> 4522715952 modul <class 'str'>
  # --> 4521061872 funktion <class 'str'>
```

```
# --> 4522188592 klasse <class 'str'>
# --> 4522214384 objekt <class 'str'>
```