

EUPHORIA - SWITCH STATEMENTS

http://www.tutorialspoint.com/euphoria/euphoria_switch_statement.htm

Copyright © tutorialspoint.com

The switch Statement

The **switch** statement is used to run a specific set of statements, depending on the value of an expression. It often replaces a set of **if-elsif** statements giving you more control and readability of your program.

Syntax

The syntax of simple switch statement is –

```
switch expression do
  case <val> [, <val-1>....] then
    -- Executes when the expression matches one of the values
  case <val> [, <val-1>....] then
    -- Executes when the expression matches one of the values
    .....
  case else
    -- Executes when the expression does not matches any case.
end if
```

The <val> in a case must be either an atom, literal string, constant or enum. Multiple values for a single case can be specified by separating the values by commas. By default, control flows to the end of the switch block when the next case is encountered.

Example

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

atom marks = 'C'

switch marks do
  case 'A' then
    puts(1, "Excellent!\n" )
  case 'B', 'C' then
    puts(1, "Well done!\n" )
  case 'D' then
    puts(1, "You passed!\n" )
  case 'F' then
    puts(1, "Better try again!\n" )
  case else
    puts(1, "Invalid grade!\n" )
end switch
```

This would produce following result –

```
Well done!
```

The switch...with fallthru Statement

The **case** statement of a **switch** is executed when it matches with the given expression value and by default it comes out. By default, control flows to the end of the switch block when the next case is encountered.

The default for a particular switch block can be changed so that control passes to the next executable statement whenever a new case is encountered by using **with fallthru** in the switch statement:

Syntax

The syntax of simple *switch...with fallthru* statement is –

```

switch expression with fallthru do
  case <val> [, <val-1>....] then
    -- Executes when the expression matches one of the values
    break -- optional to come out of the switch from this point.
  case <val> [, <val-1>....] then
    -- Executes when the expression matches one of the values
    break -- Optional to come out of the switch from this point.
    .....
  case else
    -- Executes when the expression does not matches any case.
    break -- Optional to come out of the switch from this point.
end if

```

Example

```

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

atom marks = 'C'

switch marks with fallthru do
  case 'A' then
    puts(1, "Excellent!\n" )
  case 'B', 'C' then
    puts(1, "Well done!\n" )
  case 'D' then
    puts(1, "You passed!\n" )
  case 'F' then
    puts(1, "Better try again!\n" )
  case else
    puts(1, "Invalid grade!\n" )
end switch

```

This would produce following result –

```

Well done!
You passed!
Better try again!
Invalid grade!

```

You can use optional **break** statement to come out from a point inside a switch statement as follows –

```

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

atom marks = 'C'

switch marks with fallthru do
  case 'A' then
    puts(1, "Excellent!\n" )
    break
  case 'B', 'C' then
    puts(1, "Well done!\n" )
    break
  case 'D' then
    puts(1, "You passed!\n" )
    break
  case 'F' then
    puts(1, "Better try again!\n" )
    break
  case else
    puts(1, "Invalid grade!\n" )
    break
end switch

```

This would produce following result –

```
Well done!
```

The switch....label Statement

The **switch** statement can have an optional **label** to name the switch block. This name can be used in nested switch break statements to break out of an enclosing switch rather than just the owning switch.

A switch label is used just to name the block and label names must be double quoted constant strings having single or multiple words. The label keyword is a case sensitive and should be written as **label**.

Syntax

The syntax of simple *switch...label* statement is –

```
switch expression label "Label Name" do
  case <val> [, <val-1>....] then
    -- Executes when the expression matches one of the values
    break "LEBEL NAME"
  case <val> [, <val-1>....] then
    -- Executes when the expression matches one of the values
    break "LEBEL NAME"
    .....
  case else
    -- Executes when the expression does not matches any case.
    break "LEBEL NAME"
end if
```

Example

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

atom marks = 'C'
atom scale = 'L'

switch marks label "MARKS" do
  case 'A' then
    puts(1, "Excellent!\n" )
  case 'B', 'C' then
    puts(1, "Well done!\n" )
    switch scale label "SCALE" do
      case 'U' then
        puts(1, "Upper scale!\n" )
        break "MARKS"
      case 'L' then
        puts(1, "Lower scale!\n" )
        break "MARKS"
      case else
        puts(1, "Invalid scale!\n" )
        break "MARKS"
      end switch
  case 'D' then
    puts(1, "You passed!\n" )
  case 'F' then
    puts(1, "Better try again!\n" )
  case else
    puts(1, "Invalid grade!\n" )
  end switch
end switch
```

This would produce following result –

```
Well done!
Lower scale!
```

Note: If you are not using a *with fallthru* statement then you do not need to use a lable because switch statement would come out automatically.