

## How to use SWI-Prolog on elvis at Rowan University

- 1) **Create your Prolog program file** using your favorite text editor (emacs, vi, pico, etc.). When you save the file, you can choose any file name that you like, since SWI-Prolog doesn't look for any special file extension, but your life will be easiest if you select a file name that follows the rules for Prolog symbolic constants: a file name that begins with a lower case letter and includes only letters, digits, and underscore (\_) symbols.

- 2) **Start the SWI-Prolog interpreter:** At the elvis prompt, type

```
p1
```

- 3) **Load your program into the interpreter:** Suppose you have named your Prolog homework file `homework`. At the interpreter prompt (`?-`), you have a choice of two commands to load the program. One is `consult`:

```
?- consult(homework).
```

The other is to list the file name in square brackets:

```
?- [homework].
```

If your program contains several files (for example, `homework`, `library`, and `utility`), then you can either list all the file names in a set of square brackets:

```
?- [homework, library, utility].
```

or you can use three `consult` statements in a row:

```
?- consult(homework).  
?- consult(library).  
?- consult(utility).
```

It doesn't matter what order you list the file names in when you do this.

If your file name starts with a capital letter or contains characters other than letters, digits, and underscores, you should surround the name with single quotes when you load it in. For example, if you wanted to use the full path name of the file, this path name would include some punctuation characters, and the file could be loaded with a command like this:

```
?- ['~nlt/proglang/prologfiles/homework'].
```

or this:

```
?- consult('~nlt/proglang/prologfiles/homework').
```

- 4) **Run the program:** To run the program, give the interpreter goals to prove, one at a time.
- 5) **Debugging:** If you need to debug your program, the standard Prolog debugger is described in chapter 8 of Clocksin & Mellish, *Programming in Prolog*, 5th edition (Springer 2003). The most important predicates to read about are:

<code>?-trace.</code>	<i>Turns on exhaustive tracing</i>
<code>?-notrace.</code>	<i>Turns off exhaustive tracing</i>
<code>?-spy(PREDICATE).</code>	<i>Sets a spy point for PREDICATE</i>
<code>?-nospy(PREDICATE).</code>	<i>Turns off the spy point for PREDICATE</i>
<code>?-debug.</code>	<i>Turns on debugging that stops at spy points</i>
<code>?-nodebug.</code>	<i>Turns off debugging that stops at spy points</i>

- 6) **Exit SWI-Prolog:** To exit the SWI-Prolog interpreter, type `control-D` or `halt`.

### Sample Prolog program

```
cat(fluffy).
cat(pandora).
lion(leo).
feline(X) :- cat(X).
feline(X) :- lion(X).
furry(X) :- feline(X).
```

Assume this program has been created and saved under the name *felines*.

### Sample Prolog session

```
elvis 1% pl
Welcome to SWI-Prolog (Multi-threaded, Version 5.6.38)
Copyright (c) 1990-2007 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.
```

For help, use `?- help(Topic).` or `?- apropos(Word).`

```
?- [felines].
% felines compiled 0.00 sec, 1,344 bytes
```

```
Yes
?- lion(X).
```

```
X = leo
```

```
Yes
?- cat(fido).
```

```
No
?- cat(pandora).
```

```
Yes
?- feline(X).
```

```
X = fluffy ;
```

```
X = pandora ;
```

```
X = leo ;
```

```
No
?- trace.
```

```
Yes
[trace] ?- furry(fluffy).
Call: (7) furry(fluffy) ? creep
Call: (8) feline(fluffy) ? creep
Call: (9) cat(fluffy) ? creep
Exit: (9) cat(fluffy) ? creep
Exit: (8) feline(fluffy) ? creep
```

Exit: (7) furry(fluffy) ? creep

Yes

[debug] ?- spy(cat).  
% Spy point on cat/1

Yes

[debug] ?- debug.

Yes

[debug] ?- furry(fluffy).  
Call: (9) cat(fluffy) ? creep  
Exit: (9) cat(fluffy) ? creep  
Exit: (8) feline(fluffy) ? creep  
Exit: (7) furry(fluffy) ? creep

Yes

[debug] ?- nodebug.

Yes

?- furry(fluffy).

Yes

?- ^D  
elvis 2%