LISP - DATA TYPES

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

In LISP, variables are not typed, but data objects are.

LISP data types can be categorized as.

- **Scalar types** - for example, number types, characters, symbols etc.
- **Data structures** - for example, lists, vectors, bit-vectors, and strings.

Any variable can take any LISP object as its value, unless you have declared it explicitly.

Although, it is not necessary to specify a data type for a LISP variable, however, it helps in certain loop expansions, in method declarations and some other situations that we will discuss in later chapters.

The data types are arranged into a hierarchy. A data type is a set of LISP objects and many objects may belong to one such set.

The **typep** predicate is used for finding whether an object belongs to a specific type.

The **type-of** function returns the data type of a given object.

Type Specifiers in LISP

Type specifiers are system-defined symbols for data types.

array
atom
bignum
bit
bit-vector
character
[common]
compiled-function
complex
cons
double-float
fixnum
float
function
hash-table
integer
keyword
list
long-float
nill
null
number
package
pathname
random-state
ratio
rational
readtable
sequence
short-float
signed-byte
simple-array
simple-bit-vector
simple-string
simple-vector
single-float
standard-char
stream
string
[string-char]
symbol
t
unsigned-byte
vector

Apart from these system-defined types, you can create your own data types. When a structure type is defined using **defstruct** function, the name of the structure type becomes a valid type symbol.

Example 1

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

```
(setq x 10)
(setq y 34.567)
(setq ch nil)
(setq n 123.78)
(setq bg 11.0e+4)
```

```
(setq r 124/2)

(print x)
(print y)
(print n)
(print ch)
(print bg)
(print r)
```

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

```
10
34.567
123.78
NIL
110000.0
62
```

Example 2

Next let's check the types of the variables used in the previous example. Create new source code file named main. lisp and type the following code in it.

```
(setq x 10)
(setq y 34.567)
(setq ch nil)
(setq n 123.78)
(setq bg 11.0e+4)
(setq r 124/2)

(print (type-of x))
(print (type-of y))
(print (type-of n))
(print (type-of ch))
(print (type-of bg))
(print (type-of r))
```

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

```
(INTEGER 0 281474976710655)
SINGLE-FLOAT
SINGLE-FLOAT
NULL
SINGLE-FLOAT
(INTEGER 0 281474976710655)
```