Walkthrough 1: Earth to Mars Weight Converter

Hi everyone! If you're working through this Python program and need clarity, let's walk through the code line by line so you understand exactly what's going on.

Program 1: Earth to Mars Weight

```
"""
Prompts the user for a weight on Earth
and prints the equivalent weight on Mars.
"""
```

This comment at the top tells us what the program does. It’s always a good practice!

The Main Function

```
def main():
```

- This is the entry point for the program. Everything happens inside this function.

Step 1: Constants

```
MARS_GRAVITY_FACTOR = 0.378
```

- This tells the program that Mars has about **37.8%** of Earth's gravity.
- We store this number in a **constant** so it’s easy to reuse and change if needed.

Step 2: Ask the User

```
earth_weight_str = input("Enter your earth weight: ")
```

- This shows a prompt to the user and **captures their response** as a string.
- Even if the user types a number, Python treats it as text (`"150"` not `150`).

Step 3: Convert to Number

```
earth_weight = float(earth_weight_str)
```

- We convert the string to a **float**, so we can do math with it.

Step 4: Calculate Mars Weight

```
mars_weight = earth_weight * MARS_GRAVITY_FACTOR
```

- Simple multiplication gives us the user's **Mars weight**.

Step 5: Round It

```
rounded_mars_weight = round(mars_weight, 2)
```

- We round to **2 decimal places** so the result looks neat.
- `2` here means "two digits after the decimal point".

Step 6: Print the Result

```
print("Your weight on Mars is: " + str(rounded_mars_weight))
```

- Why do we use `str()`? Because:
  - `print()` wants **text**, and
  - `rounded_mars_weight` is a number (`float`), so we must **convert it to a string**.

Program Execution

```
if __name__ == "__main__":
    main()
```

- This starts the program by calling `main()`.

Program 2: Weight on Any Planet

This version lets the user choose **which planet** they want!

The Main Function

```
def main():
```

Step 1: Ask for Planet

```
planet_choice = input("Enter your planet choice: ")
```

- The user types `"Mars"`, `"Venus"`, `"Jupiter"`, or `"Mercury"`.

Step 2: Gravity Constants

```
MARS_GRAVITY = 0.378

MERCURY_GRAVITY = 0.376

VENUS_GRAVITY = 0.889

JUPITER_GRAVITY = 2.36
```

- Each planet has a different gravity, stored as **named constants**.

Step 3: Ask for Earth Weight

```
earth_weight_str = input("Enter your earth weight: ")

earth_weight = float(earth_weight_str)
```

- As before, we ask the user for a weight and convert it to a number.

Step 4: Decide Which Planet

```
final_weight = 0.0

if planet_choice == "Mars":
    final_weight = MARS_GRAVITY * earth_weight
...
```

- We use `if` statements to check the planet name and do the correct math.

Step 5: Round and Print

```
final_weight_rounded = round(final_weight, 2)
print("Your weight on " + planet_choice + "  is " + str(final_weight))
```

- Again, we round for clarity.
- And we convert the number to a string for printing.

Note: It would be better to print `final_weight_rounded` here instead of `final_weight`, for a cleaner result.

Mini-Lesson: Local vs. Global Variables

- **Local Variables** (like `earth_weight` or `mars_weight`) are created **inside a function**.
    - They're only visible and usable within that function.
- **Global Variables** live outside any function and can be used anywhere (but are generally avoided unless really needed).

In our programs, all variables are **local**, which is great!
It keeps things **modular** and **easy to debug**.

Again, why Use `str()` to Print Numbers?

When you write:

```
print("Weight is " + some_number)
```

Python throws an error — you can't add a string and a number.
So we **convert the number to a string** first:

```
print("Weight is " + str(some_number))
```

This way, Python knows you want to **combine text with a number**.

Recap

These programs:

- Collect user input (weight + planet).
- Convert text to numbers.
- Use multiplication and rounding.
- Print a clear result.
- Teach you about constants, variables, and simple control flow.