

◆ Modulo 3: Stringhe

10. String methods ~

Obiettivi di apprendimento:

- Metodi principali per manipolare stringhe
- Trasformazione del testo
- Ricerca e sostituzione
- Validazione stringhe

Contenuto teorico:

```
# Metodi di trasformazione
testo = " Ciao Mondo Python! "

print("==== METODI DI TRASFORMAZIONE ===")
print(f"Originale: '{testo}'")
print(f"Mayuscolo: '{testo.upper()}'")
print(f"Minuscolo: '{testo.lower()}'")
print(f"Capitalizzato: '{testo.capitalize()}'")
print(f"Title Case: '{testo.title()}'")
print(f"Swap Case: '{testo.swapcase()}'")

# Metodi di pulizia
print(f"\n==== METODI DI PULIZIA ===")
print(f"Strip (rimuove spazi): '{testo.strip()}'")
print(f"Strip sinistro: '{testo.lstrip()}'")
print(f"Strip destro: '{testo.rstrip()}'")

# Metodi di ricerca
print(f"\n==== METODI DI RICERCA ===")
print(f"Trova 'Python': {testo.find('Python')}") 
print(f"Conta 'o': {testo.count('o')}") 
print(f"Inizia con 'Ciao': {testo.strip().startswith('Ciao')}") 
print(f"Finisce con '!': {testo.strip().endswith('!')}")

# Metodi di sostituzione
print(f"\n==== METODI DI SOSTITUZIONE ===")
print(f"Sostituisci 'Python' con 'Java': '{testo.replace('Python', 'Java')}')

# Metodi di divisione e unione
frase = "mela,banana,arancia,uva"
print(f"\n==== DIVISIONE E UNIONE ===")
print(f"Originale: {frase}")
frutti = frase.split(',')
print(f"Diviso: {frutti}")
print(f"Riunito: {' - '.join(frutti)}")

# Metodi di validazione
```

```
print(f"\n==== METODI DI VALIDAZIONE ===")
test_strings = ["123", "abc", "123abc", "ABC", "abc123"]

for s in test_strings:
    print(f'{s}: digit={s.isdigit()}, alpha={s.isalpha()}, alnum={s.isalnum()}, upper={s.isupper()}, lower={s.islower()}'")
```

Processore di testo avanzato:

```
def elabora_testo():
    print("📝 ELABORATORE DI TESTO 📝")

    testo = input("Inserisci un testo: ")

    # Statistiche base
    print(f"\n==== STATISTICHE ===")
    print(f'Lunghezza: {len(testo)} caratteri')
    print(f'Parole: {len(testo.split())} parole')
    print(f'Linee: {len(testo.splitlines())} linee')

    # Analisi caratteri
    lettere = sum(1 for c in testo if c.isalpha())
    numeri = sum(1 for c in testo if c.isdigit())
    spazi = sum(1 for c in testo if c.isspace())

    print(f'Lettere: {lettere}')
    print(f'Numeri: {numeri}')
    print(f'Spazi: {spazi}')

    # Trasformazioni
    print(f"\n==== TRASFORMAZIONI ===")
    print(f'Maiuscolo: {testo.upper()}')
    print(f'Minuscolo: {testo.lower()}')
    print(f'Capitalizzato: {testo.capitalize()}')
    print(f'Invertito: {testo[::-1]}')

    # Pulizia
    testo_pulito = testo.strip().replace(" ", " ")
    print(f'Pulito: '{testo_pulito}'')

elabora_testo()
```

Validatore di input:

```
def valida_input():
    print("✅ VALIDATORE DI INPUT ✅")

    # Validazione nome
    while True:
```

```

nome = input("Nome (solo lettere): ").strip()
if nome.isalpha():
    print(f"✓ Nome valido: {nome.title()}")
    break
else:
    print("✗ Il nome deve contenere solo lettere!")

# Validazione età
while True:
    eta_str = input("Età (solo numeri): ").strip()
    if eta_str.isdigit():
        eta = int(eta_str)
        if 0 <= eta <= 120:
            print(f"✓ Età valida: {eta} anni")
            break
        else:
            print("✗ L'età deve essere tra 0 e 120!")
    else:
        print("✗ L'età deve essere un numero!")

# Validazione email semplice
while True:
    email = input("Email: ").strip().lower()
    if "@" in email and "." in email:
        print(f"✓ Email valida: {email}")
        break
    else:
        print("✗ Email non valida!")

valida_input()

```

Esercizi pratici:

1. Contatore di parole e caratteri
 2. Generatore di username da nome e cognome
 3. Sistema di censura parole
-

11. String indexing

Obiettivi di apprendimento:

- Accesso ai caratteri tramite indici
- Slicing delle stringhe
- Indici negativi
- Step nel slicing

Contenuto teorico:

```
# String indexing base
testo = "Python Programming"
```

```

print(f"Testo: '{testo}'")
print(f"Lunghezza: {len(testo)}")

print("\n==== INDEXING BASE ====")
print(f"Primo carattere [0]: '{testo[0]}'")
print(f"Secondo carattere [1]: '{testo[1]}'")
print(f"Ultimo carattere [-1]: '{testo[-1]}'")
print(f"Penultimo carattere [-2]: '{testo[-2]}'")

# Visualizzazione indici
print("\n==== MAPPA DEGLI INDICI ====")
print("Indici positivi:")
for i, char in enumerate(testo):
    print(f"[{i}:2]: '{char}'")

print("\nIndici negativi:")
for i, char in enumerate(testo):
    print(f"[{i}-len(testo):2]: '{char}'")

# Slicing
print("\n==== SLICING ====")
print(f"Primi 6 caratteri [0:6]: '{testo[0:6]}'")
print(f"Dal carattere 7 alla fine [7:]: '{testo[7:]}'")
print(f"Fino al carattere 5 [:6]: '{testo[:6]}'")
print(f"Ultimi 11 caratteri [-11:]: '{testo[-11:]}'")
print(f"Dal 2° al 5° carattere [2:6]: '{testo[2:6]}'")

# Slicing con step
print("\n==== SLICING CON STEP ====")
print(f"Ogni 2 caratteri [::2]: '{testo[::2]}'")
print(f"Ogni 3 caratteri [::3]: '{testo[::3]}'")
print(f"Stringa invertita [::-1]: '{testo[::-1]}'")
print(f"Primi 6 al contrario [5::-1]: '{testo[5::-1]}'")

```

Analizzatore di stringhe:

```

def analizza_stringa():
    print("🔍 ANALIZZATORE DI STRINGHE 🔎")

    testo = input("Inserisci una stringa: ")

    if not testo:
        print("Stringa vuota!")
        return

    print(f"\n==== ANALISI DI: '{testo}' ====")
    print(f"Lunghezza: {len(testo)}")

    # Caratteri speciali
    print(f"Primo carattere: '{testo[0]}'")
    print(f"Ultimo carattere: '{testo[-1]}'")

```

```

if len(testo) >= 2:
    print(f"Primi due: '{testo[:2]}''")
    print(f"Ultimi due: '{testo[-2:]}''")

if len(testo) >= 3:
    print(f"Carattere centrale: '{testo[len(testo)//2]}''")

# Palindromo check
testo_pulito = testo.lower().replace(" ", "")
è_palindromo = testo_pulito == testo_pulito[::-1]
print(f"È palindromo: {è_palindromo}")

# Estrai vocali e consonanti
vocali = "aeiouAEIOU"
vocali_trovate = [c for c in testo if c in vocali]
consonanti_trovate = [c for c in testo if c.isalpha() and c not in vocali]

print(f"Vocali: {vocali_trovate}")
print(f"Consonanti: {consonanti_trovate}")

analizza_stringa()

```

Giochi con le stringhe:

```

def giochi_stringhe():
    print("🎮 GIOCHI CON LE STRINGHE 🎮")

    # 1. Acronimo generator
    frase = input("Inserisci una frase per l'acronimo: ")
    parole = frase.split()
    acronimo = ''.join([parola[0].upper() for parola in parole if parola])
    print(f"Acronimo: {acronimo}")

    # 2. Alternating case
    testo = input("Inserisci testo per alternating case: ")
    alternato = ''.join([c.upper() if i % 2 == 0 else c.lower()
                         for i, c in enumerate(testo)])
    print(f"Alternato: {alternato}")

    # 3. Estrai numeri
    stringa_mista = input("Inserisci stringa con numeri: ")
    numeri = ''.join([c for c in stringa_mista if c.isdigit()])
    print(f"Numeri estratti: {numeri}")

    # 4. Cipher semplice (Caesar cipher)
    testo_cipher = input("Inserisci testo da cifrare: ")
    shift = 3
    cifrato = ''
    for char in testo_cipher:
        if char.isalpha():
            ascii_offset = 65 if char.isupper() else 97
            cifrato += chr((ord(char) - ascii_offset + shift) % 26 + ascii_offset)
        else:
            cifrato += char
    print(f"Cifrato: {cifrato}")

```

```

        cifrato += chr((ord(char) - ascii_offset + shift) % 26 +
ascii_offset)
    else:
        cifrato += char
print(f"Testo cifrato: {cifrato}")

giochi_stringhe()

```

Esercizi pratici:

1. Estrattore di informazioni da codice fiscale
 2. Validatore di formato (telefono, targa, etc.)
 3. Generatore di password con pattern
-

12. Format specifiers

Obiettivi di apprendimento:

- f-strings (formatted string literals)
- Metodo .format()
- Formattazione di numeri, date, allineamento
- Format specifiers avanzati

Contenuto teorico:

```

# F-strings (Python 3.6+) – Metodo moderno e preferito
nome = "Mario"
eta = 25
altezza = 1.75
stipendio = 2500.50

print("== F-STRINGS ==")
print(f"Nome: {nome}")
print(f"Età: {eta} anni")
print(f"Altezza: {altezza} metri")
print(f"Stipendio: €{stipendio}")

# Formattazione numerica
pi = 3.14159265359
numero_grande = 1234567.89

print(f"\n== FORMATTAZIONE NUMERICA ==")
print(f"Pi con 2 decimali: {pi:.2f}")
print(f"Pi con 4 decimali: {pi:.4f}")
print(f"Numero grande con separatori: {numero_grande:,.2f}")
print(f"Percentuale: {0.85:.1%}")
print(f"Notazione scientifica: {numero_grande:.2e}")

# Allineamento e padding
print(f"\n== ALLINEAMENTO ==")

```

```

parole = ["Python", "Java", "C++", "JavaScript"]
for parola in parole:
    print(f"Sinistra: '{parola:<15}' | Centro: '{parola:^15}' | Destra: '{parola:>15}'")

# Padding con caratteri personalizzati
print(f"Padding con zero: {42:05d}")
print(f"Padding con trattini: {nome:-^20}")

# Formattazione di date
from datetime import datetime
ora_attuale = datetime.now()

print(f"\n==== DATE E ORARI ====")
print(f"Data completa: {ora_attuale:%d/%m/%Y %H:%M:%S}")
print(f"Solo data: {ora_attuale:%d-%m-%Y}")
print(f"Solo ora: {ora_attuale:%H:%M}")
print(f"Giorno della settimana: {ora_attuale:%A}")
print(f"Mese: {ora_attuale:%B}")

```

Metodo .format() (alternativa):

```

# Metodo .format() – Compatibile con versioni precedenti
print("\n==== METODO .FORMAT() ====")
template = "Nome: {}, Età: {}, Stipendio: €{:.2f}"
print(template.format("Luigi", 30, 2800.75))

# Con indici
template_indici = "Il {} costa €{1:.2f}. {} è in offerta!"
print(template_indici.format("laptop", 899.99))

# Con nomi
template_nomi = "Ciao {}, hai {} anni e vivi a {}"
print(template_nomi.format(nome="Anna", eta=28, citta="Roma"))

```

Sistema di report avanzato:

```

def genera_report():
    print("📊 GENERATORE DI REPORT 📊")

    # Dati di esempio
    vendite = [
        {"prodotto": "Laptop", "quantita": 15, "prezzo": 899.99},
        {"prodotto": "Mouse", "quantita": 50, "prezzo": 25.50},
        {"prodotto": "Tastiera", "quantita": 30, "prezzo": 75.00},
        {"prodotto": "Monitor", "quantita": 8, "prezzo": 299.99}
    ]

    # Header del report

```

```

print("=" * 70)
print(f"{'REPORT VENDITE':^70}")
print("=" * 70)
print(f"{'Prodotto':<15} {'Quantità':>10} {'Prezzo':>12}")
{'Totale':>15} {'%':>8}")
print("-" * 70)

totale_generale = 0
for item in vendite:
    totale_item = item['quantita'] * item['prezzo']
    totale_generale += totale_item

# Rigue del report
for item in vendite:
    prodotto = item['prodotto']
    quantita = item['quantita']
    prezzo = item['prezzo']
    totale_item = quantita * prezzo
    percentuale = (totale_item / totale_generale) * 100

    print(f"{{prodotto:<15} {quantita:>10d} €{prezzo:>10.2f} €
{totale_item:>12.2f} {percentuale:>6.1f}%)")

print("-" * 70)
print(f"{'TOTALE GENERALE':<40} €{totale_generale:>12.2f}")
print("=" * 70)

genera_report()

```

Formatter personalizzati:

```

def esempi_formatting_avanzato():
    print("🎭 FORMATTING AVANZATO 🎭")

    # Numeri binari, ottali, esadecimali
    numero = 255
    print(f"Decimale: {numero}")
    print(f"Binario: {numero:b}")
    print(f>Ottale: {numero:o}")
    print(f"Esadecimale: {numero:x}")
    print(f"Esadecimale maiuscolo: {numero:X}")

    # Segni e spazi
    positivo = 42
    negativo = -42
    print(f"\nCon segno sempre: {positivo:+d}, {negativo:+d}")
    print(f"Spazio per positivi: {positivo: d}, {negativo: d}")

    # Formattazione condizionale
    voti = [95, 87, 76, 45, 92]
    print(f"\n==== VOTI ===")
    for voto in voti:

```

```
status = "PASS" if voto >= 60 else "FAIL"
colore = "□" if voto >= 60 else "●"
print(f"Voto: {voto:3d}/100 {colore} {status}")

# Template per email
template_email = """
Gentile {nome},

La sua prenotazione per {evento} è confermata.
Data: {data:%d/%m/%Y}
Ora: {data:%H:%M}
Prezzo: €{prezzo:.2f}

Codice prenotazione: {codice}

Cordiali saluti,
Staff Eventi
"""

from datetime import datetime
dati = {
    'nome': 'Mario Rossi',
    'evento': 'Concerto Rock',
    'data': datetime(2024, 6, 15, 21, 0),
    'prezzo': 45.50,
    'codice': 'MR2024001'
}

print(template_email.format(**dati))

esempi_formatting_avanzato()
```

Esercizi pratici:

1. Sistema di fatturazione con formattazione
2. Dashboard di monitoraggio sistema
3. Generatore di certificati personalizzati