Statements and control flow

Python's statements include (among others):

A: The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).

B: The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.

B: The while statement, which executes a block of code as long as its condition is true.

A: The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.

A: The raise statement, used to raise a specified exception or re-raise a caught exception.

B: The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.

A: The def statement, which defines a function or method.

A: The with statement, from Python 2.5 released in September 2006,[67] which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior and replaces a common try/finally idiom.[68]

A: The break statement, exits from the loop.

B: The continue statement, skips this iteration and continues with the next item.

B: The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.

A: The assert statement, used during debugging to check for conditions that ought to apply.

B: The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.

B: The import statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: import <module name> [as <alias>] or from <module name> import \* or from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>], ....

B: The print statement was changed to the print() function in Python 3.