Here are the answers to your questions:

1. \*\*Why are functions advantageous to have in your programs?\*\*

- Functions are advantageous because they:

- Promote code reuse, reducing redundancy.

- Improve readability and organization by encapsulating functionality.

- Make debugging easier by isolating code into manageable pieces.

- Enhance maintainability by allowing changes to be made in one place rather than throughout the code.

- Facilitate testing by providing isolated units of code.

2. \*\*When does the code in a function run: when it's specified or when it's called?\*\*

- The code in a function runs when it is called, not when it is specified.

3. \*\*What statement creates a function?\*\*

- The `def` statement is used to create a function in Python. For example:

```python

def my\_function():

# code block

```

4. \*\*What is the difference between a function and a function call?\*\*

- A function is a block of code that performs a specific task and can be reused. A function call is an expression that invokes the function to execute its code. For example, `my\_function()` is a function call that executes the code inside `my\_function`.

5. \*\*How many global scopes are there in a Python program? How many local scopes?\*\*

- There is one global scope in a Python program. There can be multiple local scopes, one for each function or class.

6. \*\*What happens to variables in a local scope when the function call returns?\*\*

- Variables in a local scope are destroyed when the function call returns, meaning they are no longer accessible outside the function.

7. \*\*What is the concept of a return value? Is it possible to have a return value in an expression?\*\*

- The return value is the result that a function sends back to the caller when it completes execution. Yes, it is possible to use a return value in an expression. For example:

```python

def add(a, b):

return a + b

result = add(2, 3) \* 10 # The return value of add(2, 3) is used in the expression.

```

8. \*\*If a function does not have a return statement, what is the return value of a call to that function?\*\*

- If a function does not have a return statement, it returns `None` by default.

9. \*\*How do you make a function variable refer to the global variable?\*\*

- To make a function variable refer to a global variable, you can use the `global` keyword inside the function. For example:

```python

x = 10

def update\_x():

global x

x = 20

```

10. \*\*What is the data type of `None`?\*\*

- The data type of `None` is `NoneType`.

11. \*\*What does the sentence `import areallyourpetsnamederic` do?\*\*

- The sentence `import areallyourpetsnamederic` attempts to import a module named `areallyourpetsnamederic`. If such a module does not exist, it will raise an `ImportError`.

12. \*\*If you had a `bacon()` feature in a `spam` module, what would you call it after importing `spam`?\*\*

- You would call it using:

```python

import spam

spam.bacon()

```

13. \*\*What can you do to save a program from crashing if it encounters an error?\*\*

- You can use error handling mechanisms such as `try` and `except` blocks to catch and handle exceptions, preventing the program from crashing.

14. \*\*What is the purpose of the `try` clause? What is the purpose of the `except` clause?\*\*

- The `try` clause contains code that might raise an exception. The `except` clause contains code that handles the exception if it occurs. This allows for graceful error handling and recovery without crashing the program.