Q1. Creating variables with different types of data:

```
# String
variable_string = "Hello, world!"
# List
variable_list = [1, 2, 3, "apple", "banana"]
# Float
variable_float = 3.14
# Tuple
variable_tuple = (10, 20, "cat", "dog")
```

Q2. Data types of given variables:

```
# var1 is an empty string, so its data type is 'str'
# var2 is a string containing a list in string format, so its data type is 'str'
# var3 is a list containing strings, so its data type is 'list'
# var4 is a float, so its data type is 'float'
```

Q3. Explanation of operators:

/: Division operator - used for normal division.

%: Modulus operator - gives the remainder of division.

//: Floor division operator - gives the quotient without the decimal part.

**: Exponentiation operator - raises a number to a power.

Example:

```
a = 10
b = 3
division_result = a / b  # 3.333...
remainder = a % b  # 1 (remainder of 10 divided by 3)
floor_division_result = a // b  # 3 (quotient of 10 divided by 3)
exponent_result = a ** b  # 1000 (10 raised to the power of 3)
```

Q4. Creating a list and printing element types using a for loop:

python

Copy code

```
my_list = [5, "apple", 3.14, True, [1, 2, 3], 'c', 7.5, False, (1, 2), None]
for element in my_list:
    print(f"Element: {element}, Data Type: {type(element)}")
```

Q5. Using a while loop to verify divisibility:

python

Copy code

```
A = 30
B = 5
count = 0
while A % B == 0:
    A /= B
    count += 1
print(f"A can be divided by B {count} times.")
```

Q6. Checking divisibility by 3 using for loop and if-else:

python

Copy code

```
numbers = [2, 6, 8, 9, 12, 15, 18, 21, 25, 30, 35, 42, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105]
for num in numbers:
    if num % 3 == 0:
        print(f"{num} is divisible by 3")
    else:
        print(f"{num} is not divisible by 3")
```

Q7. Mutable and immutable data types:

Mutable: Objects where we can change values value after creation. Lists and dictionaries are mutable.

Immutable: Objects whose value cannot be changed after creation. Integers, floats, strings, and tuples are immutable.

Examples:

Mutable: Lists

python

Copy code

```
my_list = [1, 2, 3]

my_list[0] = 10  # Changing the first element

print(my_list)   # Output: [10, 2, 3]
```

Immutable: Strings

python

Copy code

```
my_string = "Hello"

# Attempting to change a character in the string will result in an error.

# Strings are immutable, so you can't change their characters directly.
```