Syntax	Example	Description
Variables	`x = 5`	Assign a value to a variable

Conditionals	`if x == 5:`
	`  print("x is 5")`
		Execute code only if a condition is met

Loops	`for i in range(5):`
	`  print(i)`
		Execute code repeatedly

Functions	`def my_func(x, y):`
	`  return x + y`
		Reusable block of code that performs a specific task

Classes	`class MyClass:`
	`  def __init__(self, x):`
	`    self.x = x`
	`  def my_method(self):`
	`    print(self.x)`
		Blueprint for creating objects with specific attributes and methods

Imports	`import math`
	`print(math.sqrt(4))`
		Use code from another module or package

Exception handling	`try:`
	`  x = 1 / 0`
	`except ZeroDivisionError:`
	`  print("Cannot divide by zero")`
		Handle errors that might occur during program execution

Boolean operators	`and`, `or`, `not`	Operators that operate on boolean values
Math operators	`+`, `-`, `*`, `/`, `//`, `%`, `**`	Operators that perform mathematical operations
Comparison operators	`==`, `!=`, `<`, `>`, `<=`, `>=`	Operators that compare values
Comprehensions	`[i**2 for i in range(5)]`	Concise syntax for creating lists, dictionaries, and sets