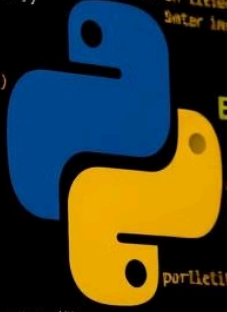


# Mastering Python Data Types

Welcome to the world of Python Data Types. This presentation will guide you through the basics of data types in Python, helping you understand how to work with different types of data in your programs.

```
1 ddd(ants:
2   the tisttands:
3   foerianan; #nceferLast-COM_tiorvet)
4   ver inneetensi; #NOV/FT:101)
5   tr= innegetess:
6   7: Antides: Inegalantes the Piccip/Feativen 200):
7   77 ettaction filats Thatinntefers And Conlietats, Innesinet This for 11 fionest
8   Yrch:
9   in ranglaies:
10  inf (aroltinction0, CHEIT711ATPY: CAROPT)
11  ans
12  TS. inststiogionan. 'PDT7-bloges' 0]
13  CMchesintyi, upcephert ( FLOATS)
14  ar denectual
15  ds: racelity <--thati ressilitemfyerst =1))
16  tuntegiy == corinestion, rauttont
17  corvisatemcet)
18  11: Forltgater: INA7.5ffrtadd)
19  11: Pungdsert: IN(80-70V).jec fc:
20  Tocking == L885-1082, lingsianty
21  js chactert: FLo00, 107C-15 imngfane:
22  dhet ilautfier: Flaats: Vanld, DNvagiy
23  11: langtalon: ANOTion. "Cardata)
24
25  js: fandtent, sestatts (interve)
26  Candter: Vails Mapaca:
27  dhet ilactert: inolles, fFich Fawtiner -warta/Flagersssention, innittles itaats, with fionest.
28  1: longttaner: Thecasig iorflas your pythom 'oytanmert. Thek eregion from paitillan
29  js: Inceterts: Puntlles, RS0] ==> Tonpe(ant: stand laite 10-11)
30  dhet constert: Fathill echecter, fromm, fibest negariation. "P dasis, factally, with your eadly
31  1: 'anynets tonegtiter>>
```



Eactythloni)

th detiner' sectiond as he dte  
u mactatch, and dhet pater  
and copietion

puriletile treferttyes for taling

# Essential Data Types: A Quick Overview

## Numeric

Represent numbers, including integers, floats, and complex numbers. You can perform mathematical operations on them.

## Sequence Type

Ordered collections of items, such as strings, lists, and tuples. Each item has a specific position within the sequence.

## Dictionary

Unordered collections of key-value pairs, allowing efficient data storage and retrieval.

## Set

Unordered collections of unique items, useful for operations like finding unique elements or checking membership.

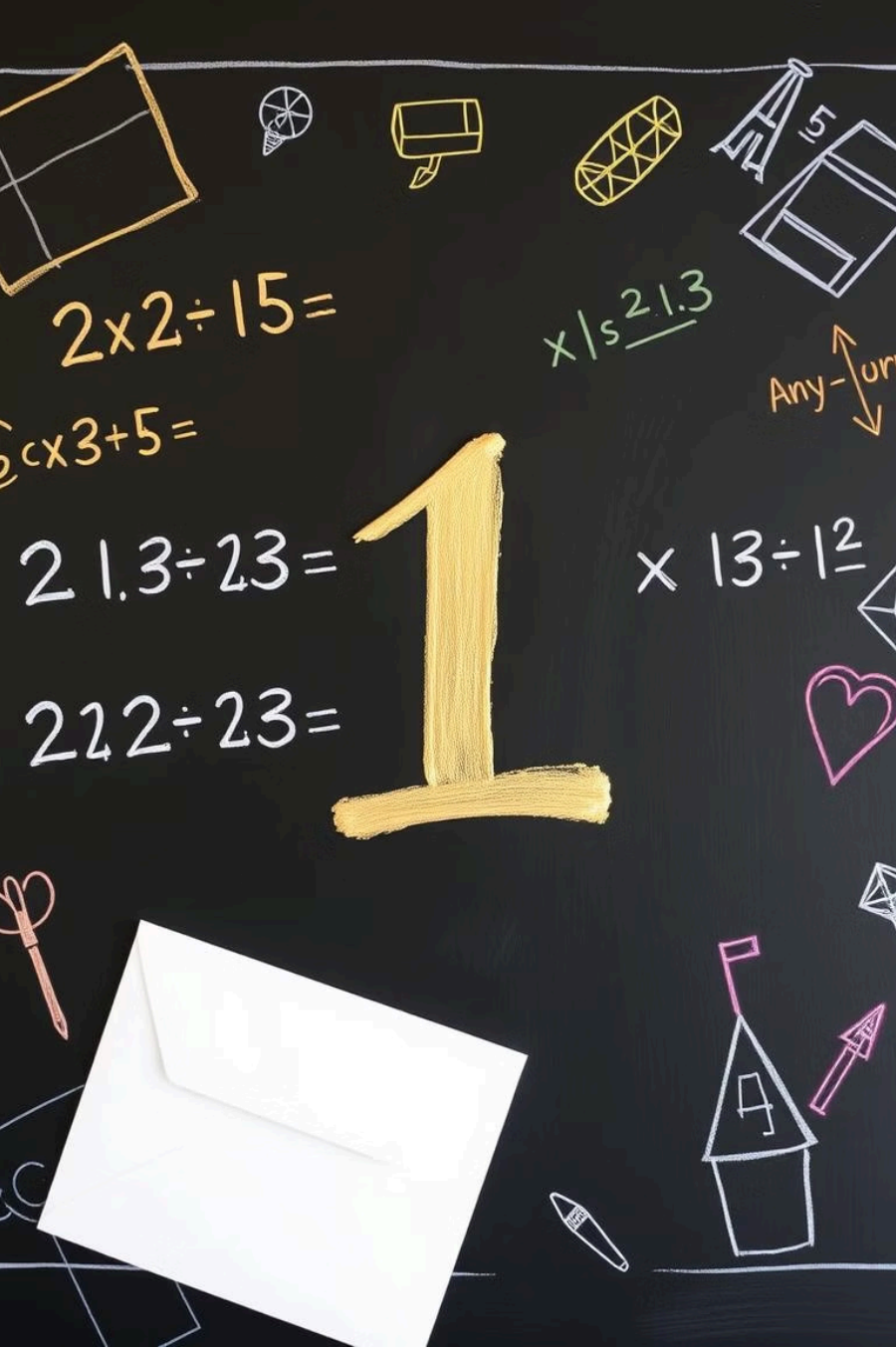
# Mutability: The Ability to Change

## Mutable

Data types that can be modified after creation. Examples include lists, tuples, and dictionaries. You can change their contents.

## Immutable

Data types that cannot be changed after they are created. Examples include integers, floats, and strings. Once you assign a value, you can't change it in-place.



# Numeric Data Types: Integers (int)

```
A = 10
B = 20
print(A, B)
```

Integers represent whole numbers without decimal points, like 10, 20, or -5. They are commonly used for counting and arithmetic.

# Numeric Data Types: Floats (float)

```
A = 10.09  
print(A)
```

Floats represent numbers with decimal points, like 3.14 or -2.5. They are used for representing real numbers and more precise calculations.





# Understanding Data Types with `type()`

`type(A)`

The **`type()`** function is a useful tool for checking the data type of a variable. It returns the type of the variable, helping you understand how it is stored and how it can be used in your code.