Exercises

1) Write a function to calculate the area of a rectangle.
2) Write a function to calculate the area of a circle.
3) Write a function to calculate the perimeter of a rectangle.
4) Write a function to calculate the circumference of a circle.
5) Write a function to check if a number is even or odd.
6) Write a function to check if a string is a palindrome.
7) Write a function to check if a string is a valid email address.
8) Write a function to reverse a string.
9) Write a function to find the maximum value in a list.
10) Write a function to find the minimum value in a list.

Exercises and solution

1) Write a function to calculate the area of a rectangle.

```
def rectangle_area(length, width):
    """
    Calculates the area of a rectangle given its length and width.

    Args:
        length (float): The length of the rectangle.
        width (float): The width of the rectangle.

    Returns:
        float: The area of the rectangle.
    """
    return length * width
```

2) Write a function to calculate the area of a circle.

```
import math

def circle_area(radius):
    """
    Calculates the area of a circle given its radius.

    Args:
        radius (float): The radius of the circle.

    Returns:
        float: The area of the circle.
    """
    return math.pi * radius ** 2
```

3) Write a function to calculate the perimeter of a rectangle.

```
def rectangle_perimeter(length, width):
    """
    Calculates the perimeter of a rectangle given its length and width.

    Args:
        length (float): The length of the rectangle.
        width (float): The width of the rectangle.

    Returns:
        float: The perimeter of the rectangle.
    """
    return 2 * (length + width)
```

4) Write a function to calculate the circumference of a circle.

```
import math

def circle_circumference(radius):
    """
    Calculates the circumference of a circle given its radius.

    Args:
        radius (float): The radius of the circle.

    Returns:
        float: The circumference of the circle.
    """
    return 2 * math.pi * radius
```

5) Write a function to check if a number is even or odd.

```
def is_even(num):
    """
    Checks if a given number is even.
    Args:
        num (int): The number to check.

    Returns:
        bool: True if the number is even, False otherwise.
    """
    return num % 2 == 0
```

6) Write a function to check if a string is a palindrome.

```
def is_palindrome(s):
    """
    Checks if a given string is a palindrome.

    Args:
        s (str): The string to check.

    Returns:
        bool: True if the string is a palindrome, False otherwise.
    """
    return s == s[::-1]
```

7) Write a function to check if a string is a valid email address.

```
import re

def is_valid_email(email):
    """
    Returns True if the given string is a valid email address, False otherwise.
    """
    # Define regular expression pattern for matching email addresses
    pattern = r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"
    # Use the re.match function to see if the email matches the pattern
    match = re.match(pattern, email)

    # If a match is found, the email is valid, otherwise it is not
    return match is not None
```

In this function, we use the re module to define a regular expression pattern that matches valid email addresses. The pattern is then used with the re.match() function to see if the given email matches the pattern. If it does, the function returns True, indicating that the email is valid. If it does not match the pattern, the function returns False.

8) Write a function to reverse a string.

```
def reverse_string(s):
    """
    Reverses a given string.

    Args:
        s (str): The string to reverse.

    Returns:
        str: The reversed string.
    """
    return s[::-1]
```

9) Write a function to find the maximum value in a list.

```
def find_max(lst):
    """
    Finds the maximum value in a given list.

    Args:
        lst (list): The list to search.
    Returns:
        int or float: The maximum value in the list.
    """
    return max(lst)
```

10) Write a function to find the minimum value in a list.

```
def find_min(numbers):
    min_num = numbers[0] # set the first number in the list as the minimum
    for num in numbers:
        if num < min_num:
            min_num = num
    return min_num
```