"What's new" in Python-Versionen (C) 2016-2025 T.Birnthaler OSTC GmbH
==========

Die in jeder Python-Version hinzugefügten Erweiterungen bzw. entfernten Eigenschaften ("What's new in Python x.y")

Doku --> http://docs.python.org/3/whatsnew/index.html

Ver	Jahr	Beschreibung	PEP
3.15	2026	Make UTF-8 mode default	686
		Subscriptable functions	718
		TypedDict with Typed Extra Items	728
		Type checking annotated metadata	746
		Annotating Type Forms	747
		Inline typed dictionaries	764
		Annotating Read-Only Attributes	767
		Reimagining native threads	788
		imath – module for integer-specific mathematics functions	791
3.14	2025	Deferred Evaluation of Annotations Using Descriptors	649
		Implementing PEP 649	749
		Template Strings (Tag strings for domain-specific lang.)	750
		Safe external debugger interface for CPython	768
		Python Configuration C API	741
		C API to import-export Python integers	757
		Allow except and except* expressions without parentheses	758
		Disallow return/break/continue that exit a finally block	765
		Adding Zstandard to the standard library	784
		Static description file for build details of Py. install.	739
		Add Py_COMPAT_API_VERSION to the Python C API	743
		Type checking Annotated metadata	746
		Annotationg Type Forms	747
		A file format to record Python dep. for install.repr.lity	
- Python lock file format to specify dependencies			
		Add PyUnicode_Export() and PyUnicode_Import() C functions	756
		Deprecating PGP signatures for CPython artifacts	761
		Preventing task-cancel. bugs by limit. yield in async gen.	789
		Zero overhead external debugger interface for CPython	768
		Multiple interpreters in the stdlib (subinterpreters)	734
		Free-threaded Python is officially supported	779
		TypedDict with Typed Extra Items	728
		Shorthand syntax for keyword arguments at invocation	736
3.13	2024	Consistent views of namespaces	667
		Type Defaults for Type Parameters	696
		Marking deprecations using the type system	702
		Making the Global Interpreter Lock Optional in CPython	703
		TypedDict: Read-only items	705
		Adding iOS as a supported platform	730
		Multiple Interpreters in the Stdlib	734
		C API to format a type fully qualified name	737
		Adding Android as a supported platform	738
		Narrowing types with TypeIs	742
		JIT Compilation	744
		REPL-acing the default REPL	762
		Emscripten Support	776
3.12	2023	Low impact monitoring for CPython	669
		General purpose template literal strings	501
		Python local packages directory	583
		Type Aliases	
		Syntax for late-bound function argument defaults	671
		Allow parentheses in assert statements	679

		Cross-compiling Python packages	720
		Immortal Objects, Using a Fixed Refcount	683
		A per-interpreter GIL	684
		Making the buffer protocol accessible in Python	688
		Using TypedDict for more precise **kwargs typing	692
		Type Parameter Syntax and the type statement	695
		Type Defaults for Type Parameters statement	695
		Override Decorator for Static Typing	698
		Syntactic formalization of F-Strings	701
		Inlined Comprehensions	709
		Remove wstr from Unicode objects in Python's C API	623
		Remove distuils package	632
		Isolating modules in the standard library	687
3.11	2022	Include Fine-Grained Error Locations in Tracebacks	657
		Column information for code objects	...
		Zero-cost exceptions	...
		Exception Groups and except*	654
		Marking indiv. TypedDict items as required or pot. missing	655
		Self Type	673
		Variadic Generics	646
		Arbitrary literal string type	675
		Library lib2to3 and 2to3 tool deprecated	617
		tomllib: Support for Parsing TOML in the Standard Library	680
		Data Class Transforms	681
		10-60% faster than 3.10 (1.22x speedup on standard benchmark suite) Faster Cpython Project)	
		Regex: Atomic grouping (?>...)	
		Regex: Possessive quantifiers *+, ++, ?+, {m,n}+	
3.10	2021	Allow writing union types as X | Y	604
		Parameter specification variables	612
		Explicit type aliases	613
		Add optional length-checking to zip	618
		Parenthesized context managers officially allowed	
		Precise line numbers for debugging and other tools	626
		Deprecate module distutils	632
		Structural pattern matching: Specification	634
		Structural pattern matching: Motivation and Rationale	635
		Structural pattern matching: Tutorial	636
		Add optional EncodingWarning	597
3.9	2020	Union operator | for dict	584
		Type hinting generics in standard collections	585
		Relaxed grammar restrictions on decorators	614
		String methods to remove prefixes and suffixes	616
		Flexible function and variable annotations	593
		CPython now uses a new parser based on PEG	617
		range/tuple/set/frozenset/list/dict faster vectorcall	590
		Garbage collection does not block on resurrected objects	
		IANA time zone database in new module "zoneinfo"	615
		Topological graph sort offered in new module "graphlib"	
		CPython adopts an annual release cycle	602
3.8	2019	Type annotation/Hinting	
		Assignment Expression := ("Walrus" Operator)	572
		Positional only parameters (a, b, /, c, d, *, d, e)	
		PYTHONCACHEPREFIX (Option -X)	
		Formatierung F"{EXPR=}" --> "EXPR=VALUE-OF-EXPR"	
3.7	2018	PYTHONCOERCELOCALE	
		Data Class @dataclass	557
		Neue C-API für Thread-lokalen Speicher in CPython	
		Module "time" kennt Nanosekunden	
		Neuer UTF-8 Modus "surrogateescape" (für POSIX locale)	
		(-X utf8, PYTHONUTF8)	

		Neuer "Development mode" (-X dev, PYTHONDEVMODE)	
		Hash-basierte "*.pyc"-Checks (statt Zeitstempel-basiert)	
		Funktionsdef/Aufruf mit mehr als 255 Parametern/Argumenten	
		Viele Standard-Module erweitert	
		dict bewahrt Schlüssel-Reihenfolge (jede Implementierung)	
		Schlüsselworte "async" und "await"	
3.6	2016	Literal "f" für formatierte Strings	
		Syntax für Annotation von Variablen (Funktionen	
		Asynchrone Generatoren und Comprehensions	
		CPython-dict-Implementierung bewahrt Schlüssel-Reihenfolge	
3.5	2015	Operator "@" für Matrizen-Multiplikation	
		Koroutinen mit "async" und "await"	
		Type hints (Modul "typing")	484
3.4	2014	Umfassende PIP-Integration	
		Modul "enum" (Aufzählungs-Datentyp)	
		"asyncio"-Bibliothek	
3.3	2012	Syntax "yield from" (Generator Delegation)	
		Syntax u"STRING" wieder erlaubt (Unicode explizit)	
		Virtual Environments "venv"	405
		Implicit Namespace Packages	420
		Module: faulthandler ipaddress lzma unittest.mock venv	
3.2	2011	"*.pyc"-Kompilate in Unterverz. "__pycache__" sammeln	
		Viele Standard-Module erweitert	
3.1	2009	Datentyp "OrderedDict" im Modul "collections"	
		Methode "bit_length" des Datentyps "int"	
		Autom. Numerierung von Platzhaltern in String-Formatierung	
		Neuer Algorithmus zur internen Repr. von "float"-Werten	
		atexit	
3.0	2008	Änderungen die nicht auch in Version 2.6 enthalten sind:	
		Viele Funktionen/Methoden geben Iterator statt Liste zurück	
		Datentyp "long" entfernt	
		Datentyp "bytes" für Binärdaten	
		Datentyp "str" speichert ausschließlich Unicode-Strings	
		Dictionary-Comprehension	
		Reine Schlüsselwort-Parameter	
		Anweisung "nonlocal"	
		Literale für Datentyp "set"	
		Viele Standard-Module entfernt, umbenannt, verändert	
		Annotationen	
2.7	2010	Letzte Version von Python 2 (enthält Neuerungen von 3.1/2	
		Dictionary und Set-Comprehension	
		Modul "argparse"	
2.6	2008	Zeitgleich zu 3.0 entwickelt (enthält Neuerungen, die	
		die Abwärtskompatibilität nicht beeinträchtigen)	
		Kontext-Manager "with"	
		Paket "multiprocessing"	
		Neue Syntax zur String-Formatierung ("...".format(...))	
		Funktion "print" (statt Anweisung)	
		Oktale und binäre Zahlen-Literale	
2.5	2006	Conditional Expression	
		Relative "import"-Anweisung	
		Vereinheitlichung "try-except-finally"-Anweisung	
		(vorher "except" und "finally" nicht gemeinsam verwendbar)	
2.4	2004	Generator Expressions	
		Dekoratoren für Funktionen und Klassen	

		Modul "decimal"	
2.3	2003	Datentyp "set" (erst im Modul "sets", ab 2.4 eingebaut) Generatorfunktionen Datentyp "bool"	
2.2	2001	Verschachtelte Namensräume New-Style Klasse erlaubt Erben von built-in Datentypen Mehrfachvererbung Properties Iterator-Konzept Statische Methoden + Klassenmethoden Automatische Konvertierung Datentypen "int" und "long"	
2.1	2001	Modul "__future__" Vergleichsoperationen einzeln überladbar "import" auch unter Windows/macOS case-sensitiv	
2.0	2000	Datentyp "unicode" für Unicode-Strings List Comprehension Erweiterte Zuweisung (augmented assignment +=, ...) Garbage Collection für zyklische Referenzen	
1.0	1994	Veröffentlichung 1. Version	