



27/07/23

**master.D**

# INTERFACCIA FUNZIONALE IN PYTHON

Campari Mirko

## **COS'È UN'INTERFACCIA FUNZIONALE?**

- **IN INFORMATICA, UN'INTERFACCIA FUNZIONALE È UN CONCETTO CHE SI BASA SULLA PROGRAMMAZIONE FUNZIONALE.**
- **LA PROGRAMMAZIONE FUNZIONALE È UNO STILE DI PROGRAMMAZIONE IN CUI LE OPERAZIONI VENGONO ESEGUITE ATTRAVERSO FUNZIONI E L'USO DI DATI IMMUTABILI.**
- **LE INTERFACCE FUNZIONALI CONSENTONO DI UTILIZZARE FUNZIONI COME PARAMETRI E RESTITUIRE FUNZIONI COME RISULTATI.**

## **ORIGINE DELL'INTERFACCIA FUNZIONALE**

- **L'INTERFACCIA FUNZIONALE HA ORIGINE NEL PARADIGMA DELLA PROGRAMMAZIONE FUNZIONALE.**
- **QUESTO PARADIGMA È BASATO SULLA MATEMATICA E HA RADICI NEL CALCOLO LAMBDA DI ALONZO CHURCH NEGLI ANNI '30.**
- **CON L'AVVENTO DEI LINGUAGGI DI PROGRAMMAZIONE FUNZIONALI NEGLI ANNI '50 E '60, IL CONCETTO DI INTERFACCIA FUNZIONALE HA GUADAGNATO POPOLARITÀ.**

# **CARATTERISTICHE DELL'INTERFACCIA FUNZIONALE**

## **FUNZIONI DI PRIMA CLASSE:**

- **LE FUNZIONI SONO TRATTATE COME CITTADINI DI PRIMA CLASSE.**
- **POSSONO ESSERE ASSEGNATE A VARIABILI, PASSATE COME ARGOMENTI E RESTITuite COME RISULTATI DA ALTRE FUNZIONI.**

# **CARATTERISTICHE DELL'INTERFACCIA FUNZIONALE**

## **FUNZIONI PURE:**

- **LE FUNZIONI SONO PURE SE RESTITUISCONO UN VALORE BASATO SOLO SUI LORO ARGOMENTI, SENZA EFFETTI COLLATERALI.**
- **LE FUNZIONI PURE FAVORISCONO IL PARALLELISMO E LA FACILITÀ DI TESTING.**

# **CARATTERISTICHE DELL'INTERFACCIA FUNZIONALE**

## **CLOSURES:**

- **LE FUNZIONI POSSONO ACCEDERE ALLE VARIABILI DEFINITE NEL LORO AMBIENTE DI DEFINIZIONE.**
- **QUESTE VARIABILI RIMANGONO "LEGATE" ALLA FUNZIONE ANCHE QUANDO VIENE CHIAMATA ALTROVE.**

# **CARATTERISTICHE DELL'INTERFACCIA FUNZIONALE**

## **CLOSURES:**

- **LE FUNZIONI POSSONO ACCEDERE ALLE VARIABILI DEFINITE NEL LORO AMBIENTE DI DEFINIZIONE.**
- **QUESTE VARIABILI RIMANGONO "LEGATE" ALLA FUNZIONE ANCHE QUANDO VIENE CHIAMATA ALTROVE.**

# **CARATTERISTICHE DELL'INTERFACCIA FUNZIONALE**

## **HIGH-ORDER FUNCTIONS:**

- **LE HIGH-ORDER FUNCTIONS SONO FUNZIONI CHE ACCETTANO ALTRE FUNZIONI COME ARGOMENTI O LE RESTITUISCONO COME RISULTATO.**
- **QUESTO PERMETTE DI SCRIVERE CODICE PIÙ FLESSIBILE E RIUTILIZZABILE.**



# CARATTERISTICHE DELL'INTERFACCIA FUNZIONALE

**# Definiamo una funzione di ordine superiore che accetta una funzione come argomento**

```
1. def applica_funzione(funzione, lista):  
2.     risultati = []  
3.     for elemento in lista:  
4.         risultati.append(funzione(elemento))  
5.     return risultati  
6.  
7. # Definiamo una funzione da utilizzare  
8. def quadrato(x):  
9.     return x * x  
10.  
11. # Utilizziamo la funzione di ordine superiore con la nostra funzione  
12. numeri = [1, 2, 3, 4, 5]  
13. risultati_quadrati = applica_funzione(quadrato, numeri)  
14.  
15. print(risultati_quadrati) # Output: [1, 4, 9, 16, 25]
```

**IN QUESTO ESEMPIO, ABBIAMO CREATO UN'APPLICAZIONE DI FUNZIONE DI ORDINE SUPERIORE APPLICA\_FUNZIONE CHE PRENDE UNA FUNZIONE FUNZIONE E UNA LISTA DI NUMERI LISTA.**

**LA FUNZIONE APPLICA\_FUNZIONE APPLICA LA FUNZIONE FUNZIONE A CIASCUN ELEMENTO DELLA LISTA E RESTITUISCE I RISULTATI IN UNA NUOVA LISTA.**

**ABBIAMO QUINDI DEFINITO UNA FUNZIONE QUADRATO E L'ABBIAMO UTILIZZATA CON APPLICA\_FUNZIONE PER CALCOLARE I QUADRATI DEI NUMERI DATI.**

## **CONCLUSIONI**

- **L'INTERFACCIA FUNZIONALE IN PYTHON È UN POTENTE STRUMENTO CHE CI PERMETTE DI SCRIVERE CODICE PIÙ PULITO, MODULARE E FLESSIBILE.**
- **UTILIZZARE LE FUNZIONI COME OGGETTI DI PRIMA CLASSE CI AIUTA A CREARE CODICE PIÙ ELEGANTE E RIUTILIZZABILE.**
- **LA PROGRAMMAZIONE FUNZIONALE È SEMPRE PIÙ POPOLARE IN PYTHON GRAZIE ALLA SUA FACILITÀ DI LETTURA E CAPACITÀ DI RISOLVERE COMPLESSI PROBLEMI IN MODO CONCISO.**