Python Cheatsheet

Python Basics

- **`print()`** - prints the specified message to the console
- **`input()`** - prompts the user to enter input
- **`type()`** - returns the data type of a variable
- **`str()`** - converts the specified object into a string
- **`int()`** - converts the specified object into an integer
- **`float()`** - converts the specified object into a float

Control Structures

- **`if/else`** - performs conditional logic
- **`for`** - iterates over a sequence of values
- **`while`** - repeats a block of code as long as a certain condition is true

Functions

- **`def function_name(arguments):`** - defines a function with the specified name and arguments
- **`return`** - specifies the value that the function should return
- **`lambda arguments: expression`** - defines an anonymous function

Data Structures

- **`list`** - a collection of ordered and mutable elements
- **`tuple`** - a collection of ordered and immutable elements
- **`set`** - a collection of unordered and unique elements
- **`dict`** - a collection of unordered key-value pairs

File I/O

- **`open()`** - opens a file and returns a file object
- **`read()`** - reads the contents of a file

- **`write()`** - writes the specified string to a file
- **`close()`** - closes a file

Object-Oriented Programming

- **`class ClassName:`** - defines a class with the specified name
- **`__init__(self, arguments):`** - initializes a new instance of the class
- **`self`** - a reference to the current instance of the class
- **`def method_name(self, arguments):`** - defines a method of the class
- **`inheritance`** - the ability to create a new class that is a modified version of an existing class

Exception Handling

- **`try/except`** - handles exceptions that occur during the execution of a program
- **`raise`** - raises a specified exception
- **`finally`** - specifies code that should be executed regardless of whether an exception occurs

Modules

- **`import module_name`** - imports a module with the specified name
- **`from module_name import function_name`** - imports a specific function from a module
- **`pip`** - a package manager used to install and manage Python packages

Decorators

- **`@decorator_name`** - applies a specified decorator to a function
- **`@property`** - specifies that a method should be treated as a property of an object

Advanced Data Structures

- **`collections`** module - provides alternative data structures such as deque, OrderedDict, and Counter

- **`numpy`** library - provides support for arrays and numerical operations
- **`pandas`** library - provides support for data analysis and manipulation