02NPYPD - LINGUAGGI E AMBIENTI MULTIMEDIALI A

PRIMI PASSI CON RUBY

Questa esercitazione ha lo scopo di farvi esplorare e provare alcuni concetti e metodi alla base di Ruby. I singoli esercizi hanno una parte di spiegazione e una di esecuzione vera e propria.

COSA SERVE?

Per svolgere questa esercitazione, potete usare RubyMine:

- Create un nuovo progetto vuoto
- Inserite un nuovo file con estensione .rb

ESERCIZIO 1 - STRINGHE

In Ruby, le stringhe si possono unire in (almeno) due modi diversi: *concatenandole* (con `+` o `<<`) oppure *interpolandole* (con `#{}`). Provate a definire due variabili `nome` e `cognome`, assegnandogli rispettivamente il vostro nome e il vostro cognome. Provate a stampare le stringhe *nome* o *cognome* con `puts` e `print`: differenze?

Stampate la stringa "Io mi chiamo *nome cognome*" usando i diversi metodi (con `+`, `<<` e `#{}`).

Ruby riconosce come stringhe sia del testo racchiuso tra virgolette singole (') sia racchiuso tra virgolette doppie ("). Provate a scrivere `ciao #{nome}` prima tra virgolette singole e poi tra virgolette doppie. Cosa succede? Quale delle due virgolette permette di esprimere letterali puri (cioè rappresenta esattamente quello che viene digitato)?

Provate a scrivere l'espressione corrispondente a "*stampa il contenuto della variabile nome se la variabile nome non è vuota*" sia nella forma 'tradizionale' che nella 'modifier form' .

Date le stringhe "ciao mondo" e "ciao_mondo", provate ad applicare il metodo `split` a entrambe in modo che il risultato visualizzato a video sia simile a `["ciao", "mondo"]`.

ESERCIZIO 2 - ARRAY E RANGE

Un array è una lista di elementi ordinata. Definite l'array `vett=[15, 30, 90, 22, 70]` e provate a stamparlo: `puts` o `print`?

A un array è possibile applicare vari metodi, ad esempio: `first`, `last`, `length`, `sort`, `reverse`, `shuffle`. Dopo averli applicati, visualizzate il contenuto di `vett`. Il contenuto dell'array è cambiato? Se, invece, digitate `vett.sort!` e poi visualizzate il suo contenuto, cambia qualcosa rispetto a prima?

Inserite altri valori in vett utilizzando prima il metodo `push()` e poi l'operatore `<<`. Il risultato cambia? È possibile inserire delle stringhe in `vett`? Perché?

Date ora il comando `vett[0..2]`. Cosa succede?

`0..2` è un range. I range permettono anche di creare vettori di vario tipo: provate ad eseguire `(0..9).to_a` e poi `('a'..'z').to_a`. Cosa fanno queste istruzioni?

ESERCIZIO 3 - BLOCCHI

I blocchi sono una delle più potenti (e confusionarie) funzionalità di Ruby. Un blocco può essere dichiarato su una linea sola (e allora inizierà e terminerà con una parentesi graffa) oppure su più linee (sarà compreso tra un `do` e un `end`). Consideriamo, come esempio, l'istruzione: `3.times { puts "Ciao!" }`. Questa istruzione ripeterà tre volte il blocco `puts "Ciao!"` e quindi stamperà sullo schermo:

Ciao!
Ciao!
Ciao!

Con i blocchi si possono eseguire operazioni complicate a piacere. Provate a immaginare cosa fanno queste istruzioni (prima di farle eseguire):

1. ```
   animals = %w{cat dog horse rabbit owl}
   animals.each {|anim| puts anim}
   ```

2. ```
   (1..5).each { |i| puts 2 * i }
   ```

3. ```
   ('a'..'z').to_a.shuffle[0..7].join
   ```

4. ```
   my_str = "blah, blah"
   puts my_str.split(",")[0].split("")[2] * 3
   ```

Solo dopo controllate di avere ragione…

ESERCIZIO 4 - SIMBOLI E HASH

Creare una hash table adatta a rappresentare la tavola degli elementi chimici, associando a ciascun simbolo il suo nome (es. H -> idrogeno). Utilizzare i simboli Ruby per rappresentare i simboli chimici.

Provare poi, ad esempio, a:

1) Visualizzare tutto il contenuto della tabella secondo un formato a vostra scelta (ad esempio: "L'elemento idrogeno ha come simbolo H.")
2) Sperimentare i metodi `include?`, `has_key?`, `has_value?`, controllando se la tabella contiene un determinato simbolo o elemento.
3) Visualizzare il numero di elementi presenti nella tabella.

4) Data una formula chimica (ad esempio H2O) verificare se contiene un determinato elemento (es: H) utilizzando le espressioni regolari.

ESERCIZIO 5 – HASH e BLOCCHI

Creare un programma per cifrare un messaggio segreto: il programma deve modificare una stringa sostituendo ogni carattere con un altro, secondo un codice memorizzato in una hash table (e.g. “A” diventa “K”, ecc. a vostro piacere).

Esempio di output:

La mia stringa e':
Ciao sono Laura
Il messaggio in codice e':
ANUY EYVY MUXTU