

Take this multiple-choice quiz to assess your mastery of core concepts taught in the *Anypoint Platform Development: Fundamentals (Mule 4)* (/instructor-led-training/apdev-fundamentals4), the self-study *MuleSoft.U Development Fundamentals (Mule 4)* (https://training.mulesoft.com/course/mulesoftu-fundamentals4), the *Anypoint Platform Development: Mule 4 for Mule 3 Users* (/instructor-led-training/apdev-mule443), and the self-study *MuleSoft.U Mule 4 for Mule 3 Users* (https://training.mulesoft.com/course/mulesoftu-mule443) courses and to help you prepare for the *MuleSoft Certified Developer - Level 1 (Mule 4)* (https://training.mulesoft.com/exam/mcd-level1) or *MuleSoft Certified Developer - Level 1 (Mule 4) DELTA* (https://training.mulesoft.com/exam/mcd-level1-delta) certification exam.

- 5+ multiple-choice questions for each topic
- Identifies strengths and weaknesses
- Comparable difficulty to the proctored exam

Overview

Writing DataWeave 2.0 Transformations

Quiz

Oh no! You got 2 / 5. Please review your answers.

1.

Refer to the exhibit. An event payload contains an unordered array of flight objects, where every object has a price key and a toAirport key.

What is valid DataWeave code to return flights with price under 500, grouped by toAirport in ascending order, with the lowest price first?

```
[
  {
    "planeType": "Boeing 787",
    "code2": "0001",
    "toAirport": "LAX",
    "takeOffDate": "2015-01-20",
    "fromAirport": "MUA",
    "price": 541,
    "airlineName": "American Airlines",
    "seatsAvailable": "none",
    "code1": "rree"
  },
  {
    "planeType": "Boeing 747",
    "code2": "0123",
    "toAirport": "CLE",
    "takeOffDate": "2015-01-25",
    "fromAirport": "MUA",
    "price": 300,
    "airlineName": "American Airlines",
    "seatsAvailable": "7",
    "code1": "rree"
  }
]
```



```
{
  "CLE": [
    {
      "planeType": "Boeing 737",
      "code2": "1000",
      "toAirport": "CLE",
      "takeOffDate": "2015-01-20",
      "fromAirport": "MUA",
      "price": 200,
      "airlineName": "American Airlines",
      "seatsAvailable": "5",
      "code1": "rree"
    }
  ],
  {
    "planeType": "Boeing 747",
    "code2": "0123",
    "toAirport": "CLE",
    "takeOffDate": "2015-01-25",
    "fromAirport": "MUA",
    "price": 300,
    "airlineName": "American Airlines",
    "seatsAvailable": "7",
    "code1": "rree"
  }
],
  "SFO": [
    {
      "planeType": "Boeing 737",
      "code2": "1093",
      "toAirport": "SFO",
      "takeOffDate": "2015-01-20",
      "fromAirport": "MUA",
      "price": 200,
      "airlineName": "American Airlines",
      "seatsAvailable": "5",
      "code1": "rree"
    }
  ]
}
```

- ☐ payload groupBy \$.toAirport filter \$.price < 500 orderBy \$.price
- ☐ payload groupBy \$.toAirport filter \$.price > 500 orderBy \$.price
- ☒ payload filter \$.price < 500 orderBy \$.price groupBy \$.toAirport
- ☐ payload filter \$.price > 500 orderBy \$.price groupBy \$.toAirport

2.

Refer to the exhibit. What is valid DataWeave code to transform the input JSON payload to the output XML payload?

The screenshot shows the MuleSoft IDE interface. On the left, the file 'list\_unknown.dwl' contains the following JSON payload:

```
[
  {
    "employee1": {
      "firstName": "Aanya",
      "lastName": "Anand",
      "details": {
        "dept": "Engineering",
        "region": "APAC"
      }
    }
  },
  {
    "employee2": {
      "firstName": "Bob",
      "lastName": "Brown",
      "details": {
        "dept": "Marketing",
        "region": "USA"
      }
    }
  }
]
```

On the right, the 'Output Payload' tab shows the resulting XML:

```
<?xml version='1.0' encoding='UTF-8'?>
<employees>
  <employee firstName="Aanya" lastName="Anand"/>
</employees>
```

## Answers

A.

```
%dw 2.0
output application/xml
---
employees :
  employee
    @(
      firstName: payload[0].employee1.firstName[0],
      lastName: payload[0].employee1.lastName[0]
    ) : null
```

B.

```
%dw 2.0
output application/xml
---
employees :
  employee
    (
      firstName: payload[0].employee1.firstName[0],
      lastName: payload[0].employee1.lastName[0]
    ) : ""
```

C.

```
%dw 2.0
output application/xml
---
employees :
  employee
    @(
      firstName: payload[0].employee1.firstName[0];
      lastName: payload[0].employee1.lastName[0]
    ) : null
```

D.

```
%dw 2.0
output application/xml
---
employees :
  employee
  (
    firstName: payload[0].employee1.firstName[0];
    lastName: payload[0].employee1.lastName[0]
  ) : ""
```

- ☒ A
- ☐ B
- ☐ C
- ☐ D

*Sorry, that's incorrect.*

3.

A Mule application has a main flow and a combineNames flow. In the main flow, a variable named fullName is set to the object {firstName: "Max", lastName: "Mule"}.

What is valid DataWeave code to call the combineNames flow with the input object stored in the fullName variable?

- ☐ #[ dw::Flow::lookup( "combineNames", vars.fullName ) ]
- ☐ #[ dw::Runtime::lookup( "combineNames", vars.fullName ) ]
- ☒ #[ lookup( "combineNames", vars.fullName ) ]
- ☐ #[ combineNames( vars.fullName ) ]
- ☐ #[ lookup( combineNames( vars.fullName ) ) ]

*Sorry, that's incorrect.*

4.

What DataWeave 2.0 type can be used as input to a map operation?

- ☒ Array
- ☐ Object
- ☐ Map
- ☐ String
- ☐ Key

*Sorry, that's incorrect.*

5.

What is the correct way to format the decimal 20.3844 as a string to two decimal places?

- ☒ 20.3844 as String {format: ".0#"}
- ☐ 20.3844 as :string {format: ".0#"}
- ☐ 20.3844 as String as format: ".0#"
- ☐ 20.3844 as :string as format: ".0#"
- ☐ 20.3844 as String (format = ".0#")
- ☐ 20.3844 as :string (format = ".0#")

*Correct!*

**Oh no!** You got 2 / 5. Please scroll up to review your answers.

**Next: Triggering Flows**