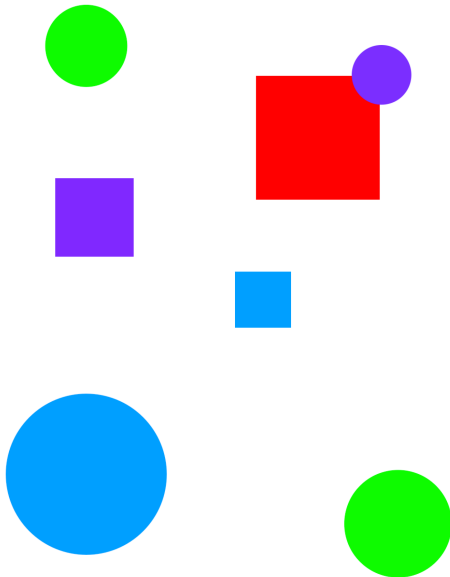

Lesson 4: Example Program - Free Floating Shapes World

What's It All About?

1. A knowledge base about 2 dimensional shapes is presented as an image, without many specifics.
2. From the image, a KB is imaginatively rendered in English.
3. The KB is translated to Prolog.
4. A Prolog demo is presented.
5. The demo is annotated.
6. Big picture, this lesson is merely about gaining familiarity with knowledge representation in Prolog.

Image



KB in English

From the image, using considerable imagination, I came up with the following KB, expressed a regular sort of English.

FACTS ABOUT SQUARES

sera is a purple square with side 7.
sara is a blue square with side 5.

sarah is a red square with side 11.

FACTS ABOUT CIRCLES

carla is a green circle with radius 4.
cora is a blue circle with radius 7.
connie is a purple circle with radius 3.
claire is a green circle with radius 5.

RULES FOR LISTING CIRCLES AND SQUARES AND SHAPES

if you sequentially, exhaustively, grab a circle and write its name
then you will have written the names of all the circles.

if you sequentially, exhaustively, grab a square and write its name
then you will have written the names of all the squares.

if you write the names of all the circles and all the squares
then you will have written the names of all the shapes.

RULES FOR FINDING SPECIAL SHAPES

if a square is blue
then its name represents a blue shape.
if a circle is blue
then its name represents a blue shape.

if the area of a shape is at least 100
then it is a large shape

if the area of a shape is less than 100
then it is a small shape

UTILITIES

if a circle has radius R, and A is the area determined by R
then it is a shape of area A.
if a square has side S, and S is the area determined by S
then it is a shape of area A.

KB in Prolog

```
% -----  
% -----  
% --- File: shapes_world_1.pro  
% --- Line: Loosely represented 2-D shapes world (simple take on SHRDLU)  
% -----  
  
% -----  
% --- Facts ...  
% -----
```

```

% -----
% --- square(N,side(L),color(C)) :: N is the name of a square with side L
% --- and color C

square(sera,side(7),color(purple)).
square(sara,side(5),color(blue)).
square(sarah,side(11),color(red)).

% -----
% --- circle(N,radius(R),color(C)) :: N is the name of a circle with
% --- radius R and color C

circle(carla,radius(4),color(green)).
circle(cora,radius(7),color(blue)).
circle(connie,radius(3),color(purple)).
circle(claire,radius(5),color(green)).

% -----
% Rules ...
% -----

% -----
% --- circles :: list the names of all of the circles

circles :- circle(Name,_,_), write(Name),nl,fail.
circles.

% -----
% --- squares :: list the names of all of the squares

squares :- square(Name,_,_), write(Name),nl,fail.
squares.

% -----
% --- squares :: list the names of all of the shapes

shapes :- circles,squares.

% -----
% --- blue(Name) :: Name is a blue shape

blue(Name) :- square(Name,_,color(blue)).
blue(Name) :- circle(Name,_,color(blue)).

% -----
% --- large(Name) :: Name is a large shape

large(Name) :- area(Name,A), A >= 100.

% -----
% --- small(Name) :: Name is a small shape

small(Name) :- area(Name,A), A < 100.

```

```
% -----  
% --- area(Name,A) :: A is the area of the shape with name Name  
  
area(Name,A) :- circle(Name,radius(R),_), A is 3.14 * R * R.  
area(Name,A) :- square(Name,side(S),_), A is S * S.
```

Demo

```
bash-3.2$ swipl  
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.2.5)  
Copyright (c) 1990-2012 University of Amsterdam, VU 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).
```

```
?- consult('shapes_world_1.pro').  
% shapes_world_1.pro compiled 0.00 sec, 19 clauses  
true.
```

```
?- listing(squares).  
squares :-  
square(A, _, _),  
write(A),  
nl,  
fail.  
squares.
```

```
true.
```

```
?- squares.  
sera  
sara  
sarah  
true.
```

```
?- listing(circles).  
circles :-  
circle(A, _, _),  
write(A),  
nl,  
fail.  
circles.
```

```
true.
```

```
?- circles.  
carla  
cora  
connie  
claire
```

```
true.

?- listing(shapes).
shapes :-
circles,
squares.

true.

?- shapes.
carla
cora
connie
claire
sera
sara
sarah
true.

?- blue(Shape).
Shape = sara ;
Shape = cora ;
false.

?- large(Name),write(Name),nl,fail.
cora
sarah
false.

?- small(Name),write(Name),nl,fail.
carla
connie
claire
sera
sara
false.

?- area(cora,A).
A = 153.86

?- area(carla,A).
A = 50.24

?- halt.
bash-3.2$
```

Annotated Demo

By post editing the demo, I came up with this ...

```
bash-3.2$ swipl
```

Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.2.5)
Copyright (c) 1990-2012 University of Amsterdam, VU 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).`

```
?- consult('shapes_world_1.pro').  
% shapes_world_1.pro compiled 0.00 sec, 19 clauses  
true.
```

```
*- LIST THE PART OF THE KB PERTAINING TO THE squares RELATION.
```

```
?- listing(squares).
```

```
squares :-  
square(A, _, _),  
write(A),  
nl,  
fail.  
squares.
```

```
true.
```

```
*- LIST THE NAMES OF ALL THE SQUARES.
```

```
?- squares.
```

```
sera  
sara  
sarah  
true.
```

```
*- LIST THE PART OF THE KB PERTAINING TO THE circles RELATION.
```

```
?- listing(circles).
```

```
circles :-  
circle(A, _, _),  
write(A),  
nl,  
fail.  
circles.
```

```
true.
```

```
*- LIST THE NAMES OF ALL THE CIRCLES.
```

```
?- circles.
```

```
carla  
cora  
connie  
claire  
true.
```

```
*- LIST THE PART OF THE KB PERTAINING TO THE shapes RELATION.
```

```
?- listing(shapes).
```

```
shapes :-  
circles,  
squares.
```

```
true.
```

```
*- LIST THE NAMES OF ALL THE SHAPES.
```

```
?- shapes.
```

```
carla  
cora  
connie  
claire  
sera  
sara  
sarah  
true.
```

```
*- FIND A BLUE SHAPE.
```

```
?- blue(Shape).
```

```
Shape = sara ;  
Shape = cora ;  
false.
```

```
*- LIST THE NAMES OF ALL THE LARGE SHAPES.
```

```
?- large(Name),write(Name),nl,fail.
```

```
cora  
sarah  
false.
```

```
*- LIST THE NAMES OF ALL THE SMALL SHAPES.
```

```
?- small(Name),write(Name),nl,fail.
```

```
carla  
connie  
claire  
sera  
sara  
false.
```

```
*- WHAT IS SHAPE cora'S AREA?
```

```
?- area(cora,A).
```

```
A = 153.86
```

```
*- WHAT IS SHAPE carla'S AREA?
```

```
?- area(carla,A).
```

```
A = 50.24
```

```
*- TERMINATE EXECUTION
```

```
?- halt.
```

```
bash-3.2$
```