Python Basics Cheat Sheet

The Right Way to Start

Michael Borck

Welcome to the Python Basics Cheat Sheet!

This cheat sheet is designed to help you get started with the basics of Python programming. Python is a high-level, interpreted programming language that is easy to learn and fun to use. With Python, you can create a wide range of programs, from simple scripts to complex applications.

What's in this cheat sheet?

This cheat sheet covers the basic syntax and features of Python, including:

- Variables and data types
- Control structures (if/else, for loops, while loops)
- Functions
- Import statements

How to use this cheat sheet:

- Use this cheat sheet as a reference guide to help you learn the basics of Python.
- Try out the examples and exercises to practice your skills.
- Use this cheat sheet to review and reinforce your understanding of Python basics.

What's next?

Once you've mastered the basics of Python, you can move on to more advanced topics, such as:

- **Fetching Data from APIs** (Session 2): Learn how to fetch data from APIs using Python and the `requests` library.

- **Parsing JSON Data** (Session 3): Learn how to parse JSON data using Python and the `json` library.
- **Building the Weather Dashboard** (Session 4): Use your new Python skills to build a basic weather dashboard.

Remember, practice makes perfect! The more you practice, the better you'll become at writing Python code.

Happy coding!

Variables

- Assign a value to a variable using the assignment operator (=)
- Use the `print()` function to print the value of a variable
- Use the `type()` function to check the data type of a variable

Example:

```
x = 5  # assign the value 5 to the variable x
print(x)  # print the value of x
print(type(x))  # print the data type of x
```

Data Types

- Integers (int): whole numbers, e.g. 1, 2, 3, etc.
- Floats (float): decimal numbers, e.g. 3.14, -0.5, etc.
- Strings (str): sequences of characters, e.g. "hello", 'hello', etc.
- Boolean (bool): true or false values
- Lists (list): ordered collections of values, e.g. [1, 2, 3], ["a", "b", "c"], etc.
- Tuples (tuple): ordered, immutable collections of values, e.g. (1, 2, 3), ("a", "b", "c"), etc.

Control Structures

- Conditional statements:
  - `if` statement: execute a block of code if a condition is true
  - `elif` statement: execute a block of code if a condition is true and the previous conditions are false
  - `else` statement: execute a block of code if none of the previous conditions are true
- Loops:

- `for` loop: iterate over a sequence of values
- `while` loop: execute a block of code while a condition is true

Example:

```
x = 5
if x > 10:
    print("x is greater than 10")
elif x == 5:
    print("x is equal to 5")
else:
    print("x is less than 5")
```

Functions

- Define a function using the `def` keyword
- Use the `return` statement to return a value from a function
- Call a function by its name, passing in arguments as needed

Example:

```
def greet(name):
    print("Hello, " + name + "!")
greet("John")  # call the function with the argument "John"
```

Import Statements

- Use the `import` statement to import a module or function from another file
- Use the `from` keyword to import specific functions or variables from a module

Example:

```
import math
print(math.pi)  # print the value of pi
from math import sin
print(sin(3.14))  # print the sine of 3.14
```

This is just a basic cheat sheet, and there is much more to learn about Python. But this should give you a good starting point for your Python jopurney!