Answer_1:

Here are the values and expressions in the given elements:

- Values:
  - ‘hello’
  - -87.8
  - 6
- Expressions:
  - *
  - -
  - /
  - +

Answer_2:

A string is a sequence of characters enclosed in quotes (either single quotes or double quotes) and is used to represent text.

For example: my_string = "Hello, world!"

A variable is a name that refers to a value. The value can be of any type, including strings.

For example: my_variable = "Hello, world!"

In this case, `my_variable` is a variable that refers to the string `"Hello, world!"`.

So the main difference between a string and a variable is that a string is a specific type of value (a sequence of characters), while a variable is a name that can refer to any type of value.

Answer_3:

Here are three different data types:

1. Integer: An integer is a whole number (i.e., a number with no decimal places). For example: `1`, `2`, `-3`, `0`.
2. String: A string is a sequence of characters enclosed in quotes (either single quotes or double quotes) and is used to represent text. For example: `"Hello, world!"`, `'Python is fun!'`.
3. Boolean: A boolean is a data type that can have one of two values: `True` or `False`. Booleans are often used in conditional statements and loops to control the flow of a program. For example: `True`, `False`.
4. Float: A float is a number with a decimal point (e.g., `3.14`, `-2.5`, `0.0`).

Answer_4:

An expression is made up of one or more values and operators that can be evaluated to produce a result. For example, `2 + 3` is an expression that evaluates to `5`.

Expressions can be used in many different ways in programming. For example:

- Assigning a value to a variable: `x = 2 + 3`
- Printing a value to the console: `print(2 + 3)`
- Testing a condition: `if x > 5:`

All expressions evaluate to a value. The value can be of any data type (e.g., integer, float, string, boolean) and can be used in other expressions or statements.

Answer_5:

An expression is a combination of values, variables, and operators that can be evaluated to produce a value. For example, `2 + 3` is an expression that evaluates to `5`.

A statement is a unit of code that performs some action. For example, an assignment statement like `spam = 10` assigns the value `10` to the variable `spam`.

The main difference between an expression and a statement is that an expression evaluates to a value, while a statement does not. However, many statements contain expressions. For example, the assignment statement `spam = 10` contains the expression `10`.

Answer_6:

After running the following code:
bacon = 22
bacon + 1

The variable `bacon` contains the value `22`.

The expression `bacon + 1` evaluates to `23`, but the result is not assigned to any variable. If you want to update the value of `bacon`, you need to assign the result of the expression back to the variable:
bacon = bacon + 1

Now the variable `bacon` contains the value `23`.

Answer_7:

The values of the following two terms are:

```
'spam' + 'spamspam'    # 'spamspamspam'
'spam' * 3             # 'spamspamspam'
```

In the first term, the `+` operator concatenates the two strings `'spam'` and `'spamspam'` to produce the string `'spamspamspam'`.

In the second term, the `*` operator repeats the string `'spam'` three times to produce the string `'spamspamspam'`.

Answer_8:
In Python, variable names must follow certain rules:

- They can only contain letters (a-z, A-Z), digits (0-9), and underscores (_).
- They cannot start with a digit.
- They cannot be a reserved word (e.g., `if`, `else`, `while`, etc.).

Since `eggs` only contains letters and is not a reserved word, it is a valid variable name.

On the other hand, `100` starts with a digit, which is not allowed in variable names. Variable names must start with a letter or an underscore.

Answer_9:
In Python, you can use the following functions to convert a value to a different data type:

- `int()`: Converts a value to an integer.
- `float()`: Converts a value to a floating-point number.
- `str()`: Converts a value to a string.

For example:

```
x = 42
y = 3.14
z = 'hello'

# Convert x to a float
x_float = float(x)

# Convert y to an integer
y_int = int(y)

# Convert z to an integer (this will raise an error)
z_int = int(z)

# Convert x to a string
x_str = str(x)
```

Answer_10:
This expression causes an error because you cannot concatenate a string and an integer. In the expression `'I have eaten ' + 99 + ' burritos.'`, the value `99` is an integer, so you cannot concatenate it directly with the strings.

To fix this error, you can convert the integer to a string using the `str()` function:

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

This will produce the string `'I have eaten 99 burritos.'`