Python Data Type Conversion Table

Created By @Mohamed Hamed

1st – year Artificial Intelligence student at KFS university, you can find me down:

- LinkedIn
- GitHub
- YouTube
- ✔ You also can get a good ***folder.*** (Quizzes, Answers and Notes)

Basic Conversion Table:

☑ = Conversion works directly

✕ = Conversion not possible directly

⚠ = Conversion works with special considerations

From \ To	list()	tuple()	set()	str()	dict()	Notes
list	☑	☑	☑	⚠	⚠	list→str creates string representation; list→dict requires [key,value] pairs
tuple	☑	☑	☑	⚠	⚠	tuple→str creates string representation; tuple→dict requires (key,value) pairs
set	☑	☑	☑	⚠	✕	set→str creates string representation; set elements can't be used for dict
str	☑	☑	☑	☑	✕	str→list/tuple/set creates collection of characters
dict	⚠	⚠	⚠	⚠	☑	dict→list/tuple/set extracts keys only; dict→str creates stringrepresentation

Detailed Explanations:

List Conversions:

list → list: list([1, 2, 3]) → [1, 2, 3]

list → tuple: tuple([1, 2, 3]) → (1, 2, 3)

list → set: set([1, 2, 3]) → {1, 2, 3}

list → str: str([1, 2, 3]) → "[1, 2, 3]" (string representation, not individual characters)

list → dict: dict([["a", 1], ["b", 2]]) → {"a": 1, "b": 2} (requires key-value pairs)

Tuple Conversions:

tuple → list: list((1, 2, 3)) → [1, 2, 3]

tuple → tuple: tuple((1, 2, 3)) → (1, 2, 3)

tuple → set: set((1, 2, 3)) → {1, 2, 3}

tuple → str: str((1, 2, 3)) → "(1, 2, 3)" (string representation, not individual characters)

tuple → dict: dict([("a", 1), ("b", 2)]) → {"a": 1, "b": 2} (requires key-value pairs)

Set Conversions:

set → list: list({1, 2, 3}) → [1, 2, 3]

set → tuple: tuple({1, 2, 3}) → (1, 2, 3)

set → set: set({1, 2, 3}) → {1, 2, 3}

set → str: str({1, 2, 3}) → "{1, 2, 3}" (string representation, not individual characters)

set → dict: Not directly possible. Set elements are not key-value pairs.

String Conversions:

str → list: list("abc") → ['a', 'b', 'c']

str → tuple: tuple("abc") → ('a', 'b', 'c')

str → set: set("abc") → {'a', 'b', 'c'}

str → str: str("abc") → "abc"

str → dict: Not directly possible. String characters are not key-value pairs.

Dictionary Conversions:

dict → list: list({"a": 1, "b": 2}) → ["a", "b"] (extracts keys only)

dict → tuple: tuple({"a": 1, "b": 2}) → ("a", "b") (extracts keys only)

dict → set: set({"a": 1, "b": 2}) → {"a", "b"} (extracts keys only)

dict → str: str({"a": 1, "b": 2}) → "{'a': 1, 'b': 2}" (string representation)

dict → dict: dict({"a": 1, "b": 2}) → {"a": 1, "b": 2}

Advanced Conversion Techniques:

Dictionary to list of key-value pairs: list(d.items()) → [('a', 1), ('b', 2)]

Dictionary to list of values: list(d.values()) → [1, 2]

String to dictionary (using eval): eval("{'a': 1, 'b': 2}") → {"a": 1, "b": 2} (use with caution)

List of tuples to dictionary: dict([("a", 1), ("b", 2)]) → {"a": 1, "b": 2}