Mohammed Roshan Khan

Day 10 – Assignment

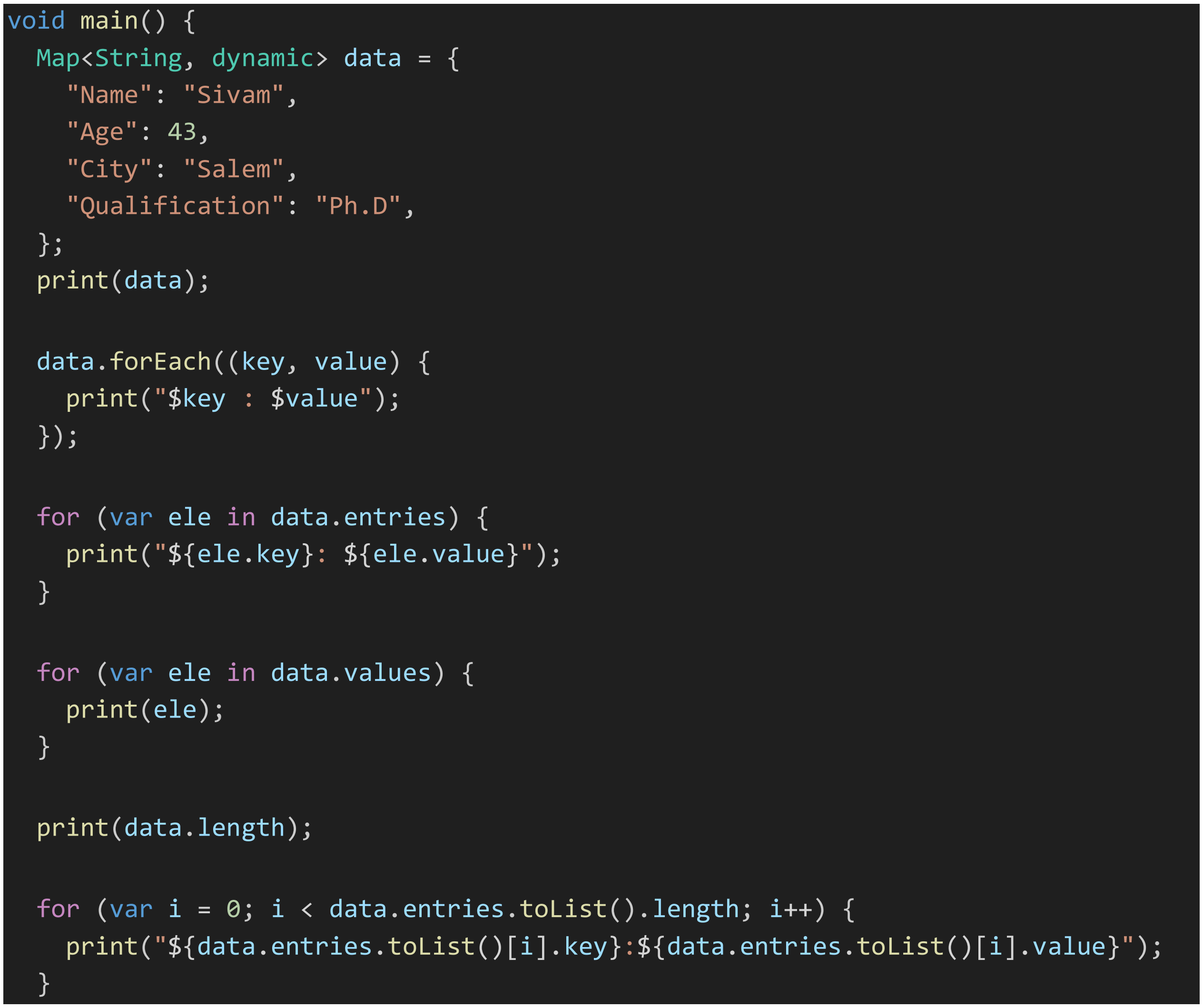
Use of for loops in dart while using Maps

In Dart, a for loop can be used with maps to iterate over each key-value pair.

You typically access the entries of the map to loop through them efficiently.

This allows you to perform actions using both the keys and their corresponding values.

Ex:-



```
void main() {
  Map<String, dynamic> data = {
    "Name": "Sivam",
    "Age": 43,
    "City": "Salem",
    "Qualification": "Ph.D",
  };
  print(data);

  data.forEach((key, value) {
    print("$key : $value");
  });

  for (var ele in data.entries) {
    print("${ele.key}: ${ele.value}");
  }

  for (var ele in data.values) {
    print(ele);
  }

  print(data.length);

  for (var i = 0; i < data.entries.toList().length; i++) {
    print("${data.entries.toList()[i].key}:${data.entries.toList()[i].value}");
  }
}
```

`Map.from()`

Definition:
Creates a new map by copying all key-value pairs from another map.

Example:

```
var original = {'a': 1, 'b': 2};
var copy = Map.from(original);
print(copy); // {a: 1, b: 2}
```

`Map.of()`

Definition:
Similar to `Map.from()`, but specifically creates a new map from another `Map<K, V>`, preserving key and value types.

Example:

```
var data = {'x': 100, 'y': 200};
var newMap = Map.of(data);
print(newMap); // {x: 100, y: 200}
```

Difference between `Map.from()` and `Map.of()`:

- Functionally very similar.
- `Map.of()` is often used when the input is strongly typed, and it's slightly safer type-wise.
- `Map.from()` works with any `Map<dynamic, dynamic>` and casts.

`Map.fromEntries()`

Definition:

Creates a new map from a list (or iterable) of `MapEntry` objects.

Example:

```
var entries = [MapEntry('id', 1), MapEntry('name', 'Alice')];
var map = Map.fromEntries(entries);
print(map); // {id: 1, name: Alice}
```

`Map.unmodifiable()`

Definition:

Creates an **immutable (read-only)** map. Any attempt to modify it will throw an error.

Example:

```
var data = {'a': 1, 'b': 2};
var fixedMap = Map.unmodifiable(data);

print(fixedMap); // {a: 1, b: 2}

// fixedMap['a'] = 100; //  Throws UnsupportedError
```