

Online Python Compiler - online

onlinegdb.com/online_python_compiler

Language Python 3

main.py

```
1 # Exercise 1
2
3 def calculate(a, b=10, c=None):
4     if c is None:
5         print("Sum of the first two arguments:", a + b)
6     else:
7         print("Product of all three arguments:", a * b * c)
8
9 # Example usage:
10 calculate(5)
11 calculate(5, 2)
12 calculate(5, 2, 3)
13
14 # Exercise2
15
16 def filter_long_strings(strings):
17     return [s for s in strings if len(s) >= 5]
18
19 # Example usage:
20 strings = ["apple", "dog", "banana", "cat", "grape"]
21 print("Filtered strings (length >= 5):", filter_long_strings(strings))
22
23 # Exercise 3: Evaluate Mathematical Expression Using `eval()`
24
25 expression = "3 * 5 + 2"
26 result = eval(expression)
27 print("Result of expression '{}':".format(expression), result)
28
29 # Exercise 4: Filter Prime Numbers Using `filter()`
30
31 def is_prime(n):
32     if n < 2:
33         return False
34     for i in range(2, int(n ** 0.5) + 1):
35         if n % i == 0:
36             return False
37     return True
```

input

```
Sum of the first two arguments: 15
Sum of the first two arguments: 7
Product of all three arguments: 30
Filtered strings (length >= 5): ['apple', 'banana', 'grape']
Result of expression '3 * 5 + 2': 17
Prime numbers from the list: [2, 3, 5, 7]
Uppercase strings: ['APPLE', 'BANANA', 'CHERRY']

...Program finished with exit code 0
Press ENTER to exit console.
```

14:16 14-01-2025

Online Python Compiler - online ...
onlinegdb.com/online_python_compiler
Run Debug Stop Share Save {} Beautify Python 3
main.py
13
14 # Exercise2
15
16 def filter_long_strings(strings):
17 return [s for s in strings if len(s) >= 5]
18
19 # Example usage:
20 strings = ["apple", "dog", "banana", "cat", "grape"]
21 print("Filtered strings (length >= 5):", filter_long_strings(strings))
22
23 # Exercise 3: Evaluate Mathematical Expression Using `eval()`
24
25 expression = "3 * 5 + 2"
26 result = eval(expression)
27 print("Result of expression '{}':".format(expression), result)
28
29 # Exercise 4: Filter Prime Numbers Using `filter()`
30
31 def is_prime(n):
32 if n < 2:
33 return False
34 for i in range(2, int(n ** 0.5) + 1):
35 if n % i == 0:
36 return False
37 return True
38
39 numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
40 prime_numbers = list(filter(is_prime, numbers))
41 print("Prime numbers from the list:", prime_numbers)
42
43 # Exercise 5: Convert List of Strings to Uppercase Using `map()`
44 def to_uppercase(strings):
45 return list(map(str.upper, strings))
46
47 strings = ["apple", "banana", "cherry"]
48 uppercase_strings = to_uppercase(strings)
49 print("Uppercase strings:", uppercase_strings)
input
Sum of the first two arguments: 15
Sum of the first two arguments: 7
Product of all three arguments: 30
Filtered strings (length >= 5): ['apple', 'banana', 'grape']
Result of expression '3 * 5 + 2': 17
Prime numbers from the list: [2, 3, 5, 7]
Uppercase strings: ['APPLE', 'BANANA', 'CHERRY']
...Program finished with exit code 0
Press ENTER to exit console.
14:16 14-01-2025