

Collection variables

GeneXus[™]

At the end of the previous video, we talked about the concept of collection variable.

Collections:

Numeric collection

1	5	8	7	3
---	---	---	---	---

Character collection

Ann	Paul	Susan	Peter
-----	------	-------	-------

SDT collection

1	2	3	4	5
Uruguay	France	United States	Italy	Argentina
Montevideo	Paris	Washington	Rome	Buenos Aires

In every GeneXus object, we can define variables based on simple or structured data types that allow us to store the data of an entity, such as numerical values, texts, or the details of a country.

Collections:

&names



Name	Type	Is Collection	Description
& Variables			
& Standard Variables			
names	Character(20)	<input checked="" type="checkbox"/>	names

&numbers

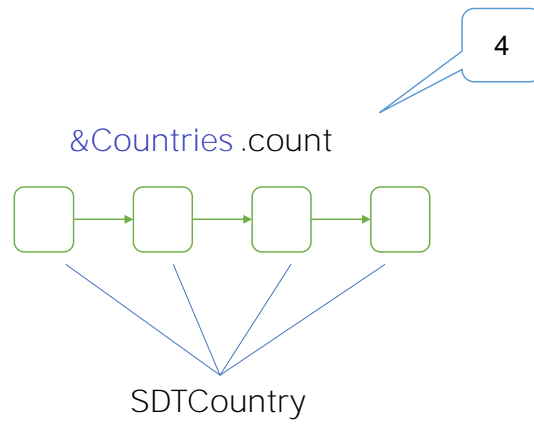


Name	Type	Is Collection	Description
& Variables			
& Standard Variables			
numbers	Numeric(4.0)	<input checked="" type="checkbox"/>	numbers

We have seen that when defining a variable, we can select the “IsCollection” check box to set this variable as a collection.

Collection variables, regardless if they are numeric, character, or SDT collections, will offer their own methods for managing that collection.

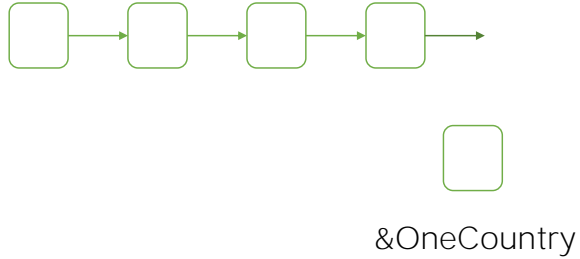
To count the number of elements:



The Count method allows us to know the number of elements in the collection. It returns that number, so it is a read-only method.

To add an element to the collection:

```
&Countries .add(&OneCountry)
```



The Add method allows us to add an element to the collection. By default this element is added at the end of the collection, but it is possible to change its position.

To add an element to the collection:

Example: Loading a collection of countries.

```

For each Country
    &CountryItem = New()
    &CountryItem.Id = CountryId
    &CountryItem.Name = CountryName
    &Countries.Add(&CountryItem)
endfor

```

We are reinitializing the SDT variable, and that is why we are using the New operator.

Name	Type	Is Collection
Variables		
Standard Variables		
Countries	SDTCountry	<input checked="" type="checkbox"/>
CountryItem	SDTCountry	<input type="checkbox"/>

Name	Type	Description
SDTCountry		SDTCountry
Id	Numeric(4.0)	Id
Name	Character(20)	Name

For example, if we want to load the collection of countries in the database, how do we do it? Even though in the next video we will see the simple way to load data using a Data Provider, let's see this other option now.

When variables are declared, the program already assigns them a memory space, but we can always request another memory space for the variables so as not to use them twice.

That is, we are re-initializing the SDT variable, and **that's** why we are using the New operator.

In this case it is necessary because otherwise we will always be overwriting that memory space. So if we don't ask for new space, we will always be adding the same item to the collection, and we will have a collection of n items with the same values. The reason is that the &countryItem variable is not restarted after the Add action, as you might think.

To remove an element from the collection:

```
&Countries .remove(3)
```



The Remove method allows you to remove an item from the collection from its index. The range of the indexes starts at 1 and ends with the value returned by the Count method..

To delete all elements in the collection:

```
&Countries .clear()
```



The Clear method allows you to remove all items from the collection..

To sort the items in the collection:

```
&Countries .sort("Name")
```



Name	Type	Description
SDTCountry		SDTCountry
• Id	Numeric(4,0)	Id
• Name	Character(20)	Name

```

For &CountryItem in &Countries
  Print Countries
Endfor
  
```

The Sort method allows you to sort the collection. If the elements are of SDT type, then we can have the collection ordered by a member of that structure; for example, by country name.

Then, to run through a collection we have the command For item in collection.

As this command runs through the collection, it loads each item to the &countryItem variable. This way we can print it on the output.

To sort the items in the collection:

Example: Loading and ordering a collection of countries

```

For each Country
  &CountryItem = New()
  &CountryItem.Id = CountryId
  &CountryItem.Name = CountryName
  &Countries.Add(&CountryItem)
endfor

&Countries.Sort("Name")
For &CountryItem in &Countries
  print Countries
Endfor

```

Name	Type	Description
SDTCountry		SDTCountry
• Id	Numeric(4,0)	Id
• Name	Character(20)	Name

Name	Type	Is Collection
Variables		
• Standard Variables		
• Countries	SDTCountry	<input checked="" type="checkbox"/>
• CountryItem	SDTCountry	<input type="checkbox"/>

Let's see then how the procedure finally looks...

*GeneXus*TM

training.genexus.com
wiki.genexus.com