

Generator exercises

- 1) Write a generator function that generates the first n squares of integers, starting from 0. For example, if $n=5$, the generator should produce the sequence 0, 1, 4, 9, 16.
- 2) Write a generator function that takes a list of integers as input, and yields the sum of the previous two integers in the list for each value in the list. For example, if the input list is [1, 2, 3, 4], the generator should produce the sequence 3, 5, 7.
- 3) Write a generator function that takes a string as input, and yields each word in the string in reverse order. For example, if the input string is "hello world", the generator should produce the sequence "olleh", "dlrow".
- 4) Write a generator function that generates all the prime numbers less than or equal to a given integer n . For example, if $n=10$, the generator should produce the sequence 2, 3, 5, 7.
- 5) Write a generator expression that generates a sequence of tuples (i, j) for all integers i and j such that $i < j$ and i is even and j is odd. For example, the generator should produce the sequence (0, 1), (0, 3), (2, 3), (2, 5), (4, 5), (4, 7),
- 6) Write a generator that produces the Fibonacci sequence in a lazy manner.
- 7) Write a generator function that takes a list of stock prices and yields the percentage change between each pair of consecutive prices. For example, if the input list is [10, 12, 8, 15, 11], the generator should produce the sequence 0.2, -0.333, 0.875, -0.267.
- 8) Write a generator function that generates the moving average of a list of stock prices for a given window size k . For example, if the input list is [10, 12, 8, 15, 11] and $k=3$, the generator should produce the sequence 10.0, 10.666, 11.666, 11.333.
- 9) Write a generator function that generates the cumulative return of a list of stock prices over time. For example, if the input list is [10, 12, 8, 15, 11], the generator should produce the sequence 0.0, 0.2, -0.2, 0.5, 0.1.
- 10) Write a generator function that takes a list of stock prices and yields the top k largest percentage price gains for any n -day period, where n is a parameter of the function. For example, if the input list is [10, 12, 8, 15, 11, 20, 18, 22] and $n=3$ and $k=2$, the generator should produce the sequence 0.667, 1.75, which represents the top two largest percentage gains for any 3-day period in the list ((8, 15, 11) and (11, 20, 18)).

11) Write a generator function that takes a list of playing cards and yields only the face cards (Jacks, Queens, and Kings) in the list. For example, if the input list is ["Ace", "Jack", "3", "Queen", "King", "10"], the generator should produce the sequence "Jack", "Queen", "King".

12) Write a generator function that generates all possible combinations of cards in a standard deck of playing cards, without repetition. For example, the generator should produce the sequence ("Ace", "Spades"), ("2", "Spades"), ..., ("King", "Diamonds").

13) Write a generator function that takes a list of playing cards and yields all possible pairs of cards in the list, without repetition. For example, if the input list is ["Ace", "Jack", "3", "Queen"], the generator should produce the sequence ("Ace", "Jack"), ("Ace", "3"), ("Ace", "Queen"), ("Jack", "3"), ("Jack", "Queen"), ("3", "Queen").

14) Write a generator function that generates all possible hands of a given size k from a list of playing cards, without repetition. For example, if the input list is ["Ace", "Jack", "3", "Queen", "King"] and k=2, the generator should produce the sequence ("Ace", "Jack"), ("Ace", "3"), ..., ("King", "Queen").

15) Write a generator function that connects to a web server and yields the HTML content of each page on the site, following links recursively to traverse the entire site.