Quiz 1 - Ruby

Full name

Q1 - What's a variable? What's the point of using variables? Give an example.

Q2 - Precisely describe the following line of code **using the correct vocabulary.**

```
1  age = 18
```

Q3 - What's a method? What's the point of defining methods?

Q4 - Precisely describe what happens at **each line** using good vocabulary.

```
1  def multiply(x, y)
2    return x * y
3  end
4
5  puts multiply(5, 8)
```

Q5 - What's the keyword `if`? Give us an example of `if` statements, using an `age` variable storing a student's age for instance.

```
1  # TODO: write some code with `if`
2  age =
3
4
5
6
7
```

Q6 - Complete the following code to compute the **exact** average of students grades (using `each`).

```
1  grades = [19, 8, 11, 15, 13]
2  # TODO: compute and store the result in a variable `average`
3
4
5
6
7
8
```

Q7 - Define a `capitalize_name` method which takes first name and last name as parameters and returns the well-formatted fullname (with capitalized first and last names).

```
1  # TODO: Implement `capitalize_name`
2
3
4
5
6
7
```

Q8 - Let's speak of the type `String`. What's the difference between **concatenation** and **interpolation**? Give an example.

Q9 - Translate each line of pseudo-code into ruby.

```
1  fruits = ["banana", "peach", "watermelon", "orange"]
2  # Print out "peach" from the fruits array in the terminal
3
4  # Add an "apple" to the fruits array
5
6  # Replace "watermelon" by "pear"
7
8  # Delete "orange"
9
```

Q10 - Translate each line of pseudo-code into ruby.

```
1   city = { name: "Paris", population: 2000000 }
2   # Print out the name of the city
3
4   # Add the Eiffel Tower to city with the `:monument` key
5
6   # Update the population to 2000001
7
8   # What will the following code return?
9   city[:mayor]
10  # =>
```

Q11 - Use the `map` iterator to convert the variable `students`, an array of arrays, into an array of hashes. Those hashes should have two keys: `:name` and `:age`. What is the type of those keys?

```
1   students = [ [ "john", 28 ], [ "mary", 25 ], [ "paul", 21 ] ]
2
3
4
5
6
7
8
9
10
```