Python Series

TYPE CASTING

- Meaning: Type Casting means converting an object from one data type to another.

- Why to use: It ensure compatibility of data type for operations, or to meet API Requirements.

- Type Casting is of two types:
  - Implicit Type Casting
  - Explicit Type Casting

Python Series

IMPLICIT TYPE CASTING

- When a language's compiler or interpreter automatically changes an object from one type to another, it is known as automatic or implicit casting.

- Unrelated type conversion is not done, like, converting string to number.

- Data loss is avoided, i.e, float will not get converted to integer automatically to prevent data loss but an integer can be cast into a float..

Python Series

EXPLICIT TYPE CASTING

- Done forcefully (explicitly) by programmer.

- Precise control over how data is converted.

- Use of data type conversion functions.

- **Use Case**:
  - Converting user input
  - Preparing data for operation that require uniform data type.

- **Data Loss Warning**: Some conversion result in loss of data.

- **Error Handling**: Non-compatible conversion will raise ValueError

Python Series

IMPLICIT TYPE CASTING - ANALOGY

Implicit conversion is like putting a small box inside a large box, ensuring no loss of data.

Python Series

EXPLICIT TYPE CASTING - ANALOGY

Explicit conversion is like trying to fit a large box into a small box, leading to data loss.

Python Series

TYPE CASTING FUNCTIONS (IMPORTANT)

Function	Description	Usage Example	Converts From	Converts To
int()	Converts to an integer.	int(3.14) → 3	float, string	int
float()	Converts to a floating-point number.	float("3.14") → 3.14	Integer, string	float
str()	Converts to a string.	str(100) → '100'	Integer, float, list, etc.	str
list()	Converts to a list.	list("hello") → ['h', 'e', 'l', 'l', 'o']	string, tuple, set, etc.	list
tuple()	Converts to a tuple.	tuple([1, 2, 3]) → (1, 2, 3)	string, list, set, etc.	tuple
set()	Converts to a set (removes duplicates).	set("hello") → {'e', 'h', 'l', 'o'}	string, list, tuple, etc.	set
bool(x)	Converts a value to a Boolean.	print(bool(1)) → True	Integer, float, string, list, and other types	Boolean

Python Series

IMPLICIT AND EXPLICIT COMPARISON

Aspect	Implicit Conversion	Explicit Conversion
Definition	Automatic by Interpreter in Python	Manual by programmer
Also Known As	Coercion	Casting
Control	Interpreter's job	Programmer's job
Safety	Usually safe	Riskier, needs caution
Use Cases	Small to large type. Less to more precise	Large to small type. More to less precise
Examples	integer to float	float to integer
Considerations	Compiler checks safety	Programmer ensures logic

Python Series

COMMON ERRORS

- Invalid String Format for Numeric Conversion

- Empty String Conversion

- Incorrect Use of Type Casting Functions

- Incorrect Data Types for Type Casting

- Naming Conflicts with Built-in Functions

- String Representing Complex Numbers

Python Series

IN THE NEXT VIDEO

Operator - Overview

Python Series