Certainly, here are the answers in a question-answer format:

Q1. Is an assignment operator like += only for show? Is it possible that it would lead to faster results at runtime?

- Answer: No, assignment operators like `+=` serve a purpose by modifying variables in place and can lead to faster results in some cases.

Q2. What is the smallest number of statements needed to replace the Python expression `a, b = a + b, a` in most programming languages?

- Answer: In most languages, you need at least three statements to achieve the same result.

Q3. In Python, what's the most effective way to set a list of 100 integers to 0?

- Answer: The most effective way is to use `[0] \* 100` to create the list.

Q4. What's the most effective way to initialize a list of 99 integers that repeats the sequence 1, 2, 3 in Python?

- Answer: You can do this with `[1, 2, 3] \* 33`.

Q5. When using IDLE to run a Python application, how can you efficiently print a multidimensional list?

- Answer: Use nested loops to iterate through and print the elements of the list.

Q6. Is it possible to use list comprehension with a string in Python? If so, how?

- Answer: Yes, you can use list comprehension with a string to create lists of characters or manipulate them.

Q7. How do you seek support for a user-written Python program from the command line? Is this possible from inside IDLE?

- Answer: You can use `python -c` from the command line or `help()` inside IDLE to access support for Python programs.

Q8. What can you do with a function (callable object) in Python that you can't do in C or C++ due to Python's "first-class functions" feature?

- Answer: Python's first-class functions allow treating functions as objects, enabling higher-order functions, dynamic function selection, and more, which is not as straightforward in C or C++.

Q9. How do you distinguish between a wrapper, a wrapped feature, and a decorator in Python?

- Answer: A wrapper adds functionality around a feature, the feature is the core, and decorators are a specific type of Python wrapper.

Q10. If a function is a generator function in Python, what does it return?

- Answer: A generator function returns a generator object, not a value, and generates values using `yield`.

Q11. What's the one change needed to turn a regular function into a generator function in Python?

- Answer: Replace `return` statements with `yield` statements.

Q12. What's a key benefit of generators in Python?

- Answer: A key benefit of generators is efficient memory usage by generating values lazily, useful for large data sets.