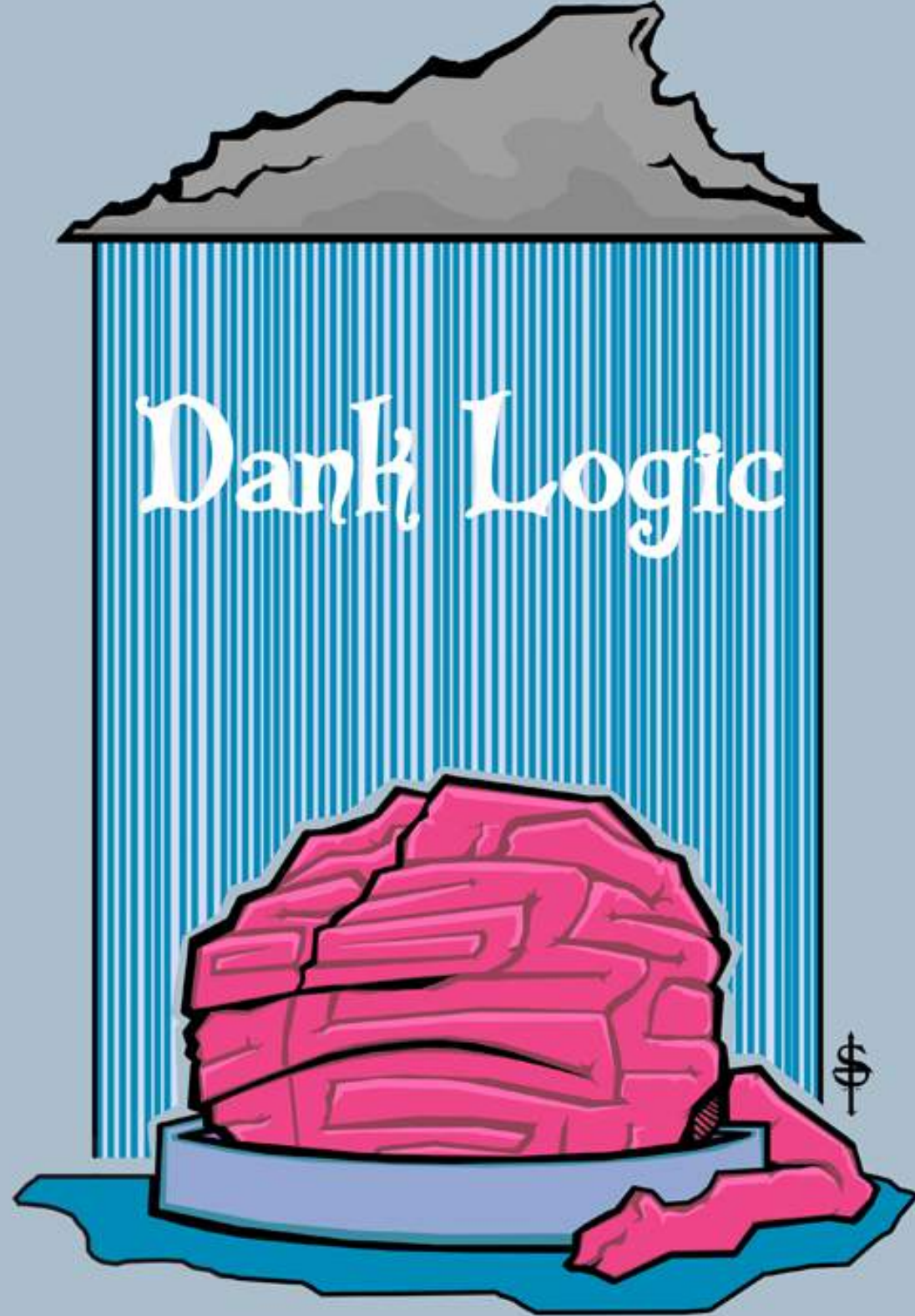


# ***Prácticas de Lógica Computacional 2018/2019***

Sesión 2

Semana 24 al 28  
de septiembre de 2018

Dank Logic  
by \*saintpepsi





# SWI Prolog

# Repasando

## Editor de textos

Find View Goto Tools Project Preferences Help

sesion1\_ejemplo\_nomansky.pl ruta\_nivel1.txt

```
1 %%%%%%%%%%%
2 %% HECHOS
3 %%%%%%%%%%%
4
5 estrella(sol).
6 estrella(p_centauri).
7
8 planeta(tierra).
9 planeta(marte).
10 planeta(phobos).
11 planeta(proxima_b).
```

## Intérprete

```
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)
Copyright (c) 1990-2015 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).

?- estrella(sol).
true.

?-
```

Preguntas

Respuestas

Salvar

Base de  
Conocimientos



Consultar

USUARIO



# *Repasando comandos básicos*

**Ejemplos:** <http://bit.ly/demosprolog>

## TERMINAL

Lanzar intérprete	<code>\$ swipl</code>
Lanzar y consultar	<code>\$ swipl -f fichero.pl</code>
Lanzar editor	<code>\$ gedit</code>

## INTÉRPRETE

Consultar Base	<code>?- consult('fichero.pl').</code>
Preguntar	<code>?- pregunta.</code>
Salir del intérprete	<code>?- halt.</code>

# *El mundo de PLMan*

Paredes  
#



PLMan  
@



# *El mundo de PLMan*

## PERSONAJES y OBJETOS

- \* PLMan @
- \* Cocos .
- \* Fantasmas F
- \* Enemigos E
- \* Llaves a
- \* Puertas ||
- \* Minas +
- ...

## ACCIONES

- \* Comer
- \* Ver
- \* Coger
- \* Dejar
- \* Usar

```
#####
# . . . . . # . . . . . E# . . . . . #
# . ## . # || . . . . . E . . . . . # # F# . # F# . #
#a# . . . # . # E . . . . . # . # . # F# . # . #
# . . . . @# . . . . . E . + . . . . . #
#####
```



# ***El mundo de PLMan***

\* Escribe tres **proposiciones (frases enunciativas)** que expresen conocimiento en el mundo de PLMan





# Web de Prácticas

<http://logica.i3a.ua.es>

The screenshot shows a web browser window with the address bar displaying [logica.i3a.ua.es](http://logica.i3a.ua.es). The website is built on Joomla! and features a dark header with the Joomla! logo and navigation icons. The main content area is divided into two columns. The left column contains a 'MENÚ PRINCIPAL' with links to 'Inicio' and 'Entrega de prácticas', and an 'ACCESO' section with login fields for 'Nombre de usuario' and 'Contraseña', a 'Recordarme' checkbox, and a 'INICIAR SESIÓN' button. Below the login fields are links for '¿Olvidó su contraseña?' and '¿Olvidó su nombre de usuario?'. The right column displays a welcome message from Francisco Gallego, followed by a 'Descargar PL-MAN' button (version 07/09/2018). Below this are two sections: 'Normas básicas de uso de la web, descarga y entrega de mapas' by Carlos Villegra and 'Instrucciones para la instalación de SWI-Prolog en Linux o MAC' by Carlos Villegra. Each section includes a 'Documento' link and a 'LAST\_UPDATED2' timestamp. At the bottom, there is a section for a 'Script para ejecutar N veces plman' by Francisco Gallego, which includes instructions on how to use the script and a 'LAST\_UPDATED2' timestamp.

Web de Prácticas de Matemáticas X +

← → ↻ 🏠

logica.i3a.ua.es

⋮ 🔔 ⭐ 🔍 Buscar

🔍 🔍 🔍

🔍 🔍 🔍

**Joomla!**  
...because open source matters

**MENÚ PRINCIPAL**

- Inicio
- Entrega de prácticas

**ACCESO**

Nombre de usuario

Contraseña

Recordarme

INICIAR SESIÓN

• ¿Olvidó su contraseña?

• ¿Olvidó su nombre de usuario?

Bienvenido a la web de prácticas de PL-Man

Francisco Gallego

En esta web podrás entregar y seguir el progreso de las prácticas en las asignaturas en las que trabajes con PL-Man en el curso 2018/19. Sigue las instrucciones de los profesores y disfruta aprendiendo con las prácticas en lenguaje Prolog 😊.

Esta web, y todo su contenido, está construida con el esfuerzo de los profesores de dichas asignaturas y con la intención de que sea lo más útil posible. Sin embargo, es muy posible que puedas encontrar problemas mientras la utilizas. Si es así, te agradeceremos enormemente que nos escribas comentándonos los problemas que hayas encontrado. Con tu ayuda haremos que esta web sea un espacio cada vez mejor para vosotros.

**Descargar PL-MAN**  
(versión 07/09/2018)

LAST\_UPDATED2

**Normas básicas de uso de la web, descarga y entrega de mapas**

Carlos Villegra

**Normas** básicas de uso, descarga y entrega de prácticas a través de esta web.

Si detectas cualquier problema comunicadlo a vuestro profesor de prácticas.

LAST\_UPDATED2

**Script para ejecutar N veces plman**

Francisco Gallego

Aquí tenéis un **script** que os permite ejecutar las veces que queráis una misma solución a un mapa de plman con 1 solo comando. El script ejecuta plman con una solución N veces, indicando el resultado de cada una de las ejecuciones realizadas y guardando logs de todas las ejecuciones que no se hayan completado al 100%. De esta forma podéis poner a prueba vuestras soluciones con 1 solo comando y recoger en los logs la información necesaria para arreglar los casos conflictivos con los que os encontréis.

Una vez os hayáis descargado el fichero, al descomprimirlo encontraréis el script launch. Para saber como funciona sólo tenéis que ejecutarlo sin parámetros (Launch). Si no podéis ejecutarlo por falta de permisos, le podéis dar permiso de ejecución escribiendo `chmod +x launch`.

El script funciona en cualquier Linux / \*NIX con el shell bash instalado.

LAST\_UPDATED2

**Instrucciones para la instalación de SWI-Prolog en Linux o MAC**

Carlos Villegra

**Documento** con las instrucciones para la instalación de SWI-Prolog en Linux.

**Documento** con las instrucciones para la instalación de SWI-Prolog en Mac OS X.

Si tenéis cualquier problema en dicha instalación comunicadlo a vuestro profesor de prácticas.

LAST\_UPDATED2



# *Instalación de PLMan*

- 1) Descargar *pl-man.zip* (por ej. en *~/Desktop/*)
- 2) Abrir un terminal y entrar donde está *pl-man.zip*
- 3) Descomprimir *pl-man.zip*: `unzip pl-man.zip`
- 4) Entrar en la carpeta *plman/*: `cd plman`
- 5) Comprobar que está todo: `ls`
  - `docs/` - manual de uso
  - `maps/` - mapas de ejemplo
  - `pl-man-game/` - código fuente del juego PL-Man
  - `plman` - script de lanzamiento



# *Instalación de PLMan*

1) Ver ayuda del script: `./plman --help`

2) Prototipo del script:

`./plman MAPA SOLUCION [PARAMETROS]`

3) Editar fichero solución básico `solucion.pl`

```
:- use_module('pl-man-game/main') .  
do(move(none)) .
```

4) Ejemplo:

`./plman maps/ejemplos/mapaej0.pl solucion.pl`

5) Dentro de PLMan:

- **ESC** - finalizar ejecución
- **Otra tecla** - ejecutar un paso



# *Uso básico de PLMan*

**maps/ejemplos/mapaej0.pl**

```
#####  
#      E      #  
#@.....#  
#      E      #  
#####
```



# ***Prolog: Comentarios***

Los comentarios son ignorados por el intérprete de Prolog. Sirven para hacer el código más fácil de entender.

```
% Comentario  
% Otro Comentario  
... código ...
```

```
% Más Comentarios  
... código ...
```

```
/* Comentario de  
2 líneas */
```



# ***Términos: átomos y números***

## **TÉRMINO**

Cualquier número, átomo o variable.

## **ÁTOMO (constante)**

Palabra para identificar a un objeto del dominio.

- **Siempre** empieza por **letra minúscula**.
- Puede contener letras, números o subrayado.
- No pueden contener espacios.
- Si va entre comillas simples ' ... ' puede contener cualquier cosa.



# *Ejemplos de constantes y números*

% constantes

llave

'llave azul'

llave\_azul

llave512

'PLMan'

% números

123

12.5

-10



# ***Términos: variables***

Representan **un término cualquiera** del dominio.

- Empiezan por **letra mayúscula o subrayado**.
- Pueden contener letras, números o subrayado.
- No pueden contener espacios.

Las variables pueden estar:

- **Sin instanciar**: aún no tienen valor.
- **Instanciadas**: están asociadas a un término.

En PROLOG, las variables siempre están cuantificadas universalmente.

Se usan para generalizar conocimiento.





# *Ejemplos de variables*

```
% variables
```

```
X
```

```
_objeto
```

```
VALOR
```

```
Fantasma_1
```

```
ENEMIGO10
```

```
_ (ivariable anónima!)
```



# ***Predicado***

Forma de relacionar datos (términos u otros predicados)

**predicado(arg1, arg2, arg3, ..., argN)**

- Empiezan siempre por letra **minúscula**.
- No hay espacio entre el nombre y el paréntesis.
- Argumentos separados por comas.
- Pueden tener 0 argumentos (y no habría paréntesis)



# *Ejemplos de predicados*

```
% llave(X): X es una llave  
llave(a)
```

```
% ver(J, O): El jugador J ve el objeto O  
ver('PLMan', a)  
ver(plman, OBJ)
```



# Acciones de PLMan

Las acciones de PLMan se realizan con el predicado **do(ACCION)**

ACCIÓN	COMANDO	EJEMPLO	DESCRIPCIÓN
Moverse	<b>move(DIR)</b>	<b>do(move(up))</b>	Moverá a PLMan hacia arriba 1 casilla en el siguiente ciclo de ejecución
Coger objeto	<b>get(DIR)</b>	<b>do(get(down))</b>	Coge el objeto que se encuentra en la casilla inferior a la de PLMan en el siguiente ciclo de ejecución
Dejar objeto	<b>drop(DIR)</b>	<b>do(drop(left))</b>	Deja el objeto que lleva PLMan en la casilla de la izquierda a la que se encuentra en el siguiente ciclo de ejecución
Usar objeto	<b>use(DIR)</b>	<b>do(use(right))</b>	Usa el objeto que lleva PLMan en la casilla sobre lo que haya a su derecha en el siguiente ciclo de ejecución



# ***Acciones de PLMan***

**DIR** representa la dirección en la que se realizará la acción, y puede tener los valores siguientes:

<b>up-left</b>	<b>up</b>	<b>up-right</b>
<b>left</b>	<b>here</b>	<b>right</b>
<b>down-left</b>	<b>down</b>	<b>down-right</b>

(no todas las acciones son válidas en todas las direcciones)



# ***fbf: fórmulas atómicas (hechos)***

## **HECHO**

Una cláusula concreta de un predicado. Indica unos valores concretos para los que ese predicado se cumple

```
% ver(DIRECCION, OBJETO)
% Veo arriba un coco
  ver(up, '.').
% Veo abajo una llave
  ver(down, 'a').

% do(ACCION)
% Me muevo a la posición de arriba
  do(move(up)).
% Cojo el objeto que hay en la posición de abajo
  do(get(down)).
```



# Ejemplos de hechos

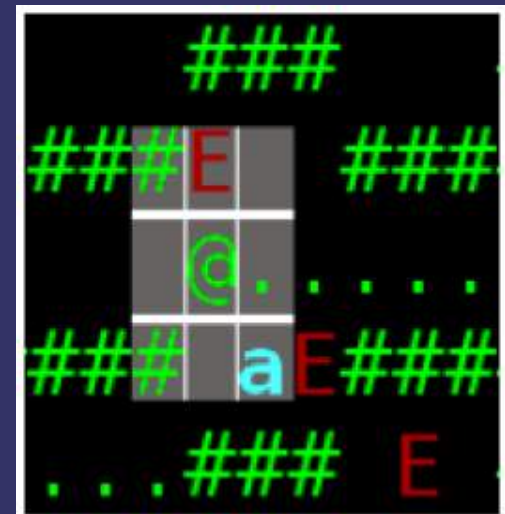
```
%%%%%%%% HECHOS
```

```
% ver(DIRECCION, OBJETO)
```

```
ver(up, 'E').  
ver(right-down, 'a').  
ver(left, ' ').  
ver(right, '.').
```

```
% do(ACCION)
```

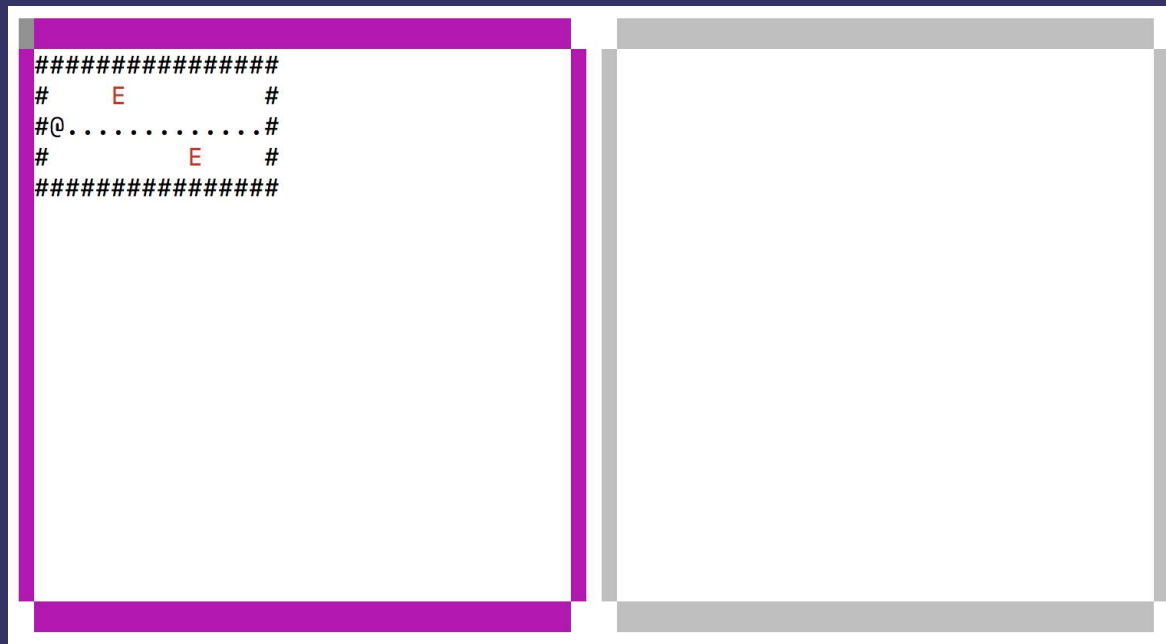
```
do(move(right)).  
do(get(down-right)). ???
```





# Uso básico de PLMan

```
./plman maps/ejemplos/mapaej0.pl sol_mapaej0.pl
```

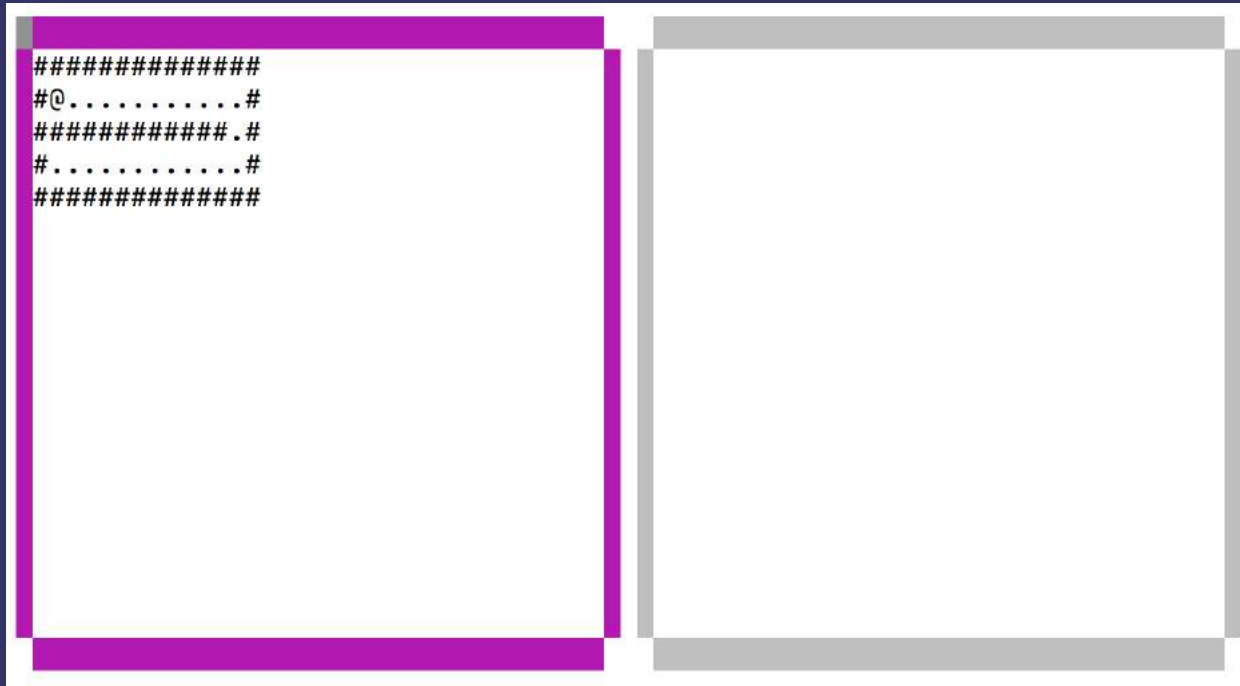


```
:- use_module('pl-man-game/main').  
  
do(move(right)).
```



# *Uso básico de PLMan*

```
./plman maps/ejemplos/mapaej1.pl sol_mapaej0.pl
```



**¡TROPIEZA  
CON LA PARED!**

```
:- use_module('pl-man-game/main').  
do(move(right)).
```



# *Sensor de visión*

La percepción del mundo de PLMan se realiza con el predicado

`see(normal, DIRECCION, OBJETO)`



# *Uso básico de PLMan*

Si ves un coco a la derecha entonces muévete a la derecha

```
ver(right, '.') --> mover(right)
```

```
do(move(right)) :- see(normal, right, '.').
```

# ***fbf: fórmulas moleculares condicionales (reglas)***

**cabeza :- cuerpo.**

**cabeza:** **objetivo** que puede ser cierto o falso, dependiendo del CUERPO. Es un dato que depende de otros.

**Cuerpo:** condiciones de las que depende que cabeza tenga éxito o fracase.

**Si cuerpo entonces cabeza**

**cuerpo --> cabeza**



# *Ejemplos de reglas*

```
%%%%%%%% REGLAS  
% do(ACCION)
```

```
do(move(right)) :- see(normal, right, '.').
```

```
do(move(DIR)) :- see(normal, DIR, '.').
```



# ***Predicados predefinidos***

## **PREDICADO PREDEFINIDO**

Un predicado que ya viene definido en SWI-PROLOG y que tiene alguna funcionalidad asociada. Sólo pueden utilizarse en preguntas al intérprete o **en el cuerpo de una regla**.

Si ponemos hechos de un predicado predefinido en un programa nuestro dará error al compilar, porque no podemos redefinirlo.

Para conocer los predicados predefinidos de swi-prolog podemos utilizar la ayuda, poniendo **?- help.** en el intérprete.



# *Ejemplos de predicados predefinidos*

## *Para escribir por pantalla*

`% write(A):` Escribe el átomo A por pantalla

`write('Aquí empieza tu aventura').`

`% writeln(A):` Añade una línea nueva al final

`writeln('Aquí empieza tu aventura').`

# *Uso básico de PLMan*

Si quieres saber lo que está pasando, escríbete mensaje

```
#####  
#      @....#  
#####.  
#.....#  
#####
```

```
Me he comido un coco  
Me he comido un coco  
Me he comido un coco  
Me he comido un coco  
Me he comido un coco  
Me he comido un coco  
Me he comido un coco
```

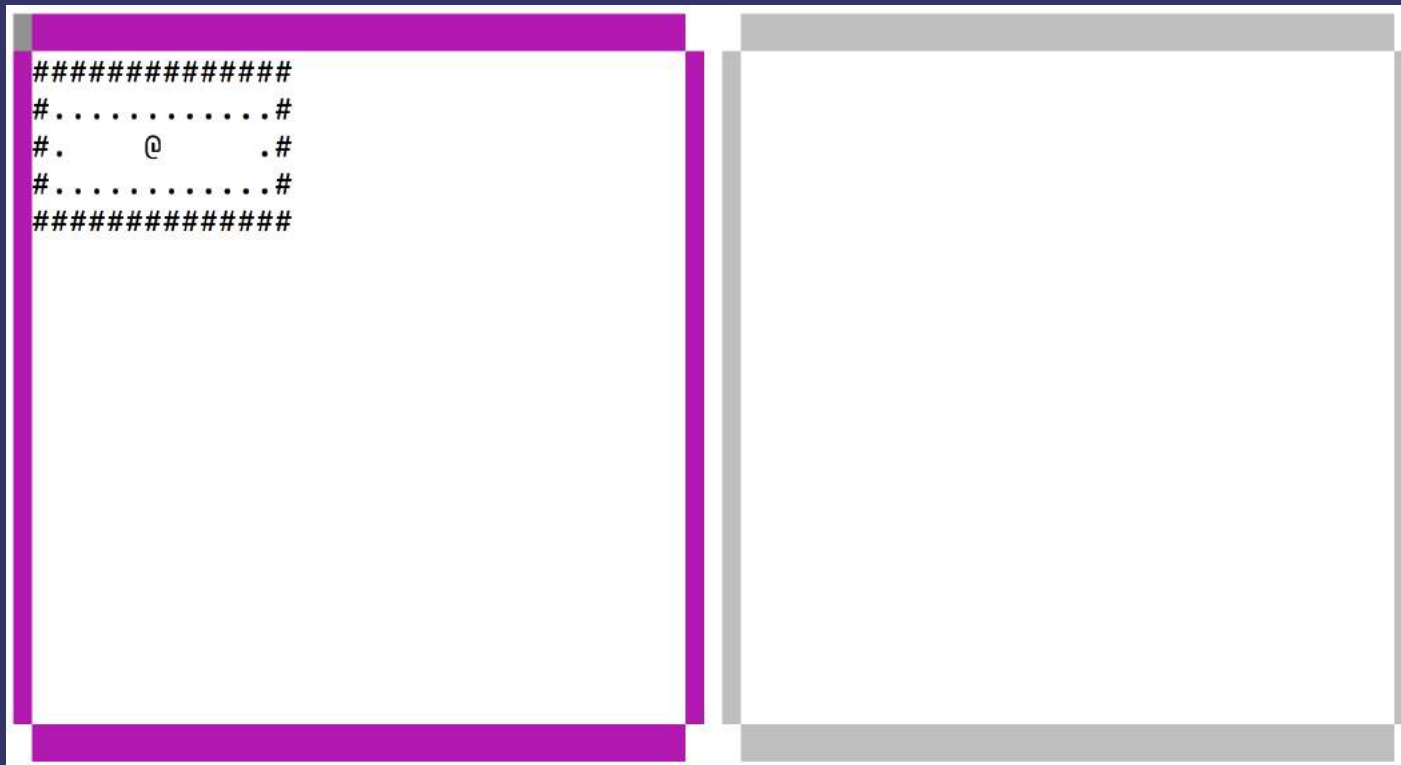
```
:- use_module('pl-man-game/main').
```

```
do(move(right)) :- see(normal, right, '.'),  
                  writeln('Me he comido un coco').
```



# *Uso básico de PLMan*

- Resolver mapaej1.pl y mapaej2.pl
- Harán falta varias reglas
- Importancia orden reglas de arriba a abajo



# *Uso básico de PLMan*

## **GENERALIZAR CONOCIMIENTO**

Cuando veas un coco entonces muévete en esa dirección

```
 $\forall x[\text{ver}(x, '.')] \rightarrow \text{mover}(x)$ 
```

```
do(move(DIR)) :- see(normal, DIR, '.').
```

# *Uso básico de PLMan*

```
./plman maps/ejemplos/mapaej1.pl sol_mapaej1.pl
```

```
#####  
#@.....#  
#####.#  
#.....#  
#####
```

**¡OJO, no todas  
las direcciones  
de movimiento  
son válidas!**

```
:- use_module('pl-man-game/main').  
  
do(move(DIR)) :- see(normal, DIR, '.').
```



# ***Resolver ejemplos***

**RESOLVER MAPAS  
carpeta EJEMPLOS**