

- i. **False** – Valor booleano, resultado de operaciones de comparación u operaciones lógicas en Python
- ii. **None** – Representa a un valor nulo
- iii. **True** – Valor booleano, igual que false, resultado de operaciones de comparación u operaciones lógicas en Python
- iv. **\_\_peg\_parser\_\_** – Llamado huevo de pascua, relacionado con el lanzamiento del nuevo analizador PEG no está definido aún.
- v. **And** – Operador lógico
- vi. **As** – Se utiliza para crear un alias al importar un módulo.
- vii. **Assert** – Se utiliza con fines de depuración
- viii. **Async** – Proporcionada por la biblioteca 'asyncio' en Python. Se utiliza para escribir código concurrente en Python
- ix. **Await** – Proporcionada por la biblioteca 'asyncio' en Python. Se utiliza para escribir código concurrente en Python
- x. **Break** – Se utiliza en el interior de los bucles for y while para alterar su comportamiento normal
- xi. **Class** – Se usa para definir una nueva clase definida por el usuario
- xii. **Continue** – Se utiliza en el interior de los bucles for y while para alterar su comportamiento normal
- xiii. **Def** – se usa para definir una función definida por el usuario
- xiv. **Del** – Para eliminar un objeto
- xv. **Elif** – Se usa en declaraciones condicionales, igual 'else' e 'if'
- xvi. **Else** – Se usa en declaraciones condicionales, igual 'elif' e 'if'
- xvii. **Except** – Se usa para crear excepciones, qué hacer cuando ocurre una excepción, igual que 'raise' y 'try'
- xviii. **Finally** – Su uso garantiza que el bloque de código dentro de él se ejecute incluso si hay una excepción no controlada
- xix. **For** – Utilizado para hacer bucles. Generalmente lo usamos cuando sabemos la cantidad de veces que queremos que se ejecute ese bucle
- xx. **From** – Para importar partes específicas de un módulo
- xxi. **Global** – Para declarar una variable global.
- xxii. **If** – Se usa en declaraciones condicionales, igual 'else' y 'elif'
- xxiii. **Import** – Para importar un módulo
- xxiv. **In** – Para comprobar si un valor está presente en una lista, tupla, etc. Devuelve True si el valor está presente, de lo contrario devuelve False
- xxv. **Is** – Se usa para probar si las dos variables se refieren al mismo objeto. Devuelve True si los objetos son idénticos y False si no
- xxvi. **Lambda** – Para crear una función anónima
- xxvii. **Nonlocal** – Para declarar una variable no local

- xxviii. **Not** – Operador lógico
- xxix. **Or** – Operador lógico
- xxx. **Pass** – Es una declaración nula en Python. No pasa nada cuando se ejecuta. Se utiliza como marcador de posición.
- xxxi. **Raise** – Se usa para crear excepciones, qué hacer cuando ocurre una excepción, igual que 'except' y 'try'
- xxxii. **Return** – Se usa dentro de una función para salir y devolver un valor.
- xxxiii. **Try** – Se usa para crear excepciones, qué hacer cuando ocurre una excepción, igual que 'raise' y 'except'
- xxxiv. **While** – Se usa para realizar bucles.
- xxxv. **With** – Se usa para simplificar el manejo de excepciones
- xxxvi. **Yield** – Se usa dentro de una función al igual que 'return', salvo que 'yield' devuelve un generador.