**type casting**

Type casting is the process of converting a value from one data type to another in programming. In Python, type casting is facilitated by various built-in functions that allow you to explicitly change the data type of a variable or value. Here are some common type casting functions in Python: suppose we have an element x:

1. **int(x) - Convert to Integer:**

Converts a value to an integer. If the value is a floating-point number, the decimal part is eliminated.

example:

float\_number = 3.14

integer\_number = int(float\_number)

print(integer\_number)  
 =3

1. **float(x) - Convert to Float:**

Converts a value to a floating-point number.

Exampla

integer\_number = 5

float\_number = float(integer\_number)

print(float\_number)  
 = 5.0

1. **str(x) - Convert to String:**

Converts a value to a string.

example

number = 42

string\_number = str(number)

print(string\_number)

= '42'

1. **bool(x) - Convert to Boolean:**

Converts a value to a boolean. Values like 0, None, and empty containers (e.g., empty lists, strings, etc.) evaluate to False, while non-zero numbers and non-empty containers evaluate to True. .

example

number = 0

bool\_value = bool(number)

print(bool\_value)

= False

1. **list(x) - Convert to List:**

Converts a sequence to a list separated by comma (,) .

example

numbers = 1, 2, 3

list\_numbers = list(numbers)

= [1, 2, 3]

Python also performs implicit type conversion in certain operations. For example, when you mix integers and floats in arithmetic operations, Python automatically converts the integer to a float for the calculation.

example

result = 5 + 2.0

print(result)

=7.0

Understanding type casting is essential for handling different data types in Python and ensuring that your code behaves as expected when working with variables and values of various types.