Laboratorio

Ingeniería de Software

@SOYPABLOG – 7 DE FEBRERO DE 2023

Anteriormente...

- **Clojure**
  - **¿Qué es?**
  - **Filosofía del lenguaje**
  - **Programación funcional**
- **Herramientas necesarias**
- **Flujo de desarrollo**
- **Sintaxis**
  - **Expresiones (o formas)**
    - **Representaciones literales de datos o estructuras de datos**
    - **Operaciones**

Operaciones

- `(operador operando-1 operando-2 … operando-n)`
- Ejemplo:
  - `(+ 1 2 3) ; => 6`
  - `(str “Hello” “ World!”) ; => “Hello World!”`

Flujo de control

- if
  - (if expresión-booleana
    expresión-then
    expresión-else)
- do
- when
- if-let
  - (if-let [variable expresión-bool]
    expresión-then
    expresión-else)
- case
  - (case literal
    literal expresión-resultado
    …
    expresión-default)
- cond
  - (cond
    expresión-booleana resultado
    …)

Valores lógicos

- Falsedad
  - `false`
  - `nil` (*no value*)
- Verdad
  - Cualquier valor que no sea de falsedad.

Operadores booleanos e igualdad

- `(and operando-1 operando-2 … operando-n)`
  Devuelve el primer valor de falsedad encontrado o, en caso de no existir, el último valor de verdad.
- `(or operando-1 operando-2 … operando-n)`
  Devuelve el primer valor de verdad encontrado o el último valor.
- `(= operando-1 operando-2 … operando-n)`

Números

- Enteros
- Punto flotante
- Racionales

Cadenas de texto

Mapas

- `(def teacher {:first-name “Pablo” :last-name “González”})`
- `(get teacher :name) ; => “Pablo”`
- `(:name teacher) ; => “Pablo”`
- `(teacher :name) ; => “Pablo”`
- `(assoc teacher :middle-name “Gerardo”)`
  `; => {:first-name “Pablo” :last-name “González” :middle-name “Gerardo”}`

Keywords

Vectores

Listas

Conjuntos

`def`

- A diferencia de otros lenguajes de programación, `Clojure` **no** cuenta con un mecanismo de asignación de valores a variables. En su lugar, provee mecanismos para **ligar** (en inglés *bind*) valores a *vars*.
- *var* ≠ variable.
- `(def x 42)`
  Asocia el valor `42` al símbolo `x`.

Funciones

- Invocar funciones (en inglés *function call*).
  - `(first [1 2 3 4])`
- Definir funciones:
  - ```
    (defn nombre
      “El ‘docstring’ de la función”
      [parámetro-1 … parámetro-n]
      (cuerpo-de-la-función))
    ```
  - `(fn [parámetro-1 … parámetro-n] (cuerpo-de-la-función))`
  - `#(cuerpo-de-la-función)`
- Devolver funciones.
    ```

Destructing

- Esta técnica consiste en ligar nombres de *vars* a valores que estén dentro de una colección.

Cheatsheet

- https://clojure.org/api/cheatsheet

🏋️ Ejercicio

- Escribir una función que regrese la cantidad de elementos en una secuencia.
  - `(= (__ '(1 2 3 3 1)) 5)`
  - `(= (__ "Hello World") 11)`
  - `(= (__ [[1 2] [3 4] [5 6]]) 3)`
  - `(= (__ '(13)) 1)`
  - `(= (__ '(:a :b :c)) 3)`

En resumen...

`Clojure` cuenta con distintas estructuras de flujo control que interpretan los datos como valores *booleanos* (solo `false` y `nil` son valores de falsedad).

Contamos con los tipos de datos básicos (*booleanos*, números, cadenas de texto, *keywords*).

También se cuenta con estructuras de datos como vectores, listas, mapas y conjuntos.

No contamos con “estado”, por lo que las variables no almacenan valores, sino que simplemente quedan ligadas a estos.

 Homework

- Realizar la *kata* `alphabet-cipher` de `wonderland-clojure-katas` (https://github.com/gigasquid/wonderland-clojure-katas)