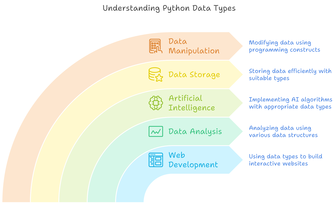
Python Data Types

In this document, we will explore the various data types available in Python, a versatile programming language widely used for web development, data analysis, artificial intelligence, and more. Understanding these data types is crucial for effective programming, as they dictate how data is stored, manipulated, and interacted with in your code.



1. Numeric Types

1.1 Integers

Integers are whole numbers, both positive and negative, without any decimal point. In Python, you can define an integer simply by writing a number.

```
x = 10
y = -5
```

1.2 Floats

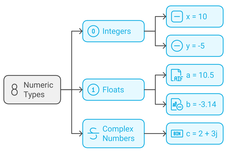
Floats are numbers that contain a decimal point. They can represent both whole numbers and fractions.

```
a = 10.5
b = -3.14
```

1.3 Complex Numbers

Complex numbers are represented by a real part and an imaginary part, denoted by **j** or **J**.

```
c = 2 + 3j
```



2. Sequence Types

2.1 Strings

Strings are sequences of characters enclosed in single, double, or triple quotes. They are immutable, meaning they cannot be changed after creation.

```
s = "Hello, World!"
```

2.2 Lists

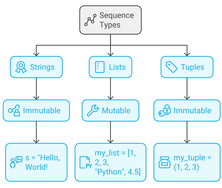
Lists are ordered collections of items that can be of different types. They are mutable, allowing changes to their content.

```
my_list = [1, 2, 3, "Python", 4.5]
```

2.3 Tuples

Tuples are similar to lists but are immutable. Once created, their content cannot be changed.

```
my_tuple = (1, 2, 3)
```



3. Mapping Type

3.1 Dictionaries

Dictionaries are collections of key-value pairs. They are unordered and mutable, allowing for dynamic changes.

```
my_dict = {"name": "Alice", "age": 25}
```

4. Set Types

4.1 Sets

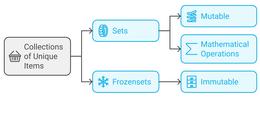
Sets are unordered collections of unique items. They are mutable and can be used to perform mathematical set operations.

```
my_set = {1, 2, 3, 4}
```

4.2 Frozensets

Frozensets are immutable versions of sets. Once created, their content cannot be changed.

```
my_frozenset = frozenset({1, 2, 3})
```

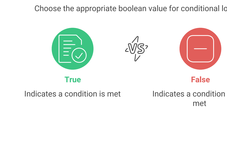


5. Boolean Type

5.1 Booleans

Booleans represent one of two values: **True** or **False**. They are often used in conditional statements.

```
is_active = True
is_logged_in = False
```



Conclusion

Understanding Python's data types is fundamental for any programmer looking to write efficient and effective code. Each data type serves a specific purpose and can be utilized in various ways to solve problems and manage data effectively. By mastering these data types, you will enhance your programming skills and be better equipped to tackle complex tasks in Python.