Assignment-1

1. **Values vs. Expressions:**
    - **Values:**
        - `'hello'` (string) - Text enclosed in quotes.
        - `-87.8` (float) - A number with a decimal point.
        - `6` (integer) - A whole number.
    - **Expressions:**
        - `-` (minus), `/` (division), `+` (addition) - Mathematical operators that combine values.

2. **Difference between String and Variable:**
    - **String:** A sequence of characters like text, enclosed in quotes (e.g., `'hello'`). It holds a fixed value.
    - **Variable:** A named storage location for data. You can assign a value (like a string, number, etc.) to it, and that value can change later in your program. Think of it like a labelled box that can hold different things.

3. **Three Data Types:**
    - **Integer:** Represents whole numbers (e.g., `10`, `-5`).
    - **Float:** Represents numbers with decimals (e.g., `3.14`, `-12.5`).
    - **String:** Represents text data (e.g., `'This is a string'`).

4. **Expressions:**
    - **Made up of:** Values, operators, function calls, and parentheses to define the order of operations.
    - **What they do:** Expressions evaluate to a single result. They can perform calculations, comparisons (e.g., `x > 5`), or manipulate data.

5. **Expressions vs. Statements:**
    - **Expression:** Evaluates to a single result (e.g., `2 + 3`).
    - **Statement:** A complete instruction that tells the program to do something (e.g., an assignment statement like `spam = 10`). Assignment statements use the `=` operator to assign the result of an expression (the value on the right) to a variable (the name on the left).

6. **Variable Value after Code:**

- `bacon = 22` assigns the value `22` (integer) to the variable `bacon`.
- `bacon + 1` is an expression, but it's not assigned to a variable. The expression itself doesn't change the value of `bacon`. It would evaluate to `23` if used in a complete statement.

7. **String Operations:**
   - `'spam' + 'spamspam'` combines the strings using concatenation, resulting in `'spamspamam'`.
   - `'spam' * 3` repeats the string three times, resulting in `'spamspamamspam'`.

8. **Valid Variable Names:**
   - `eggs` **is valid:** Variable names can start with letters (a-z, A-Z) or underscores (_), and can contain letters, numbers, and underscores.
   - `100` **is invalid:** Variable names cannot start with numbers.

9. **Functions for Data Type Conversion:**
   - `int(value)`: Converts a value to an integer (e.g., `int(3.14)` would be `3`).
   - `float(value)`: Converts a value to a floating-point number (e.g., `float(10)` would be `10.0`).
   - `str(value)`: Converts a value to a string (e.g., `str(True)` would be `'True'`).

10. The expression 'I have eaten ' + 99 + ' burritos.' causes an error because you're trying to directly add a number (99) to strings. In Python, you can only concatenate (join) strings with other strings, not numbers.

    The correct message is:

    'I have eaten ' + str(99) + ' burritos.'