Python assignment.

1. Built in modules are the ones that are already there and no need to use specific code.
2. Python modules are files containing python code that can be imported and used in other programming python. Its significance is that the modules can be used in other python codes enhancing the reusability of the codes.
3. A docstring is a string literal that occurs as the first statement in a module, function, class or method definition. It is useful in a way that they provide a convenient way of associating documentation with python modules making it easier for others to understand what your code does.
4. It will check if the module has already been imported if not it will search it in the directories and if it has it will skip the import process. The python will recognize it has already been imported therefore it will not reload or re-execute the module since it will ensure it is executed only once.
5. Argument is a value that you pass to a function or method when you call it which allows you to provide data and use to perform operations. For example; positive arguments passed to a function in specific order.
6. The built-in help() function in Python is a very useful for accessing documentation and information about Python objects, modules, functions, and classes directly from the Python interactive shell or within your code.
7.
   ```
   import math
   def sphere_volume():
      # Ask user for input
     diameter = float(input("Enter the diameter of the sphere (in centimeters): "))
     # Set radius to one-half of the diameter
     radius = diameter / 2
       # Calculate volume
       volume_cm3 = (4/3) * math.pi * (radius ** 3)
        # Convert volume to liters
       volume_liters = volume_cm3 / 1000
        # Print statement with volume estimate
       print(f"The volume of the sphere is {volume_liters:.2f} liters.") #2f is for the estimation to 2
   decimal places
      # Return the volume
       return volume_liters
   # Example usage
   result = sphere_volume()
   print(f"{result:.2f} liters")
   ```
8.
   ```
   def interest (prnc, time=2, rate=0.10): return (prnc * time * rate) print(interest(6100, 1))
   print(interest(5000, rate=0.05)) print(interest(5000, 3, 0.12)) print(interest(time=4, prnc=5000))
   #output
   610.0
   500.0
   1800.0
   2000.0
   ```

9. The math module is useful in providing mathematical functions while the random module is used to generate random numbers

10. A module list typically refers to a collection or inventory of modules that are part of a larger system or program.

The two reasons why we need modules are code reusability the modules allow you to write code once and reuse it in multiple places. Instead of duplicating code, you can import the same module across various parts of your application, making development more efficient and reducing errors and separation of concerns the modules help break down complex programs into smaller, manageable, and logically separated pieces. This improves code organization, making it easier to maintain, debug, and scale. For example, you might have a module for handling user authentication and another for database management

```
11. # the math_operation module
def add(a, b):
 return a + b
def subtract(a, b):
 return a-b # subtract

print("Adding numbers: ", add(1,2))
print("Subtracting numbers: ", subtract(1,2))
#output
Adding numbers:  3
Subtracting numbers:  -1
```

```
12. # Import the subtract function from the math_operations module
from math_operations import subtract
# Subtract 1 from 2
print("Subtract 1 from 2:", subtract(2, 1))
# Import everything from math_operations (not recommended for large modules)
from math_operations import add
# Using the functions
print("Subtract 2 from 1:", subtract(2, 1))
print("Add 1 and 1:", add(1, 1))
#output
```

```
Subtract 1 from 2: 1

Subtract 2 from 1: 1

Add 1 and 1: 2
```

13. When you use import *, all the public objects (functions, classes, variables, etc.) from the module are imported into the current namespace. This can lead to naming conflicts where a function or variable in your code has the same name as one of the imported items. This makes the code harder to debug and maintain, especially in larger projects.

14. A Python module is simply a file that contains Python code, such as functions, classes, or variables, which can be imported and used in other Python programs. A module allows you to logically organize your Python code into smaller, manageable, and reusable components.

15. A module is a file python functions and statements with .py extensions while package is a directory of a python module.

16. The "RESTART" banner is a way of informing you that the Python interpreter has been restarted before running the new code

17. # mod1 def change(a): b = [x * 2 for x in a] print(b) # mod2 def change(a): b = [x * x for x in a] print(b) from mod1 import change from mod2 import change # main s = [1, 2, 3] change(s)

```
#output

Result from mod1: [2, 4, 6]

Result from mod2: [1, 4, 9]
```

18. from math import factorial

```
print(factorial(5))

import math

print(math.factorial(5))

#output

120

120
```

19. Which of the following is not an advantage of using modules? (a) Provides a means of reusing program code (b) Provides a means of dividing up tasks (c) Provides a means of reducing the size of the program (d) Provides a means of testing individual parts of the program

The answer is c

20. import statistics

```
# List of heights in feet

heights = [5.9, 5.5, 6.1, 6.0, 7.2]

# Calculate the average height

average_height = statistics.mean(heights)

# Calculate the median height

median_height = statistics.median(heights)

# Display the results

print(f"Average Height: {average_height:.2f} feet") # 2f is answer in 2 decimal places

print(f"Median Height: {median_height:.2f} feet")

# output

Average Height: 6.14 feet

Median Height: 6.00 feet
```

```
21. import random

random_integer = random.randint(1, 9)

print("Random integer between 1 and 9:", random_integer)

#output

Random integer between 1 and 9: 3
```