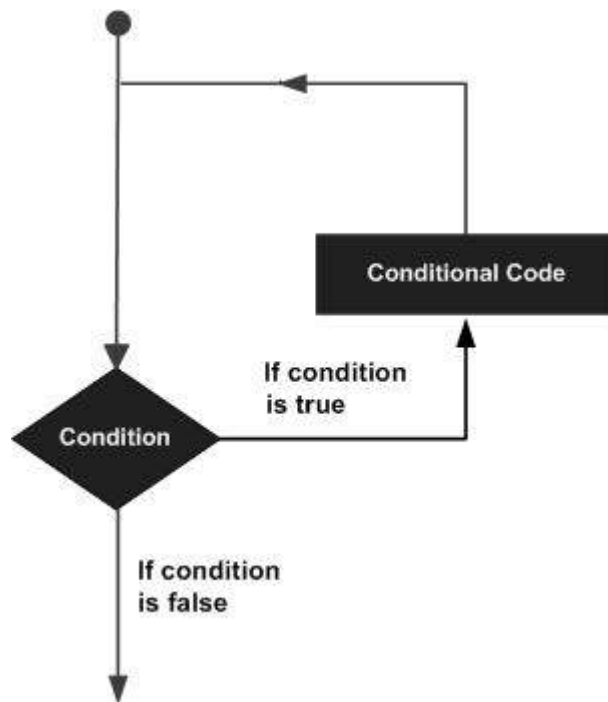


LISP - LOOPS

http://www.tutorialspoint.com/lisp/lisp_loops.htm

Copyright © tutorialspoint.com

There may be a situation, when you need to execute a block of code numbers of times. A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages.



LISP provides the following types of constructs to handle looping requirements. Click the following links to check their detail.

Construct	Description
loop	The loop construct is the simplest form of iteration provided by LISP. In its simplest form, it allows you to execute some statements repeatedly until it finds a return statement.
loop for	The loop for construct allows you to implement a for-loop like iteration as most common in other languages.
do	The do construct is also used for performing iteration using LISP. It provides a structured form of iteration.
dotimes	The dotimes construct allows looping for some fixed number of iterations.
dolist	The dolist construct allows iteration through each element of a list.

Gracefully Exiting From a Block

The **block** and **return-from** allows you to exit gracefully from any nested blocks in case of any error.

The **block** function allows you to create a named block with a body composed of zero or more statements. Syntax is:

```
(block block-name(  
...  
...  
))
```

The **return-from** function takes a block name and an optional *thedefaultisnil* return value.

The following example demonstrates this:

Example

Create a new source code file named main.lisp and type the following code in it:

```
(defun demo-function (flag)  
  (print 'entering-outer-block)  
  
  (block outer-block  
    (print 'entering-inner-block)  
    (print (block inner-block  
      (if flag  
        (return-from outer-block 3)  
        (return-from inner-block 5)  
      )  
      (print 'This-wil--not-be-printed))  
    )  
    (print 'left-inner-block)  
    (print 'leaving-outer-block)  
  t)  
)  
(demo-function t)  
(terpri)  
(demo-function nil)
```

When you click the Execute button, or type Ctrl+E, LISP executes it immediately and the result returned is:

```
ENTERING-OUTER-BLOCK  
ENTERING-INNER-BLOCK  
  
ENTERING-OUTER-BLOCK  
ENTERING-INNER-BLOCK  
5  
LEFT-INNER-BLOCK  
LEAVING-OUTER-BLOCK
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js