

Paradigmas de Programación

Práctica 2

1. Se trata de analizar la serie de expresiones Ocaml incluidas en el fichero [expresiones2.pdf](#). Para ello, abriremos el compilador interactivo de Ocaml y, con cada expresión del fichero, haremos lo siguiente:
 - La escribimos en el fichero de texto **expresiones3.ml** utilizando un editor (por ejemplo, gedit).
 - Debajo, y usando comentarios (`* . . *`), intentamos predecir el resultado que dará Ocaml sobre su compilación y ejecución, procurando usar la misma notación.
 - Copiamos la expresión en el terminal en el que tengamos abierto el compilador interactivo de Ocaml y comprobamos el resultado. Si no es el previsto, lo corregimos e intentamos razonar por qué y en qué nos hemos equivocado.
 - Para toda expresión que produzca un error:
 - i. La escribiremos en el fichero de texto entre comentarios.
 - ii. Indicaremos, también entre comentarios, el tipo de error (léxico, sintáctico, de tipo o de ejecución) y la causa del mismo.
 - iii. Trataremos de intuir la intención original de la expresión, le cambiaremos lo que sea necesario para que no produzca ningún error, y procederemos con esta nueva expresión igual que con el resto de expresiones correctas.
 - Usaremos el manual del lenguaje para averiguar el significado de los operadores y funciones que aparecen en cada expresión.
2. [Ejercicio opcional] Investigue con detalle el comportamiento de las funciones `Char.lowercase` y `Char.uppercase`, y defina con Ocaml funciones `lowercase: char -> char` y `uppercase: char -> char`, de modo que tengan exactamente el mismo valor que `Char.uppercase` y `Char.lowercase`, respectivamente. Naturalmente, no pueden utilizarse en la definición ni `Char.uppercase` ni `Char.lowercase`. Realice las implementaciones en el fichero de texto **letters.ml**.