Python keywords

Creating a list of Python keywords along with their synopses in WikiText format involves detailing each keyword's purpose and usage in a structured manner, appropriate for a wiki or documentation page. Here's how you can structure and describe Python's keywords:

Python Keywords

Contents

Python keywords are reserved words that cannot be used as identifiers (e.g., variable names, function names, class names). They are used to define the syntax and structure of the Python language. Each keyword has a specific meaning and indicates the Python interpreter to perform a certain operation.

Objects

def
: Used to define a user-defined function.

del
: Deletes objects.

class
: Used to define a new user-defined class.

pass
: A null statement, a placeholder for future code.

return
: Exits a function or method and returns a value.

yield
: Used inside a function like a return statement but returns a generator.

lambda
: Creates an anonymous function.

Logic

if
: Used to create a conditional statement.

elif
: Used in conditional statements, same as "else if".

else
: Used in conditional statements as a catch-all for anything not caught by the preceding conditions.

and
: Used to combine two boolean values and return True if both operands are true.

not
: Logical NOT, used to invert the truth value.

or
Logical OR, used to combine boolean values and return True if any operand is true.

None
Represents the absence of a value or a null value.

True
Boolean value representing the true condition.

False
Boolean value representing the false condition.

is
Tests object identity.

in
Checks if a value is present in a sequence (list, tuple, string) or used to iterate through a sequence in a for loop.

Loops

for
Used to create a for loop.

while
Used to create a while loop.

break
Terminates the current loop and resumes execution at the next statement, only to be used within loops.

continue
Skips the rest of the code inside a loop for the current iteration and moves to the next iteration of the loop.

Error

try
Specifies exception handlers and/or cleanup code for a group of statements.

except
Used in try-except blocks to catch exceptions.

finally
Used in try-except blocks to define a block of code to be executed no matter if an exception is raised or not.

assert
Used for debugging purposes to test if a condition is true. If not, it raises an AssertionError.

raise
Used to raise an exception.

Imports

from

Used in import statements to specify specific parts of a module to import.

import
Used to import modules.

as
Used in `with` statements to alias a resource, or in `import` statements to alias a module.

Miscellaneous

with
Used to wrap the execution of a block of code with methods defined by a context manager.

global
Declares that a variable inside a function is global.

nonlocal
Declares that a variable inside a nested function is not local to it.

async
define functions as coroutines

await
Suspend the execution of coroutine on an awaitable object.

This summary provides a concise overview of the reserved keywords in Python and their primary uses within the language.

This format not only lists each keyword but also provides a brief explanation of its purpose, aligning with what you might find on a wiki page dedicated to Python programming. Python's keywords play a crucial role in defining the structure and syntax of the language, and understanding their use is essential for effective Python programming.
