



Discovery Piscine

Rush - Python

Summary: Fin dove siamo arrivati.

Version:

Contents

| | | |
|------------|-----------------------------------|----------|
| I | Cosa ti aspetta? | 2 |
| II | Introduzione | 3 |
| III | Istruzioni generali | 4 |
| IV | Exercise 00: Phonebook | 5 |
| V | Bonus | 7 |
| VI | Consegna e peer-evaluation | 8 |

Chapter I

Cosa ti aspetta?

Ciao!

Stai per iniziare la prima fase di questa discovery piscine di programmazione informatica.

L'obiettivo è mostrarti il codice che sta dietro al software che usi quotidianamente e, allo stesso tempo, farti vivere l'esperienza dell'apprendimento tra pari, il modello educativo della 42.

La programmazione riguarda la logica (non la matematica). Ti mette a disposizione dei mattoni elementari, che puoi assemblare come preferisci. Non esiste mai LA soluzione a un problema. Ci sarà la tua soluzione e quelle dei tuoi compagni. Lenta o veloce, brutta o elegante, se funziona, è tutto ciò che conta! Questo assemblaggio di mattoni costituirà una serie di comandi (calcolo, visualizzazione, ecc.) che il computer eseguirà nell'ordine che hai scelto.

Invece di fornirti un corso con un'unica soluzione per ogni problema, che probabilmente diventerebbe obsoleta nel giro di qualche anno, abbiamo scelto di metterti in una situazione di apprendimento tra pari. Sarai tu a cercare gli elementi utili per affrontare le tue sfide, a selezionare quelli più interessanti sperimentandoli e manipolandoli, e a creare il tuo programma. Per farlo, discuterai con gli altri, condividerai i tuoi punti di vista, elaborerai nuove idee insieme e, infine, testerai le soluzioni da solo per convincerti che funzionano.

La valutazione tra pari è un momento fondamentale per scoprire altri approcci, oltre che per individuare casi particolari a cui potresti non aver pensato e che potrebbero compromettere il tuo script (come la frustrazione che provi quando un software si blocca). Come clienti diversi che si concentrano su aspetti differenti, ogni revisore avrà una prospettiva unica. E chissà, potresti fare nuove conoscenze per future collaborazioni.

Alla fine di questa *piscine*, non avrai seguito lo stesso percorso degli altri partecipanti, non avrai completato gli stessi progetti e avrai scelto di affrontare alcune sfide piuttosto che altre... ed è normale! È un'esperienza sia collettiva che individuale. Ognuno trarrà beneficio da ciò che vivrà durante questo periodo.

Buona fortuna a tutti, speriamo che questa discovery vi entusiasmi!

Chapter II

Introduzione

Cosa affronterai in questa cellula:

- Creare una classe complessa con metodi e variabili.

Chapter III

Istruzioni generali

Escludendo eccezioni, le seguenti regole saranno in rigore per tutta la durata della piscine.

- Questo subject è l'unica fonte affidabile. Non credere alle dicerie.
- Presta attenzione ai permessi di accesso dei tuoi file e cartelle.
- I tuoi progetti verranno valutati dai tuoi compagni di Piscine.
- Tu non devi lasciare nella consegna nessun file o cartella che non sia esplicitamente richiesto dal subject.
- Hai delle domande? Chiedi alla persona alla tua sinistra. Altrimenti, chiedi alla persona alla tua destra.
- Ogni domanda tecnica può essere risolta tramite il **man** o tramite Internet.
- Leggi attentamente gli esempi. Possono rivelare dettagli dell'esercizio che non sono espliciti nel subject
- Negli esempi troverai spesso "?>", questo rappresenta il prompt del terminale. Nel tuo caso il prompt apparirà con il numero della tua postazione "c1r1p1"
- Per Thor, per Odino! Usa il tuo cervello!!!

Chapter IV

Exercise 00: Phonebook

- Crea uno script chiamato `Phonebook.py`.
- Questo script conterrà la dichiarazione della classe `Phonebook`
- Questa classe conterrà un attributo di tipo `dict` che memorizzerà i numeri di telefono dei contatti come chiavi e i nomi come valori
- Dovrai anche implementare alcuni metodi per permetterti di interagire con i contatti:
- I metodi si chiameranno: `add_contact`, `remove_contact` e `contact_list`
- Quando eseguito, lo script chiederà all'utente la prossima azione da fare: `ADD`, `REMOVE`, `LIST` o `CLOSE`. Ognuna di queste azioni chiamerà il rispettivo metodo o chiuderà il programma.
- Il metodo `add_contact` chiederà all'utente di inserire un nuovo contatto, prima chiedendo il numero di telefono, poi il nome, e aggiungendoli al `dict` della rubrica.
- Il metodo `remove_contact` chiederà un numero di telefono e, se esiste, lo rimuoverà dal `dict`
- Il metodo `contact_list` stamperà i contatti memorizzati nel `dict`

```
>>> python3 whatsyourname.py
cosa vuoi fare dopo (ADD, REMOVE, LIST, CLOSE): ADD
numero di telefono: 4242424242
nome: Pasquale
cosa vuoi fare dopo (ADD, REMOVE, LIST, CLOSE): LIST
{'4242424242': 'Pasquale'}
cosa vuoi fare dopo (ADD, REMOVE, LIST, CLOSE): REMOVE
numero di telefono: 4242424242
cosa vuoi fare dopo (ADD, REMOVE, LIST, CLOSE): LIST

cosa vuoi fare dopo (ADD, REMOVE, LIST, CLOSE): CLOSE
?>
```



Cerca su internet come creare e utilizzare classi in python.



Nota come lo script non si chiuda mai da solo; l'unico modo per chiuderlo sarà il comando CLOSE. Cerca 'while true'

Chapter V

Bonus

Se sei all'altezza del compito, ti chiediamo di implementare alcune funzionalità extra nella tua classe

- Un nuovo metodo che ti consenta di interagire con i contatti in un modo diverso, per modificarli o per altre funzionalità
- Parsing per i metodi esistenti, verificando se l'input fornito è effettivamente un numero, i nomi non possono contenere numeri, ecc.
- Migliorare la memorizzazione dei dati per la classe in modo che accetti anche l'email e il cognome della persona; come gestirlo spetta a te

Chapter VI

Consegna e peer-evaluation

- Crea una Cartella nella root della tua home chiamata `discovery_piscine`, Utilizzerai questa per tutti gli esercizi.
- Crea una nuova cartella chiamata `Rush` e aprila nel terminale.



Durante la correzione, qualsiasi cosa non sia presente nella cartella non sarà valutato.