

```

1  def factorfinder(num):
2      factors = []
3
4      for i in range(1, num + 1):
5          if num % i == 0:-
6              factors.append(i)
7              factors.append(num // i)
8
9      factors = list(set(factors))
10     factors.sort()
11
12     for i, factor in enumerate(factors):
13         factors[i] = str(factor)
14     print(', '.join(factors))
15
16 factorfinder(26)
--

```

Factor Finder

A number's factors are any two other numbers that, when multiplied with each other, produce the number. For example, $2 \times 13 = 26$, so 2 and 13 are factors of 26. Also, $1 \times 26 = 26$, so 1 and 26 are also factors of 26. Therefore, we say that 26 has four factors: 1, 2, 13, and 26.

TO DO:

- ☐ Copy this code and make sure it runs. If you get errors, use the error messages to fix your program.
- ☐ Use your program to find some factors:
 - The factors of 36:
 - The factors of 840:
 - The factors of 7237:
 - The factors of _____:
- ☐ There are several new Python functions and methods in this program. List at least three that are unfamiliar to you and a guess about what each does:

- ☐ Try to find the answers to the following questions. Experiment with some modifications to the code and rerun the program to see what effect the changes have.
- What happens if you delete or comment out `factors.append(i)` on line 6?
 - What happens if you delete or comment out `factors = list(set(factors))` on line 9? (Hint: Enter a square number such as 25 or 36 or 49.)
 - What happens if you delete or comment out `factors.sort()` on line 10?
 - What error message do you get if you change `factors = []` on line 2 to `factors = ''`?
 - What happens if you change `factors = []` on line 2 to `factors = [-42]`?
 - What error message do you get if you change `factors = []` on line 2 to `factors = ['hello']`?

CHALLENGE:

If a number only has two factors (1 and itself), we call that a prime number. Otherwise, we call it a composite number. Use the factor finder to discover some new prime numbers! (Hint: Prime numbers always end with an odd number that isn't 5.)

How could you use Python (including the program you already wrote) to find all the prime numbers between 1 and 1000?