*Studente: Isabel Munaylla*

**Sviluppo degli esercizi**

1. Scrivi una funzione che restituisca il massimo di tre numeri.

def massimo\_tra\_tre\_numeri():

try:

num1 = float(input("Inserisci il primo numero: "))

num2 = float(input("Inserisci il secondo numero: "))

num3 = float(input("Inserisci il terzo numero: "))

massimo = num1

if num2 > massimo:

massimo = num2

if num3 > massimo:

massimo = num3

return massimo

except ValueError:

return "Inserisci tre numeri validi."

# Esempio di utilizzo:

risultato = massimo\_tra\_tre\_numeri()

print(risultato)

2. Scrivi una funzione che converta gradi Celsius in Fahrenheit.

def celsius\_to\_fahrenheit(gradi\_celsius):

gradi\_fahrenheit = (gradi\_celsius \* 9/5) + 32

return gradi\_fahrenheit

# Esempio di utilizzo:

celsius = float(input("Inserisci la temperatura in gradi Celsius: "))

fahrenheit = celsius\_to\_fahrenheit(celsius)

print(f"{celsius} gradi Celsius corrispondono a {fahrenheit} gradi Fahrenheit.")

3. Scrivi un programma che rimuova i duplicati da una lista.

def rimuovi\_duplicati(lista):

lista\_senza\_duplicati = list(set(lista))

return lista\_senza\_duplicati

# Esempio di utilizzo

lista\_originale = [1, 2, 2, 3, 4, 4, 5]

lista\_pulita = rimuovi\_duplicati(lista\_originale)

print(lista\_pulita)

4. Crea una funzione che verifichi se una parola è un palindromo.

def is\_palindromo():

parola = input("Inserisci una parola o frase per verificare se è un palindromo: ")

# Rimuovere gli spazi e convertire in minuscolo per la verifica

parola\_pulita = parola.replace(" ", "").lower()

# Confrontare la parola con la sua versione invertita

if parola\_pulita == parola\_pulita[::-1]:

print(f"'{parola}' è un palindromo.")

else:

print(f"'{parola}' non è un palindromo.")

is\_palindromo()

5. Scrivi una funzione che calcoli il fattoriale di un numero.

def fattoriale(n):

if n < 0:

return "Il fattoriale non è definito per numeri negativi"

risultato = 1

for i in range(2, n + 1):

risultato \*= i

return risultato

# Esempio di utilizzo

print(fattoriale(5)) # Dovrebbe restituire 120

print(fattoriale(0)) # Dovrebbe restituire 1

print(fattoriale(-1)) # Dovrebbe restituire un messaggio di errore

6. Crea un dizionario da due liste, una che contiene le chiavi e l’altra i valori.

def crea\_dizionario(chiavi, valori):

# Assicurarsi che le liste abbiano la stessa lunghezza

if len(chiavi) != len(valori):

return "Le liste devono avere la stessa lunghezza"

# Creare il dizionario usando zip e dict

dizionario = dict(zip(chiavi, valori))

return dizionario

# Esempio di utilizzo

chiavi = ['a', 'b', 'c']

valori = [1, 2, 3]

dizionario\_creato = crea\_dizionario(chiavi, valori)

print(dizionario\_creato)

7. Scrivi una funzione che unisca due liste in una lista di tuple.

def unisci\_in\_tuple(lista1, lista2):

return list(zip(lista1, lista2))

# Esempio di utilizzo

lista1 = [1, 2, 3]

lista2 = ['a', 'b', 'c']

lista\_combinata = unisci\_in\_tuple(lista1, lista2)

print(lista\_combinata)

8. Scrivi un programma che ordini una lista di tuple per il secondo elemento.

def ordina\_per\_secondo\_elemento(lista\_di\_tuple):

return sorted(lista\_di\_tuple, key=lambda x: x[1])

# Esempio di utilizzo

lista\_di\_tuple = [(1, 'b'), (2, 'a'), (4, 'd'), (3, 'c')]

lista\_ordinata = ordina\_per\_secondo\_elemento(lista\_di\_tuple)

print(lista\_ordinata)

9. Utilizza una list comprehension per creare una lista dei quadrati dei primi 10 numeri interi.

quadrati = [x\*\*2 for x in range(1, 11)]

print(quadrati)

10. Utilizza una list comprehension per estrarre numeri pari da una lista data.

# Lista di esempio

lista\_numeri = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# List comprehension per filtrare i numeri pari

numeri\_pari = [num for num in lista\_numeri if num % 2 == 0]

print(numeri\_pari)

11. Crea una list comprehension che converta una lista di stringhe in maiuscolo.

# Lista di esempio con alcune stringhe

lista\_stringhe = ["ciao", "mondo", "python", "programmazione"]

# List comprehension per convertire ogni stringa in maiuscolo

lista\_maiuscole = [stringa.upper() for stringa in lista\_stringhe]

print(lista\_maiuscole)

12. Utilizza una list comprehension per creare una lista di tuple (numero, quadrato del numero).

# List comprehension per creare una lista di tuple (numero, quadrato del numero)

lista\_tuple = [(x, x\*\*2) for x in range(1, 11)]

print(lista\_tuple)

13. Scrivi una list comprehension che filtri le parole di una lista che iniziano con una vocale.# Lista di esempio con alcune parole

lista\_parole = ["albero", "casa", "elefante", "isola", "ombrello", "scala"]

# Vocali

vocali = 'aeiou'

# List comprehension per filtrare le parole che iniziano con una vocale

parole\_con\_vocale = [parola for parola in lista\_parole if parola[0].lower() in vocali]

print(parole\_con\_vocale)

14. Utilizza una list comprehension per eliminare i valori negativi da una lista.

# Lista di esempio con alcuni valori, inclusi valori negativi

lista\_valori = [10, -5, 20, -3, 30, -1, 40]

# List comprehension per eliminare i valori negativi

valori\_non\_negativi = [num for num in lista\_valori if num >= 0]

print(valori\_non\_negativi)

15. Crea una lista di tutte le lettere dell’alfabeto con una list comprehension.

import string

# List comprehension per creare una lista di tutte le lettere dell'alfabeto

alfabeto = [lettera for lettera in string.ascii\_lowercase]

print(alfabeto)

**UTILIZZO NUMPY**

16. Crea un array numpy di dimensione 3x3 con valori casuali.

import numpy as np

# Creare un array 3x3 con valori casuali

array\_casuale = np.random.rand(3, 3)

print(array\_casuale)

17. Scrivi un programma che calcoli la somma di tutti gli elementi in un array numpy.

import numpy as np

# Creazione di un array NumPy di esempio

array\_esempio = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

# Calcolo della somma di tutti gli elementi dell'array

somma\_totale = np.sum(array\_esempio)

print("Somma totale degli elementi dell'array:", somma\_totale)

18. Crea un array numpy e calcolane la media, mediana e varianza.

import numpy as np

# Creazione di un array NumPy di esempio

array\_esempio = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])

# Calcolo della media

media = np.mean(array\_esempio)

# Calcolo della mediana

mediana = np.median(array\_esempio)

# Calcolo della varianza

varianza = np.var(array\_esempio)

print("Media:", media)

print("Mediana:", mediana)

print("Varianza:", varianza)

19. Scrivi un programma che moltiplichi due matrici numpy.

import numpy as np

# Creazione di due matrici NumPy di esempio

matrice\_A = np.array([[1, 2], [3, 4]])

matrice\_B = np.array([[5, 6], [7, 8]])

# Moltiplicazione delle matrici

prodotto\_matriciale = np.dot(matrice\_A, matrice\_B)

# Alternativamente, puoi usare: prodotto\_matriciale = matrice\_A @ matrice\_B

print("Prodotto matriciale delle due matrici:\n", prodotto\_matriciale)

20. Crea un array numpy e ordinalo in modo ascendente.

import numpy as np

# Creazione di un array NumPy di esempio

array\_esempio = np.array([3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5])

# Ordinamento dell'array in modo ascendente

array\_ordinato = np.sort(array\_esempio)

print("Array originale:", array\_esempio)

print("Array ordinato:", array\_ordinato)

PANDAS

21. Crea un DataFrame pandas da un dizionario

import pandas as pd

# Dizionario con i dati

dati = {

'Nome': ['Alice', 'Bob', 'Charlie', 'David'],

'Età': [25, 30, 35, 40],

'Sesso': ['F', 'M', 'M', 'M']

}

# Creazione del DataFrame

df = pd.DataFrame(dati)

# Stampare il DataFrame

print(df)

22. Leggi un file CSV in un DataFrame pandas.

import pandas as pd

# Leggi il file CSV in un DataFrame

df = pd.read\_csv('.\datos-google-ads.csv')

# Stampare il DataFrame

print(df)

23. Seleziona una colonna specifica da un DataFrame.

import pandas as pd

# Seleziona la colonna 'Nome'

colonna\_keyword = df['Keyword']

# Stampare la colonna selezionata

print(colonna\_keyword)

24. Filtra un DataFrame pandas in base a una condizione.

import pandas as pd

# Filtra il DataFrame in base alla condizione

df\_filtrato = df[df['Avg. monthly searches'] > 1000]

# Stampare il DataFrame filtrato

print(df\_filtrato)

25. Calcola la media dei valori in ogni colonna di un DataFrame.

import pandas as pd

# Leggi il file CSV in un DataFrame

df = pd.read\_csv('.\datos-google-ads.csv')

# Calcola la media solo per le colonne numeriche

media\_per\_colonna\_numerica = df.mean(numeric\_only=True)

# Stampare la media per ogni colonna numerica

print(media\_per\_colonna\_numerica)

**Creazione di PLOT**

26. Crea un grafico a linee di una serie temporale con Matplotlib.

**import pandas as pd**

**import matplotlib.pyplot as plt**

**df = pd.read\_csv('.\\Amazon\_Customer\_Behavior\_Survey.csv', parse\_dates=['Timestamp'], index\_col='Timestamp')**

**print(df.head()) # Visualizza le prime righe del DataFrame per controllare la struttura dei dati**

**print(df.info()) # Visualizza informazioni sul DataFrame, come tipi di dati e valori non nulli**

**# Creare il grafico a linee**

**plt.figure(figsize=(10, 6))**

**plt.plot(df.index, df['Gender'], color='blue', marker='o', linestyle='-')**

**# Impostare le etichette degli assi x (in mesi)**

**plt.xticks(df.index[::30], [date.strftime('%b') for date in df.index[::30]])**

**# Aggiungere titolo ed etichette degli assi**

**plt.title('Grafico a Linee di Serie Temporale')**

**plt.xlabel('Mese')**

**plt.ylabel('Age')**

**# Aggiungere la griglia**

**plt.grid(True)**

**# Mostrare il grafico**

**plt.show()**

27. Crea un istogramma di una distribuzione di valori con Matplotlib.

import matplotlib.pyplot as plt

# Dati di esempio

valori = [1, 2, 3, 4, 4, 4, 5, 5, 6, 6, 6, 6, 7, 7, 7, 8, 8, 9, 9, 9]

# Crea l'istogramma

plt.hist(valori, bins=10, color='skyblue', edgecolor='black')

# Aggiungi titolo ed etichette degli assi

plt.title('Istogramma di una Distribuzione di Valori')

plt.xlabel('Valore')

plt.ylabel('Frequenza')

# Mostra l'istogramma

plt.show()

28. Crea un grafico a barre per rappresentare i dati di un DataFrame.

**import pandas as pd**

**import matplotlib.pyplot as plt**

**# DataFrame di esempio**

**data = {'Categoria': ['A', 'B', 'C', 'D'],**

**'Valore': [10, 20, 15, 25]}**

**df = pd.DataFrame(data)**

**# Crea il grafico a barre**

**df.plot(kind='bar', x='Categoria', y='Valore', color='skyblue')**

**# Aggiungi titolo ed etichette degli assi**

**plt.title('Grafico a Barre dei Dati del DataFrame')**

**plt.xlabel('Categoria')**

**plt.ylabel('Valore')**

**# Mostra il grafico**

**plt.show()**

29. Crea un diagramma a dispersione (scatter plot) con due variabili di un DataFrame.

**import pandas as pd**

**import matplotlib.pyplot as plt**

**# DataFrame di esempio**

**data = {'X': [1, 2, 3, 4, 5],**

**'Y': [10, 20, 15, 25, 30]}**

**df = pd.DataFrame(data)**

**# Crea il diagramma a dispersione**

**plt.scatter(df['X'], df['Y'], color='red', marker='o')**

**# Aggiungi titolo ed etichette degli assi**

**plt.title('Diagramma a Dispersione di Due Variabili')**

**plt.xlabel('X')**

**plt.ylabel('Y')**

**# Mostra il diagramma a dispersione**

**plt.show()**

30. Crea un box plot per analizzare la distribuzione in un DataFrame.

**import pandas as pd**

**import matplotlib.pyplot as plt**

**# DataFrame di esempio**

**data = {'A': [1, 2, 3, 4, 5, 6, 7, 8, 9],**

**'B': [2, 4, 6, 8, 10, 12, 14, 16, 18],**

**'C': [3, 6, 9, 12, 15, 18, 21, 24, 27]}**

**df = pd.DataFrame(data)**

**# Crea il box plot**

**plt.figure(figsize=(8, 6)) # Imposta le dimensioni della figura**

**df.boxplot()**

**# Aggiungi titolo ed etichette degli assi**

**plt.title('Box Plot per la Distribuzione dei Dati')**

**plt.xlabel('Variabili')**

**plt.ylabel('Valori')**

**# Mostra il box plot**

**plt.show()**

**UTILIZZO SCRAPY**

31. Scrivi uno spider Scrapy che estragga i titoli da una pagina web.

**import scrapy**

**class MyspiderSpider(scrapy.Spider):**

**name = "myspider"**

**allowed\_domains = ["ricette.giallozafferano.it"]**

**start\_urls = ["https://ricette.giallozafferano.it/Finocchi-in-padella.html"]**

**def parse(self, response):**

**# Estrai i titoli utilizzando XPath**

**titles = response.css('h1::text').getall()**

**# Stampa i titoli estratti**

**for title in titles:**

**yield {'title': title}**

**print(title)**

32. Estrai tutti i link da una pagina web utilizzando Scrapy.

import scrapy

class LinkSpider(scrapy.Spider):

name = "link\_spider"

start\_urls = ["https://ricette.giallozafferano.it/Finocchi-in-padella.html"]

def parse(self, response):

links = response.css('a::attr(href)').getall()

for link in links:

yield {'link': link}

print(link)

#scrapy crawl link\_spider => Per eseguire sul terminale

33. Crea uno spider Scrapy che pagini attraverso un sito di e-commerce.

import scrapy

class TestSpider(scrapy.Spider):

name = "test"

allowed\_domains = ["corazonvegano.shop"]

start\_urls = ["https://corazonvegano.shop"]

def parse(self, response):

# Estrai i link dei prodotti dalla pagina corrente

product\_links = response.css('a.product-link::attr(href)').getall()

for product\_link in product\_links:

yield response.follow(product\_link, callback=self.parse\_product)

# Estrai il link alla prossima pagina

next\_page = response.css('a.next-page::attr(href)').get()

if next\_page:

yield response.follow(next\_page, callback=self.parse)

def parse\_product(self, response):

# Estrai i dettagli del prodotto dalla pagina del prodotto

product\_name = response.css('h1.product-name::text').get()

product\_price = response.css('span.product-price::text').get()

yield {

'name': product\_name,

'price': product\_price,

'url': response.url

}

34. Estrai e salva in un file JSON i dati estratti da uno spider Scrapy.

**Paso 1:** Eseguire il codice

import scrapy

class LinkSpider(scrapy.Spider):

name = "link\_spider"

start\_urls = ["https://ricette.giallozafferano.it/Finocchi-in-padella.html"]

def parse(self, response):

links = response.css('a::attr(href)').getall()

for link in links:

yield {'link': link}

**Paso 2**: Modificare il file setting.py aggiungendo e/o modificando il campo Feeds:

FEEDS = {

'items.json': {

'format': 'json',

'encoding': 'utf8',

'store\_empty': False,

'fields': None,

'indent': 4,

'item\_export\_kwargs': {

'export\_empty\_fields': True,

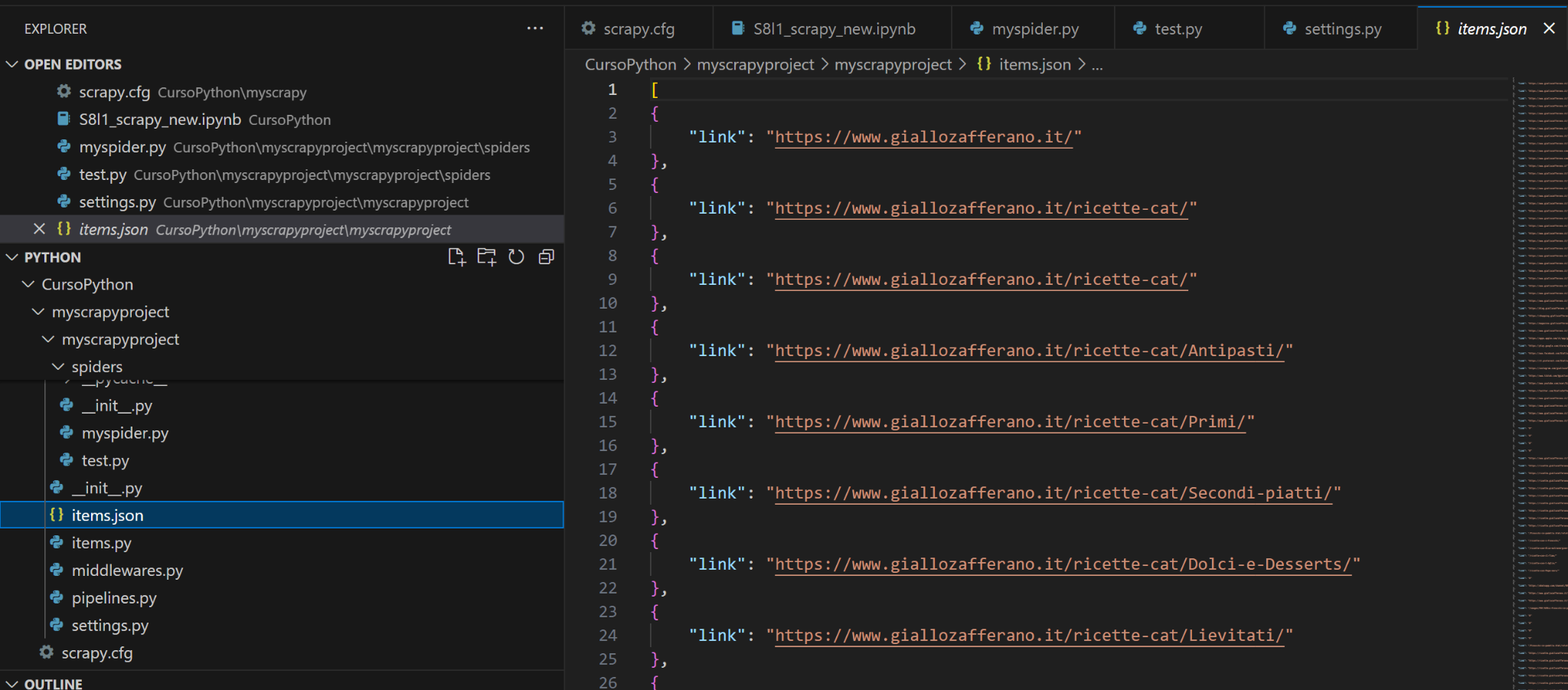
},

},

}

**Paso 3:** Dal terminale eseguire: scrapy crawl link\_spider

**Output:** Nella cartella principale del progetto si crea un file “items.json” con tutti i link ottenuti.



35. Crea uno spider Scrapy che utilizzi XPath per estrarre dati specifici da una pagina web.

import scrapy

class XPathSpider(scrapy.Spider):

name = "xpath\_spider"

start\_urls = [

'https://ricette.giallozafferano.it/Finocchi-in-padella.html',

]

def parse(self, response):

# Estrarre il titolo della pagina utilizzando XPath

title = response.xpath('//title/text()').get()

# Estrarre il testo di un paragrafo specifico utilizzando XPath

# paragraph\_text = response.xpath('//p[@class="gz-content-recipe gz-mBottom4x"]/text()').get()

# Estrarre i link delle immagini utilizzando XPath

image\_links = response.xpath('//img/@src').getall()

yield {

'title': title,

'paragraph\_text': paragraph\_text,

'image\_links': image\_links,

'url': response.url

}

36. Scrivi uno spider che estragga e stampi la metainformazione (meta tags) di una pagina web.

import scrapy

class MetaTagsSpider(scrapy.Spider):

name = "meta\_tags\_spider"

def start\_requests(self):

urls = [

'https://ricette.giallozafferano.it/Finocchi-in-padella.html',

# Aggiungi altri URL qui se vuoi estrarre le metainformazioni da più pagine

]

for url in urls:

yield scrapy.Request(url=url, callback=self.parse)

def parse(self, response):

meta\_tags = response.xpath('//meta')

for tag in meta\_tags:

name = tag.xpath('@name').get()

content = tag.xpath('@content').get()

yield {

'name': name,

'content': content

}