

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit
(AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> def testFunction(a, b, c, d):
...     print(a, b, c, d)
...
...
>>> testFunction(10, 20, 30, 40)
10 20 30 40
>>> testFunction(False, True, False, True)
False True False True
>>> testFunction(10, 20, 30, 40)
10 20 30 40
>>> testFunction(a=10, b=20, c=30, d=40)
10 20 30 40
>>> # Keyword syntax & Non-keyword syntax
>>> # With keyword syntax you need not follow the order in
>>> # formal parameter list
>>> testFunction(d=40, c=30, b=20, a=10)
10 20 30 40
>>> # Mix nonkeyword style and keyword style syntax
>>> testFunction(10, 20, c=30, d=40)
10 20 30 40
>>> testFunction(10, d=40, b=20, c=30)
10 20 30 40
>>> # GOLDEN RULE: While writing actual parameter list, once you start wring actual
parameter
>>> # using keyword syntax, you must keep writing the parameters in keyword syntax
until the end
>>> # You CANNOT FALL BACK to NON-KEYWORD SYNTAX
>>> testFunction(10, b=20, c=30, d=40)
10 20 30 40
>>> testFunction(10, b=20, 30, 40)
SyntaxError: positional argument follows keyword argument
# KEYWORD ARGUMENT

def testFunction(a, b, c, d):
    print(a, b, c, d)

testFunction(10, 20, c=30, d=40)
10 20 30 40
testFunction(10, 20, 30, 40)
10 20 30 40
#-----
def testFunction(*args):
    print(args)
```

```
testFunction(10, 20, 30, 40)
(10, 20, 30, 40)
testFunction()
()
testFunction(10)
(10,)
testFunction(10, 20)
(10, 20)
t
testFunction(10, 20, 30)
(10, 20, 30)
def testFunction(**X):
    print(X)

testFunction(a=10, b=20, c=30)
{'a': 10, 'b': 20, 'c': 30}
```