

# EUPHORIA - PROCEDURES

[http://www.tutorialspoint.com/euphoria/euphoria\\_procedures.htm](http://www.tutorialspoint.com/euphoria/euphoria_procedures.htm)

Copyright © tutorialspoint.com

A procedure is a group of reusable code which can be called anywhere in your programme. This eliminates the need of writing same code again and again. This will help programmers to write modular code.

Like any other advance programming language, Euphoria also supports all the features necessary to write modular code using procedures. You would see detail on function in subsequent chapter.

You must have seen procedures like *printf* and *length* in previous chapters. We are using these procedure again and again but they have been written in core Euphoria only once.

Euphoria allows us to write our own procedures as well. This section will explain you how to write your own procedure in Euphoria.

## Procedure Definition

Before we use a procedure we need to define that procedure. The most common way to define a procedure in Euphoria is by using the **procedure** keyword, followed by a unique procedure name, a list of parameters *thatmightbeempty*, and a statement block which ends with **end procedure** statement. The basic syntax is shown here –

```
procedure procedurename(parameter-list)
    statements
    .....
end procedure
```

## Example

A simple procedure that takes no parameters called sayHello is defined here –

```
procedure sayHello()
    puts(1, "Hello there")
end procedure
```

## Calling a Procedure

To invoke a procedure somewhere later in the script, you would simple need to write the name of that procedure as follows –

```
#!/home/euphoria-4.0b2/bin/eui

procedure sayHello()
    puts(1, "Hello there")
end procedure

-- Call above defined procedure.
sayHello()
```

This would produce following result –

```
Hello there
```

## Procedure Parameters

Till now we have seen procedure without a parameter. But there is a facility to pass different parameters while calling a procedure. These passed parameters can be captured inside the procedure and any manipulation can be done over those parameters.

A procedure can take multiple parameters separated by comma.

### Example:

Let us do a bit modification in our *sayHello* procedure. This time it will take two parameters –

```
#!/home/euphoria-4.0b2/bin/eui  
  
procedure sayHello(sequence name,atom age)  
  printf(1, "%s is %d years old.", {name, age})  
end procedure  
  
-- Call above defined procedure.  
sayHello("zara", 8)
```

This would produce following result –

```
zara is 8 years old.
```

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js