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

Functions allow you to break down your code into reusable and organized pieces. They promote modularity, readability, and maintainability by encapsulating specific tasks and logic. This makes your code easier to understand, test, and debug.

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 the function is \*\*called\*\*. Defining a function does not execute its code; it only defines the behavior of the function.

3. What statement creates a function?

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

```python

def my\_function():

# related code

```

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

A function is a block of code that performs a specific task, while a function call is the act of executing that code by invoking the function with its name followed by parentheses.

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

There is one global scope in a Python program, and it encompasses the entire program. Local scopes are created within functions and are temporary, existing only when the function is called.

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

Variables in a local scope are destroyed and their values are lost when the function call returns. They are only accessible within the scope of that function.

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

A return value is the result that a function sends back to the caller. It's used to pass data from the function's execution back to the calling code. Yes, it's possible to use a return value in an expression, like assigning it to a variable or using it as part of a larger calculation.

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

If a function doesn't have a `return` statement, it implicitly returns `None`.

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

You can use the `global` keyword within a function to indicate that a variable should refer to a global variable instead of creating a new local variable.

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

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

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

This is not a valid Python module import statement. It seems to be a humorous example and wouldn't have any actual functionality.

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

You would call the `bacon()` feature using the module name: `spam.bacon()`.

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

You can use error handling mechanisms like `try` and `except` blocks to catch and handle exceptions gracefully. This prevents the program from crashing and allows you to handle errors in a controlled manner.

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

The `try` clause is used to wrap code that might raise an exception. It's an attempt to execute the code and catch any exceptions that might occur. The `except` clause is used to specify what actions should be taken if a specific exception is raised within the `try` block.