# TYPE CONVERSION

- Type conversion in Python refers to the process of converting a variable from one data type to another. Python provides several built-in functions to convert between data types. The most common type conversions include

# 1. Implicit Type Conversion:

- Python automatically converts one data type to another when needed without any explicit instruction from the programmer. This is also known as **coercion**.
- x = 10       # integer
- y = 3.14     # float
- # Implicit conversion
- result = x + y
- print(result)    # Output: 13.14 (float)
- print(type(result))  # Output: <class 'float'>

# Explicit Type Conversion:

- In explicit type conversion (also called **type casting**), the programmer manually converts one data type into another using Python's built-in functions.
- Int( )-converts to an integer
- Float( )-Converts to a float.
- str( )-Converts to a string.
- List( ) -Converts to a list
- Tuple( )-Converts to a tuple

# Convert String to Integer:

- A = "123"
- B= int(A)
- print(B)          # Output: 123
- print(type(B))    # Output: <class 'int'>

# Convert Integer to String:

- num = 123
- A = str(num)
- print(A)          # Output: "123"
- print(type(A))    # Output: <class 'str'>

# Convert List to Tuple:

- X= [1, 2, 3]
- Y = tuple(X)
- print(Y)         # Output: (1, 2, 3)
- print(type(Y))   # Output: <class 'tuple'>

# Convert Float to Integer:

- A= 3.14
- B= int(A)
- print(B)          # Output: 3
- print(type(B))    # Output: <class 'int'>

# Important Notes:

- Loss of Data: When converting from one type to another, some information may be lost. For example, converting a float to an integer will remove the decimal part.
- **Incompatible Types**:Some conversions are not possible directly. For instance, converting a string containing alphabetic characters to an integer will raise an error.

# Example of an error:

- A= "hello"
- B= int(A)  # This will raise a ValueError
- Python tries to make operations compatible when different data types are used together by converting one of the types to another without explicit instruction from the programmer. This is done to avoid type errors

# Coercion Example with Division:

- division of two integers results in a float, even if both operands are integers:
- x = 5
- y = 2
- result = x / y
- print(result)        # Output: 2.5
- print(type(result))   # Output: <class 'float'>

# Coercion is Different from Explicit Conversion:

- **Coercion (Implicit Conversion)**: Happens automatically by the interpreter when two incompatible types are involved in an operation.
- **Explicit Conversion (Type Casting)**: Requires the programmer to manually convert types using functions like int( ),str( ).

# Example of explicit type casting

```
num1= input(“enter first number”)
num2= input(“enter second number”)
Print(num1+num2)
```

Output:

```
Enter first number:12
Enter second number:14
```

1214 but output must be 26

# Explicit type casting

```
num1=int( input(“enter first number”))
num2=int( input(“enter second number”))
Print(num1+num2)
```

# Implicit type casting

- num1=20
- num2=20.5
- Print(num1+num2)
- Print(num1*num2)