De sus orígenes en 1989 a Python 3.14 (2025)

Cronología del Desarrollo de Python

Orígenes y Primeras Versiones

Finales de 1980 1989

- Guido van Rossum crea Python en CWI (Países Bajos).
- Inspirado en ABC, con enfoque en legibilidad y manejo de excepciones.
- Incluye clases, funciones, list, dict, str y sistema de módulos.

1991: Python 0.9.0/0.9.1

- Primera publicación pública en alt.sources.
- Incluye manejo de excepciones similar a Modula-3.
- Guido lidera el desarrollo inicial.

1994: Python 1.0

- Introduce lambda, map, filter, reduce.
- Nace comp.lang.python como foro principal.

1995-1996: Python 1.2 1.4

- Última versión en CWI, Van Rossum se traslada a CNRI (EE.UU.).
- Proyecto CP4E financiado por DARPA.
- Se añaden argumentos por palabra clave y soporte para números complejos.

2000: Python 1.6 y 2.0

- Python 1.6 con licencia CNRI.
- Python 2.0 introduce list comprehensions, GC con ciclos y Unicode.
- Desarrollo más abierto y comunitario.

2001: Python 2.2

- Unificación de tipos y clases.
- Generadores (PEP 255).
- Tim Peters contribuye con Zen of Python y Timsort.

2003: PEPs clave

- PEP 308: expresiones condicionales (Hettinger).
- PEP 333: WSGI (Phillip J. Eby).
- PEP 327: Decimal (Facundo Batista).

2006: Python 2.5

- Introduce 'with' para gestores de contexto.

2008: Python 2.6 y 3.0

- 2.6: incluye características de 3.0 y warnings.
- 3.0: revisión mayor incompatible.
- Print como función, str/unicode unificados, anotaciones de funciones.

2010: Python 2.7

- Última de la serie 2.x.
- Fin de soporte en 2020.

2014: Python 3.4

- Introduce asyncio y pathlib.

2015-2016: Python 3.5 y 3.6

- 3.5: hints de tipos, async/await.
- 3.6: f-strings, underscores en literales.

2018: Python 3.7

- Data classes, contextvars.
- Van Rossum renuncia como BDFL.

2019-2020: Python 3.8 y 3.9

- 3.8: Operador walrus (:=), parámetros solo posicionales.
- 3.9: Operadores | y |= para diccionarios.

2021-2023: Python 3.10 a 3.12

- 3.10: Pattern matching.
- 3.11: Mejoras en rendimiento y errores más claros.
- 3.12: f-strings más flexibles y mejoras en mensajes de error.

2024-2025: Python 3.13 y 3.14

- 3.13 (2024): optimizaciones adicionales.
- 3.14 (2025, preview): continúa el ciclo anual (PEP 602).

Concepto de BDFL

BDFL en Python

- Benevolent Dictator for Life: líder con última palabra en decisiones.
- Guido van Rossum fue BDFL hasta 2018.
- Decidía sobre aprobación/rechazo de PEPs.
- Tras su retiro, se adopta un Steering Council de 5 miembros.

Transición de Python 2.x a 3.x

Cronología de la Coexistencia

- 2006: PEP 3000 planifica Python 3.
- 2008: Python 3.0 lanzado, con incompatibilidades intencionales.
- 2009-2010: Python 2.7 lanzado en paralelo con 3.1.
- 2011-2019: Adopción lenta, uso de librerías como six y future.
- 2020: Fin de soporte oficial para Python 2.
- 2021-2025: Consolidación de Python 3 como estándar.

Problemas y Lecciones

- Incompatibilidad hacia atrás y adopción lenta.
- Herramientas de migración (2to3, __future__).
- Costos altos en la industria.
- Comunidad dividida pero finalmente unificada en Python 3.
- Legado: Python 3 fortalecido y estándar global.