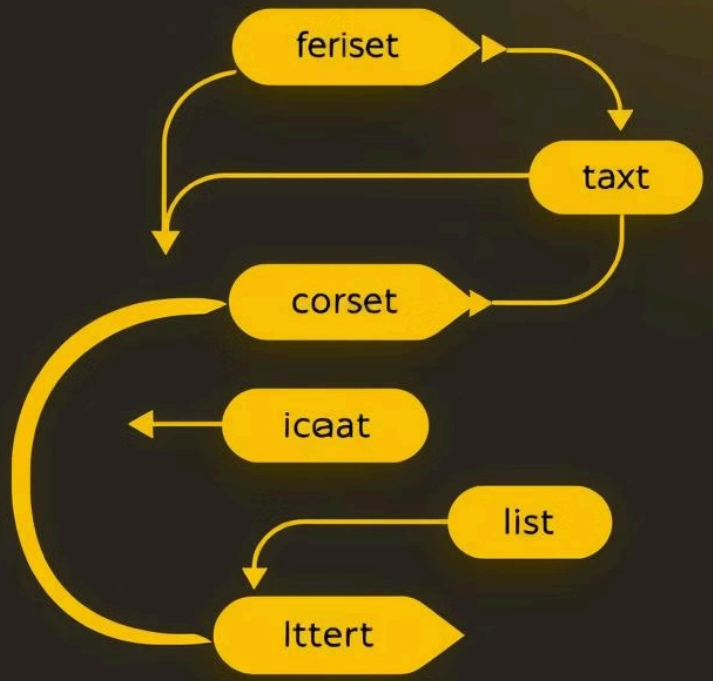


Data Type Conversion in Python

Welcome to our introduction to data type conversion in Python! We'll cover explicit and implicit typecasting, exploring how Python handles data type transformations.

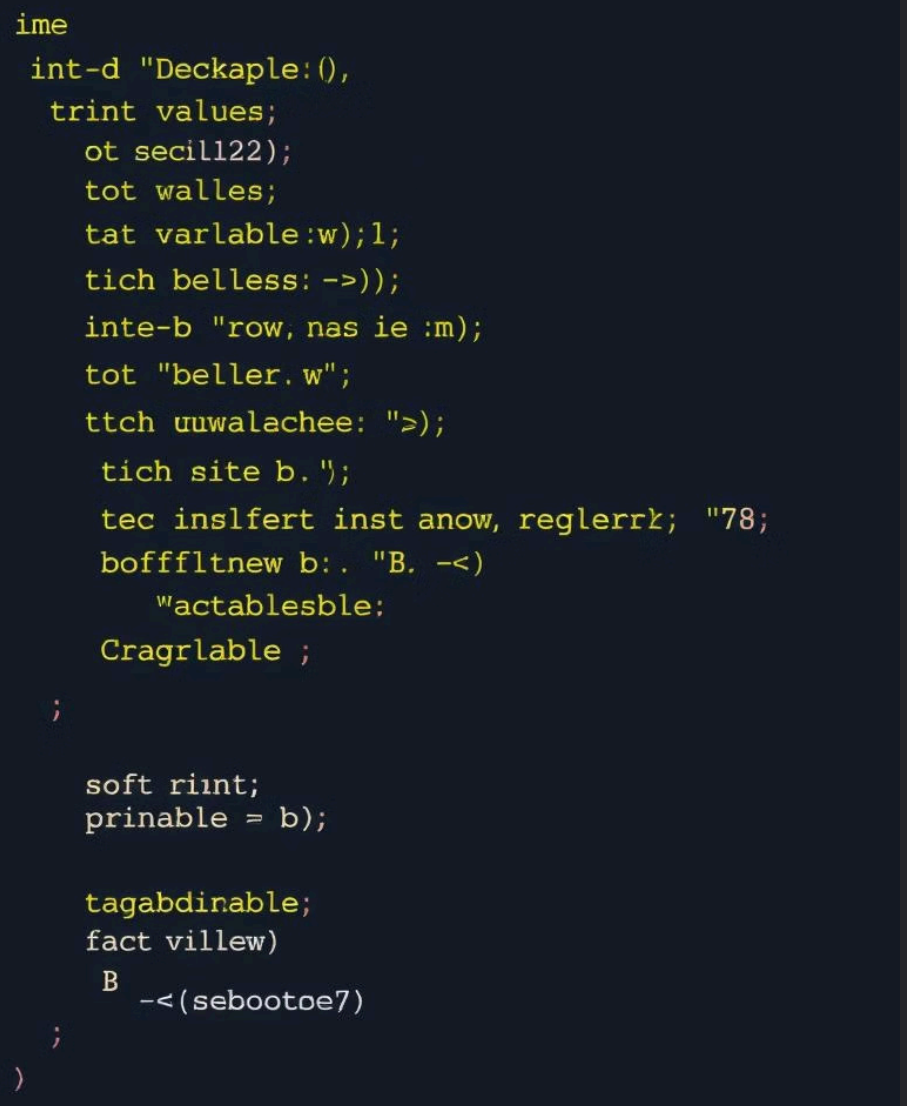
Implicit Typecasting

Automatic Conversion

Python automatically converts lower data types to higher data types. For instance, it converts an integer to a float during addition.

Example

In `a = 5` and `b = 2.5`, adding `a + b` results in `7.5` because Python automatically converts `a` to a float.



Explicit Typecasting

User-Defined

You can manually convert a variable from one type to another using functions like `int()`, `str()`, or `float()`.

Example

With `a = "1"`, you can use `b = int(a)` to explicitly convert the string to an integer, resulting in `b = 1`.

Common Typecasting Functions

int()

Converts to an integer.

float()

Converts to a float.

dict()

Converts to a dictionary

set()

Converts to a set()

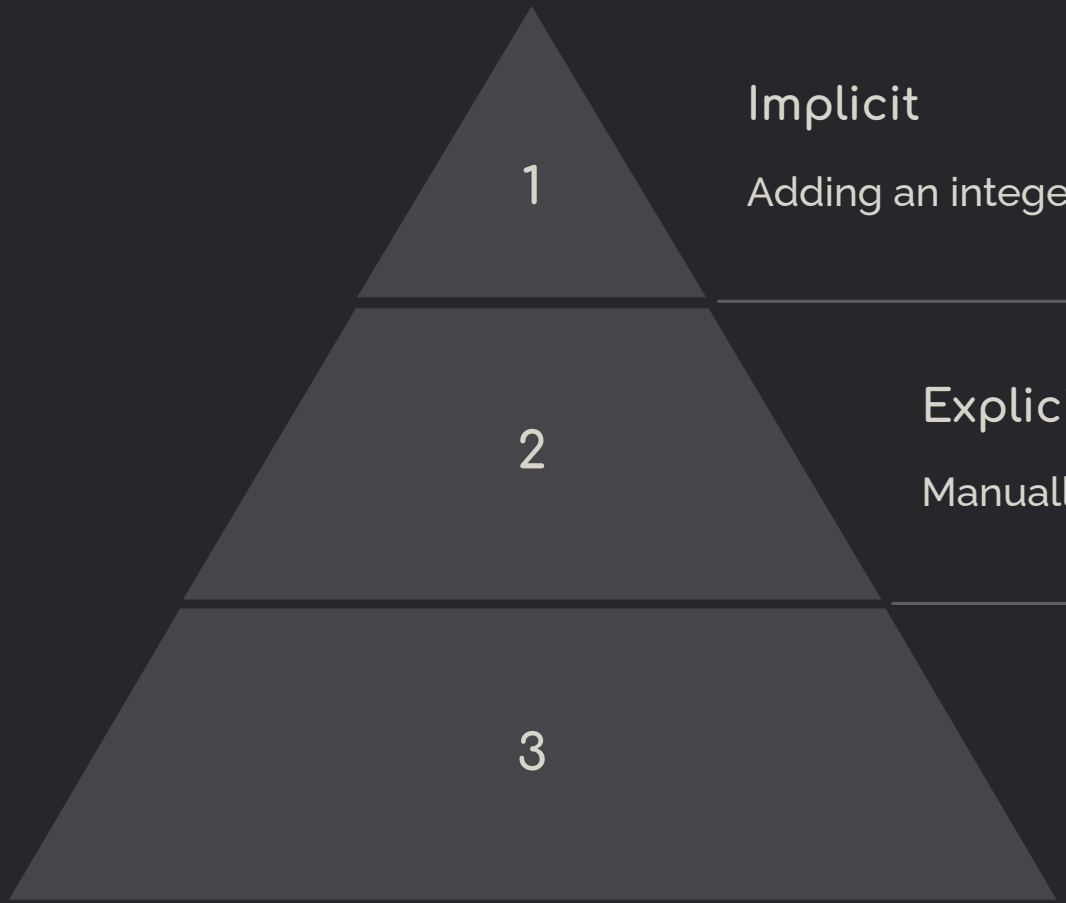
str()

Converts to a string.

list()

Converts to a list.

Example Code for Typecasting



Implicit

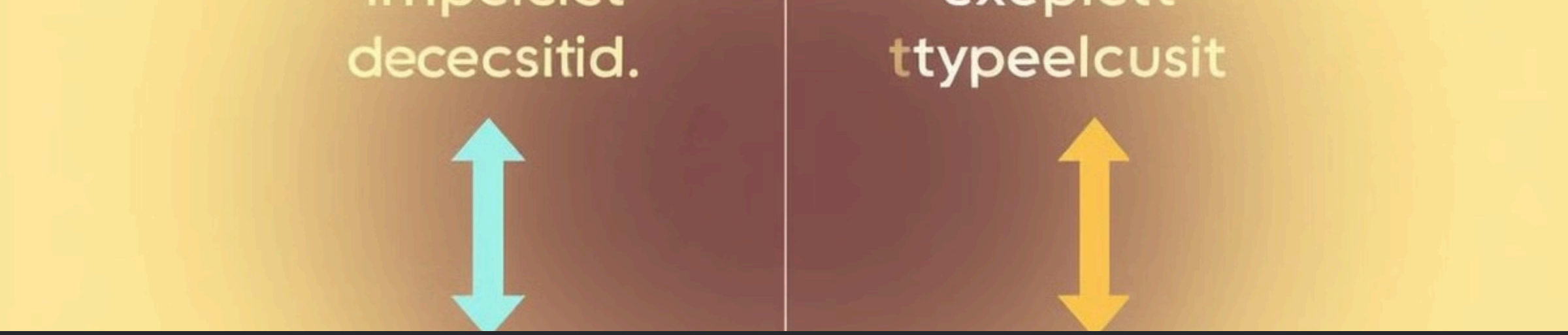
Adding an integer and a float automatically results in a float.

Explicit

Manually converting a string to a float using `float()`.

Further Examples

Converting a string to an integer and an integer to hexadecimal.



Summary

1 **Implicit Typecasting**

Done automatically by Python, e.g., converting an integer to a float.

2 **Explicit Typecasting**

Manually performed by the user using functions like `int()`, `float()`, etc.