TGH

Making Integrations Simpler

Flow Control and Scope Operations in DataWeave

Author
Anshul Singh

Contents

Introduction

You can use the following operators within any DataWeave expression:

- `do`
- `if else`
- `else if`

do

A "do" statement allows you to create a specific area where you can declare and use new variables, functions, annotations, or namespaces. It is structured like a map, with a header and a body separated by ---. The header is where you define all the things you want to declare, and the body represents the result of the expression.

This example uses do to return the string "DataWeave" when myfun() is called from the main body of the script.

DataWeave Script:

```
%dw 2.0

output application/json

fun myfun() = do {

    var name = "DataWeave"

    ---

    name

}

---

{ result: myfun() }
```

This example uses do to return the string "DataWeave" when the variable myVar is referenced from the main body of the script.

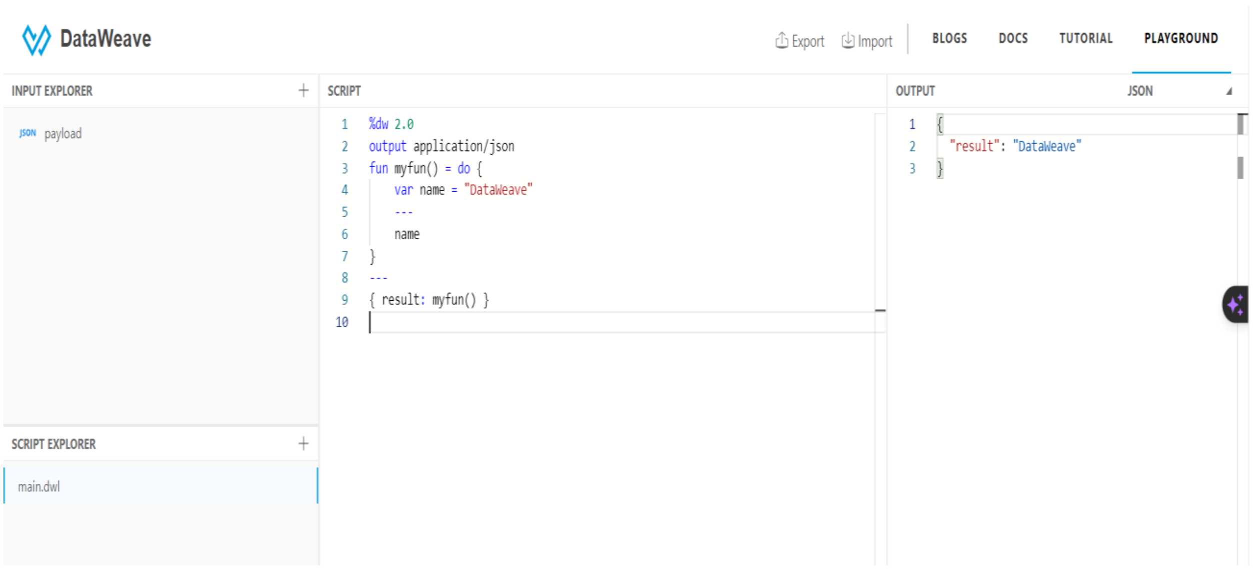
DataWeave Script:

```
%dw 2.0

output application/json

var myVar = do {

    var name = "DataWeave"

    ---

    name

}

---

{ result: myVar }
```

Both scripts produce this output:

Output:

```
{

  "result": "DataWeave"

}
```



```
%dw 2.0
output application/json
fun myfun() = do {
    var name = "DataWeave"
    ---
    name
}
---
{ result: myfun() }
```

```
{
  "result": "DataWeave"
}
```

if else

An "if" statement checks whether a condition is true or false. If the condition is true, it executes the code inside the "if" block. If the condition is false, it executes the code inside the "else" block. Each "if" statement must have a corresponding "else" statement.

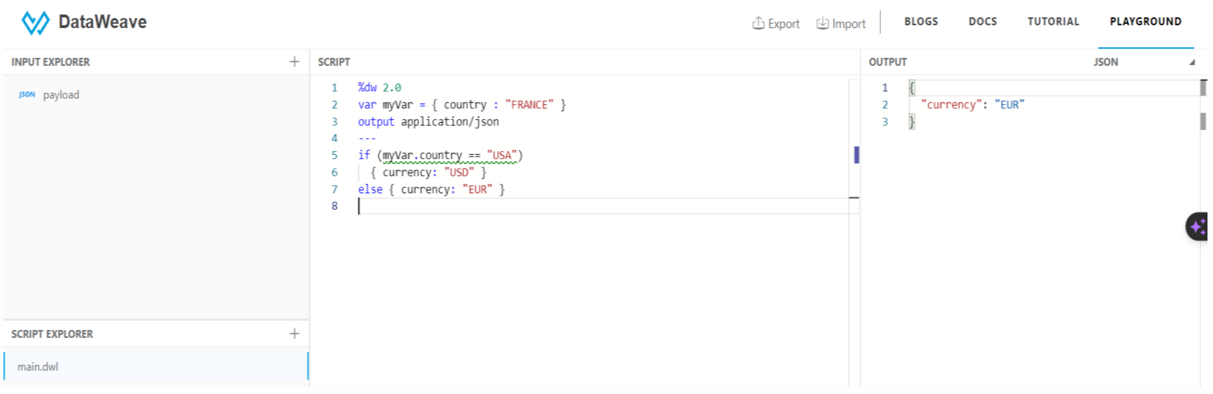
Here's an example that uses the input { country : "FRANCE" }, defined as the variable "myVar" in the header:

DataWeave Script:

```
%dw 2.0

var myVar = { country : "FRANCE" }

output application/json

---

if (myVar.country == "USA")

  { currency: "USD" }

else { currency: "EUR" }
```

Output:

```
{

  "currency": "EUR"

}
```



```
%dw 2.0
var myVar = { country : "FRANCE" }
output application/json
---
if (myVar.country == "USA")
  { currency: "USD" }
else { currency: "EUR" }
```

```
{
  "currency": "EUR"
}
```

You can use the if-else construct with various types of conditions that result in true or false. This includes mathematical, logical, equality, and relational statements. The condition can be applied to any valid input.

The following DataWeave script applies if-else statements to the outcome of certain conditional statements:

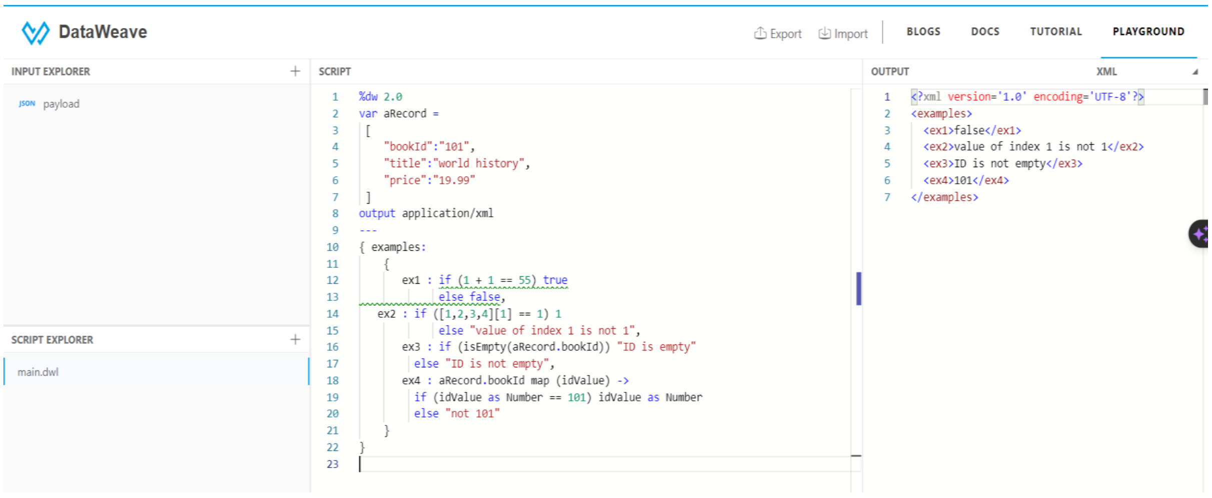
- A mathematical operation in `ex1`
  The `if else` statement returns the Boolean value `true` if the operation `1 + 1 == 55` is true and `false` if not.
- An equality operation in `ex2`
  The `if else` statement returns `1` if the value of the specified index is `1` or a string if the value is not `1`.
- An `isEmpty` function in `ex3`
  The `if else` statement returns the string `"ID is empty"` or `"ID is not empty"` depending on whether `aRecord.bookId` contains a value.
- A mapping in `ex4` that iterates over `firstInput`
  The `if else` statement returns the value of the `bookId` as a Number if the value is equal to `101`. It returns the specified string if the value is not equal to `101`.

DataWeave Script:

```
%dw 2.0


var aRecord =
 [

    "bookId":"101",

    "title":"world history",

    "price":"19.99"

 ]

output application/xml

---

{ examples:

        {

           ex1 : if (1 + 1 == 55) true

                 else false,

           ex2 : if ([1,2,3,4][1] == 1) 1

                 else "value of index 1 is not 1",

           ex3 : if (isEmpty(aRecord.bookId)) "ID is empty"

                  else "ID is not empty",
            ex4 : aRecord.bookId map (idValue) ->

                  if (idValue as Number == 101) idValue as Number

                  else "not 101"

         }

}
```

Output:

```
<?xml version='1.0' encoding='UTF-8'?>

<examples>

  <ex1>false</ex1>

  <ex2>value of index 1 is not 1</ex2>

  <ex3>ID is not empty</ex3>

  <ex4>101</ex4>

</examples>
```



```
%dw 2.0
var aRecord =
 [
    "bookId":"101",
    "title":"world history",
    "price":"19.99"
 ]
output application/xml
---
{ examples:
    {
       ex1 : if (1 + 1 == 55) true
             else false,
   ex2 : if ([1,2,3,4][1] == 1) 1
             else "value of index 1 is not 1",
       ex3 : if (isEmpty(aRecord.bookId)) "ID is empty"
         else "ID is not empty",
       ex4 : aRecord.bookId map (idValue) ->
         if (idValue as Number == 101) idValue as Number
         else "not 101"
    }
}
```

```
<?xml version='1.0' encoding='UTF-8'?>
<examples>
  <ex1>false</ex1>
  <ex2>value of index 1 is not 1</ex2>
  <ex3>ID is not empty</ex3>
  <ex4>101</ex4>
</examples>
```

else if

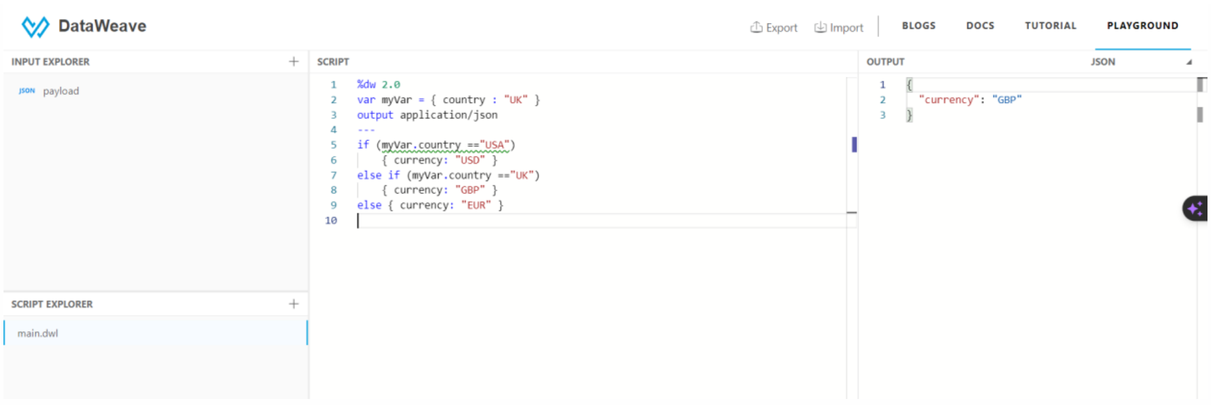
In this example, the "if-else" construct is used to chain multiple "else if" expressions together. The input variable "myVar" is defined as { country: "UK" }. This structure allows for conditional checks on the value of the "country" field. If the condition matches the value "USA," a specific code block is executed. If none of the conditions are met, the code in the final "else" block is executed.

DataWeave Script:

```
%dw 2.0

var myVar = { country : "UK" }

output application/json

---

if (myVar.country =="USA")

        { currency: "USD" }

else if (myVar.country =="UK")

        { currency: "GBP" }

else { currency: "EUR" }
```

Output

```
{
  "currency": "GBP"
}
```



```
%dw 2.0
var myVar = { country : "UK" }
output application/json
---
if (myVar.country =="USA")
    { currency: "USD" }
else if (myVar.country =="UK")
    { currency: "GBP" }
else { currency: "EUR" }
```

```
{
 "currency": "GBP"
}
```

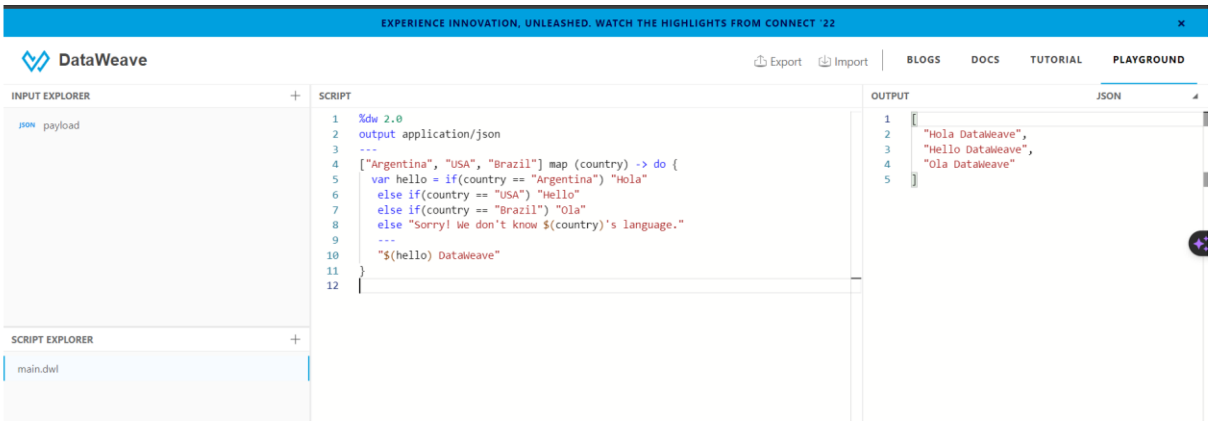
The following example is similar but takes an array as input instead of an object. The body of the script uses `if else` and `else if` statements within a `do` operation to populate the value of the `hello` variable.

DataWeave Script:

```
%dw 2.0

output application/json

---

["Argentina", "USA", "Brazil"] map (country) -> do {

  var hello = if(country == "Argentina") "Hola"

   else if(country == "USA") "Hello"

   else if(country == "Brazil") "Ola"

   else "Sorry! We don't know $(country)'s language."

   ---

   "$(hello) DataWeave"

}
```

Output:

```
[

  "Hola DataWeave",

  "Hello DataWeave",

  "Ola DataWeave"

]
```



```
%dw 2.0
output application/json
---
["Argentina", "USA", "Brazil"] map (country) -> do {
  var hello = if(country == "Argentina") "Hola"
   else if(country == "USA") "Hello"
   else if(country == "Brazil") "Ola"
   else "Sorry! We don't know $(country)'s language."
   ---
   "$(hello) DataWeave"
}
```

```
[
  "Hola DataWeave",
  "Hello DataWeave",
  "Ola DataWeave"
]
```

Reference

- https://docs.mulesoft.com/
- https://dataweave.mulesoft.com/
- https://www.youtube.com/

TGH

Making Integrations Simpler

TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.

Email address

connect@techygeekhub.com

Phone number

+ 011-40071137

+ 91-8810610395

Our offices

Noida Office

iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office

Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081