

# Funciones

El constructor *deffunction*

# Introducción

- El constructor **deffunction** permite crear nuevas funciones dentro de CLIPS, compuesta por:
  - Un nombre
  - Un comentario que es opcional
  - Una lista de cero o más parámetros regulares
  - Un parámetro comodín opcional que gestiona un número variable de argumentos
  - Una secuencia de acciones, o expresiones, que se ejecutarán de forma secuencial cuando se llame a la función definida mediante este constructor.

# Sintaxis

```
(deffunction <nombre> [<comentario>]
  (<parametro-regular>*
   [<parametro-comodin>])
  <accion>*)
```

```
<parametro-regular> ::=
  <variable-de-campo-simple>
```

```
<parametro-comodin> ::=
  <variable-multicampo>
```

# Ejemplo

```

CLIPS 6.0
File Edit Execution Browse Window Help
CLIPS (V6.04 01/17/96)
CLIPS> (load "C:/TRANSCRIPCIONES/calvo cuenca/INFORMATICA APLICADA/
CLIPS> Defining deffunction: imprime-args
TRUE
CLIPS> (ppdeffunction imprime-args)
(deffunction MAIN::imprime-args
  (?a ?b $?c)
  (printout t ?a " " ?b " and " (length ?c) " extras: " ?c crlf))
CLIPS> (imprime-args 12 13)
12 13 and 0 extras: ()
CLIPS> (imprime-args 12 13 casa ventana)
12 13 and 2 extras: (casa ventana)
CLIPS> (imprime-args casa ventana libro "salon")
casa ventana and 2 extras: (libro "salon")
CLIPS>

```

# Conceptos

- Una función debe de tener un nombre único.
- Una función debe ser declarada antes de ser llamada.
- También pueden llamarse a sí misma y ser recursivas.

# Ejemplo

```

CLIPS 6.0
File Edit Execution Browse Window Help
TRUE
CLIPS> (factorial 4)
24
CLIPS> (ppdeffunction factorial)
(deffunction MAIN::factorial
  (?a)
  (if (or (not (integerp ?a)) (< ?a 0))
      then
        (printout t "Factorial Error!" crlf)
      else
        (if (= ?a 0)
            then
              1
            else
              (* ?a (factorial (- ?a 1))))))
CLIPS> (factorial casa)
Factorial Error!
CLIPS>

```