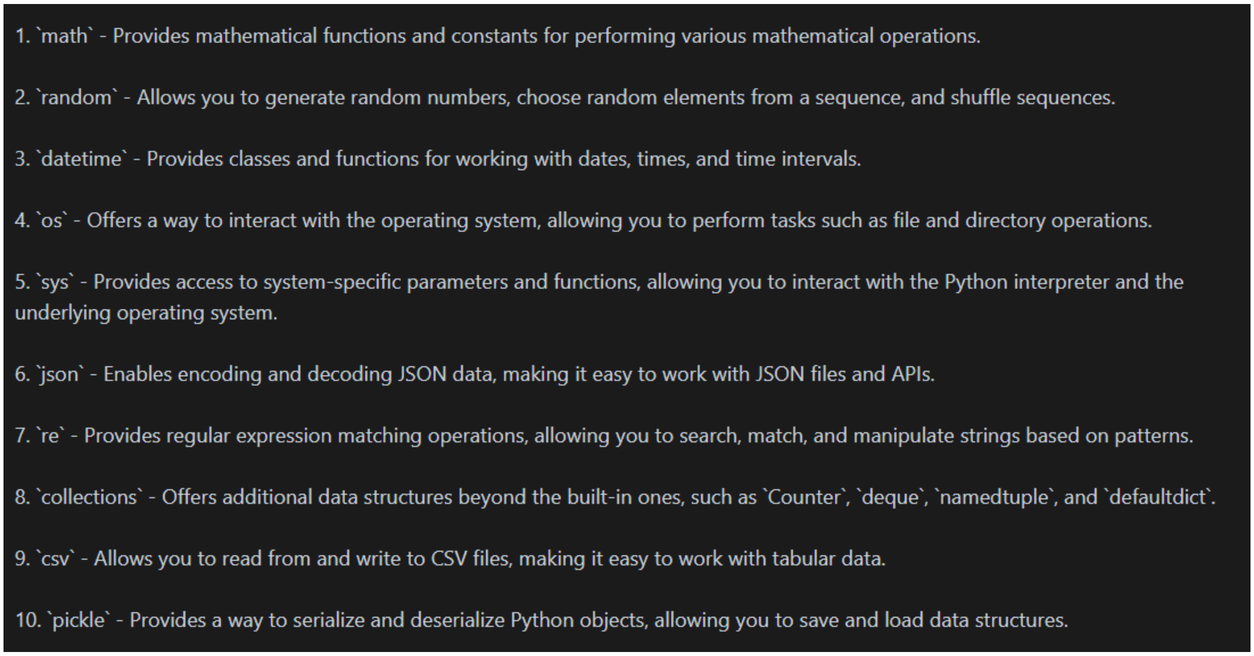


1. `math` - Provides mathematical functions and constants for performing various mathematical operations.

2. `random` - Allows you to generate random numbers, choose random elements from a sequence, and shuffle sequences.

3. `datetime` - Provides classes and functions for working with dates, times, and time intervals.

4. `os` - Offers a way to interact with the operating system, allowing you to perform tasks such as file and directory operations.

5. `sys` - Provides access to system-specific parameters and functions, allowing you to interact with the Python interpreter and the underlying operating system.

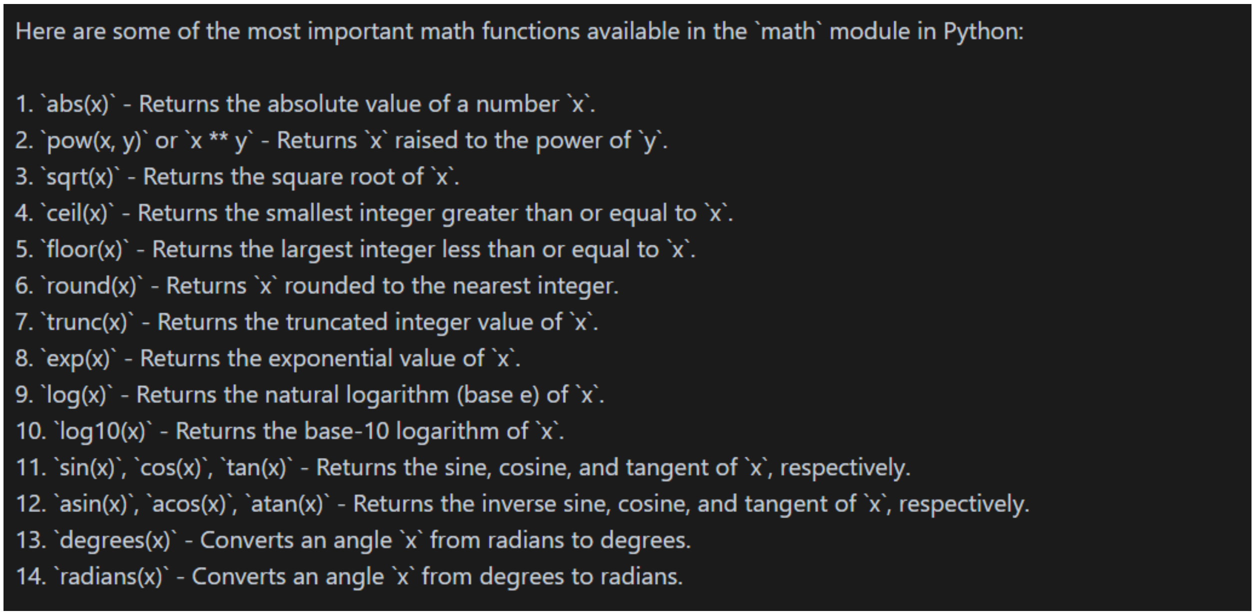
6. `json` - Enables encoding and decoding JSON data, making it easy to work with JSON files and APIs.

7. `re` - Provides regular expression matching operations, allowing you to search, match, and manipulate strings based on patterns.

8. `collections` - Offers additional data structures beyond the built-in ones, such as `Counter`, `deque`, `namedtuple`, and `defaultdict`.

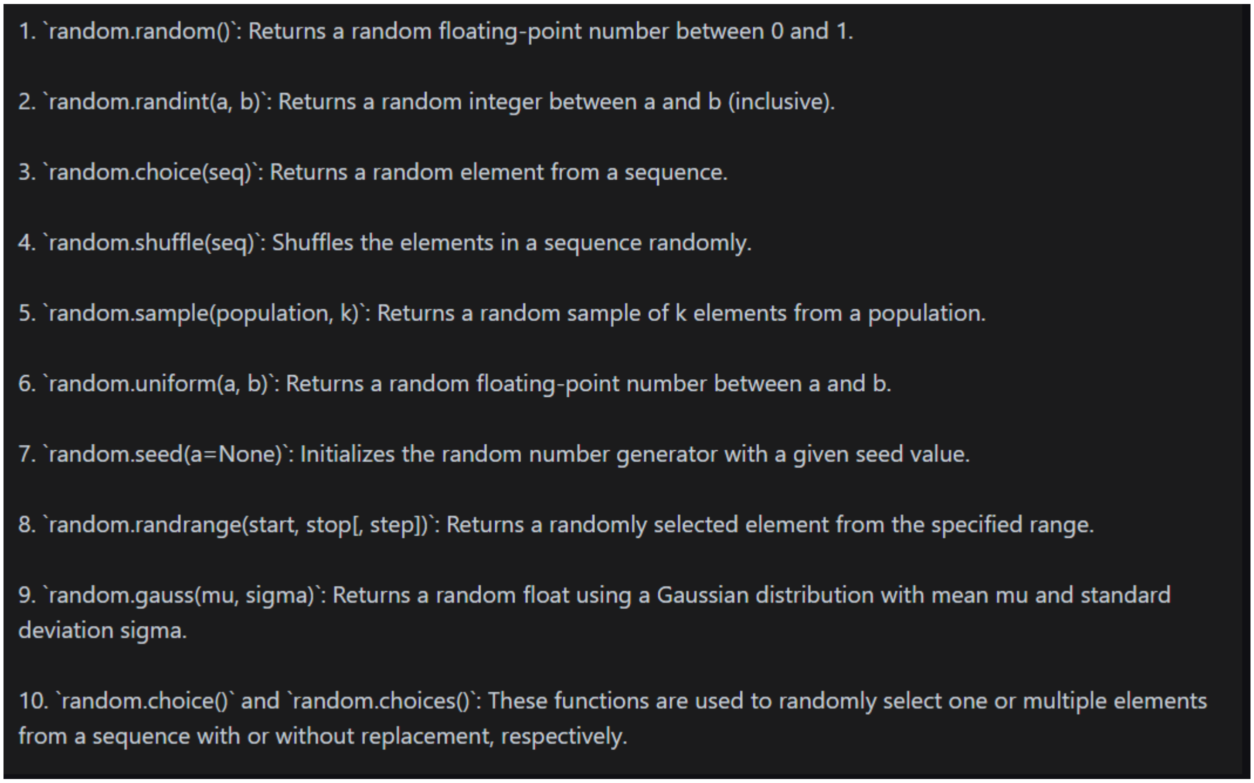
9. `csv` - Allows you to read from and write to CSV files, making it easy to work with tabular data.

10. `pickle` - Provides a way to serialize and deserialize Python objects, allowing you to save and load data structures.



Here are some of the most important math functions available in the `math` module in Python:

1. `abs(x)` - Returns the absolute value of a number `x`.
2. `pow(x, y)` or `x ** y` - Returns `x` raised to the power of `y`.
3. `sqrt(x)` - Returns the square root of `x`.
4. `ceil(x)` - Returns the smallest integer greater than or equal to `x`.
5. `floor(x)` - Returns the largest integer less than or equal to `x`.
6. `round(x)` - Returns `x` rounded to the nearest integer.
7. `trunc(x)` - Returns the truncated integer value of `x`.
8. `exp(x)` - Returns the exponential value of `x`.
9. `log(x)` - Returns the natural logarithm (base e) of `x`.
10. `log10(x)` - Returns the base-10 logarithm of `x`.
11. `sin(x)`, `cos(x)`, `tan(x)` - Returns the sine, cosine, and tangent of `x`, respectively.
12. `asin(x)`, `acos(x)`, `atan(x)` - Returns the inverse sine, cosine, and tangent of `x`, respectively.
13. `degrees(x)` - Converts an angle `x` from radians to degrees.
14. `radians(x)` - Converts an angle `x` from degrees to radians.



1. `random.random()`: Returns a random floating-point number between 0 and 1.

2. `random.randint(a, b)`: Returns a random integer between a and b (inclusive).

3. `random.choice(seq)`: Returns a random element from a sequence.

4. `random.shuffle(seq)`: Shuffles the elements in a sequence randomly.

5. `random.sample(population, k)`: Returns a random sample of k elements from a population.

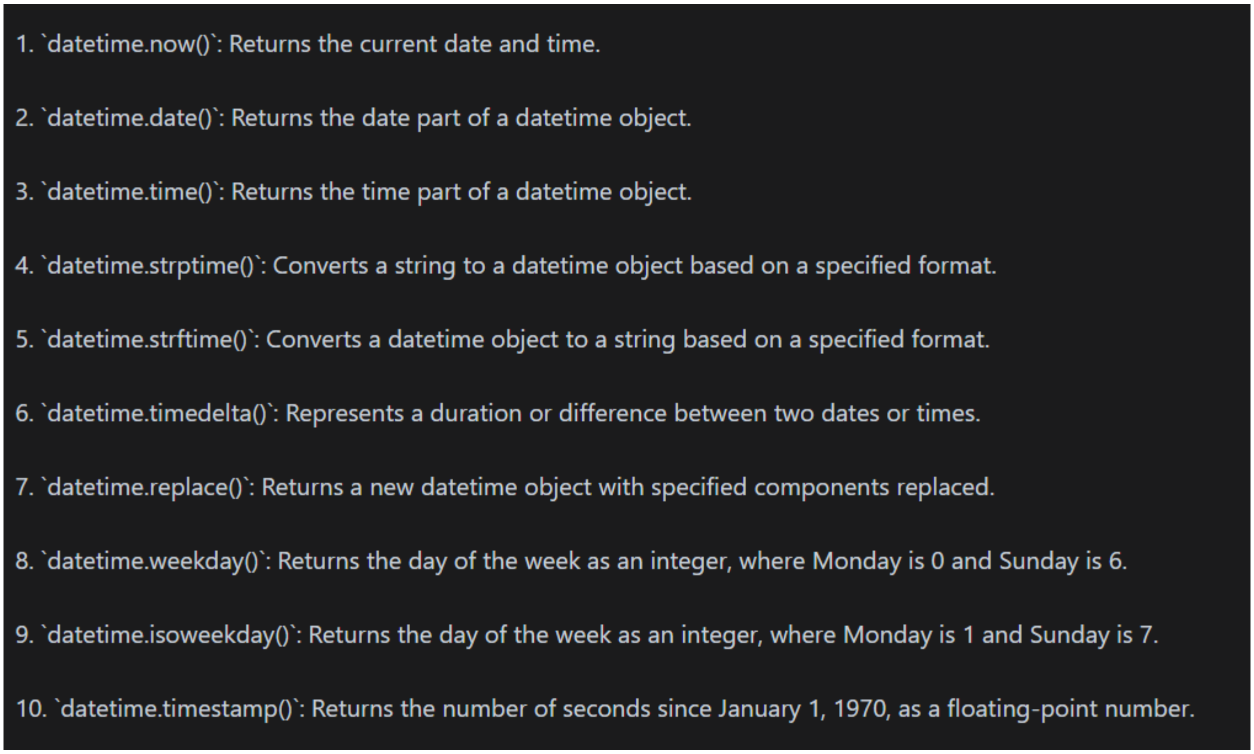
6. `random.uniform(a, b)`: Returns a random floating-point number between a and b.

7. `random.seed(a=None)`: Initializes the random number generator with a given seed value.

8. `random.randrange(start, stop[, step])`: Returns a randomly selected element from the specified range.

9. `random.gauss(mu, sigma)`: Returns a random float using a Gaussian distribution with mean mu and standard deviation sigma.

10. `random.choice()` and `random.choices()`: These functions are used to randomly select one or multiple elements from a sequence with or without replacement, respectively.



1. `datetime.now()`: Returns the current date and time.

2. `datetime.date()`: Returns the date part of a datetime object.

3. `datetime.time()`: Returns the time part of a datetime object.

4. `datetime.strptime()`: Converts a string to a datetime object based on a specified format.

5. `datetime.strftime()`: Converts a datetime object to a string based on a specified format.

6. `datetime.timedelta()`: Represents a duration or difference between two dates or times.

7. `datetime.replace()`: Returns a new datetime object with specified components replaced.

8. `datetime.weekday()`: Returns the day of the week as an integer, where Monday is 0 and Sunday is 6.

9. `datetime.isoweekday()`: Returns the day of the week as an integer, where Monday is 1 and Sunday is 7.

10. `datetime.timestamp()`: Returns the number of seconds since January 1, 1970, as a floating-point number.