Tasks – Type Casting (המרת טיפוסים)

Practice converting between types: `int()`, `float()`, `str()`, `bool()`. Do this lab before the input lab so you're ready to cast input to numbers. Create each file, run it, and check the output.

Run scripts with: `python3 script_name.py`

Part 1 – int()

Task 1.1 – String to int (`cast_to_int.py`)

- Create `cast_to_int.py`.
- Assign the string `"42"` to a variable. Convert it to an integer with `int()` and store in another variable. Print the integer and then the result of adding 8 to it.

Expected output:

```
42
50
```

Task 1.2 – Float to int (`cast_float_to_int.py`)

- Create `cast_float_to_int.py`.
- Assign `3.9` to a variable. Convert it to int with `int()` and print. (Note: int truncates toward zero, it does not round.)

Expected output:

```
3
```

Task 1.3 – Two strings to int and add (`cast_two_ints.py`)

- Create `cast_two_ints.py`.
- Assign `"10"` and `"3"` to two variables. Convert both to int, add them, and print the result.

Expected output:

```
13
```

Part 2 – float()

Task 2.1 – String to float (`cast_to_float.py`)

- Create `cast_to_float.py`.
- Assign the string `"3.14"` to a variable. Convert to float with `float()` and print. Then print that value multiplied by 2.

Expected output:

```
3.14
6.28
```

Task 2.2 – Int to float (`cast_int_to_float.py`)

- Create `cast_int_to_float.py`.
- Assign the integer `5` to a variable. Convert to float with `float()` and print.

Expected output:

```
5.0
```

Task 2.3 – String "10" to float and divide (`cast_float_ops.py`)

- Create `cast_float_ops.py`.
- Assign `"10"` to a variable, convert to float, then print the result of dividing it by 4.

Expected output:

```
2.5
```

Part 3 – str()

Task 3.1 – Int to string (`cast_to_str.py`)

- Create `cast_to_str.py`.
- Assign the integer `100` to a variable. Convert to string with `str()` and print. Then print the result of concatenating that string with `" items"` (so the output is `100 items`).

Expected output:

```
100
100 items
```

Task 3.2 – Float to string (`cast_float_to_str.py`)

- Create `cast_float_to_str.py`.
- Assign `2.5` to a variable. Convert to string and print. Use that string in a message like `"Price: " + str(price)` and print it.

Expected output (example):

```
2.5
Price: 2.5
```

Task 3.3 – Number and string in one print (`cast_str_print.py`)

- Create `cast_str_print.py`.
- Assign an integer (e.g. `7`) to a variable. Use `str()` so you can build one string that contains a label and the number (e.g. `"Result: " + str(n)`), and print that string.

Expected output (example):

```
Result: 7
```

Part 4 – bool()

Task 4.1 – Numbers to bool (`cast_to_bool.py`)

- Create `cast_to_bool.py`.
- Print `bool(0)` and `bool(1)`. Then print `bool(-5)` (any non-zero number is truthy).

Expected output:

```
False
True
True
```

Task 4.2 – Strings to bool (`cast_str_to_bool.py`)

- Create `cast_str_to_bool.py`.
- Print `bool("")` and `bool("hello")`. Empty string is falsy; non-empty is truthy.

Expected output:

```
False
True
```

Part 5 – Mixed casting

Task 5.1 – Chain: string → int → float (`cast_chain.py`)

- Create `cast_chain.py` .
- Assign `"100"` to a variable. Convert to int, then assign that int to another variable. Convert that int to float and print the float.

Expected output:

```
100.0
```

Task 5.2 – Print label and number using str() (`cast_label_number.py`)

- Create `cast_label_number.py` .
- Assign an int and a float (e.g. `age = 25` , `height = 1.75`). Build two messages using `str()` so you can concatenate: e.g. "Age: 25" and "Height: 1.75". Print both (one per line or with sep).

Expected output (example):

```
Age: 25
Height: 1.75
```

Done

You've used: `int()` , `float()` , `str()` , and `bool()` for type casting. In the next lab (input) you'll use `int()` and `float()` to convert user input to numbers.