

Fondamenti di Python — Descrizione ed Esempi

◆ 1. Variables (Variabili)

Descrizione

Una **variabile** è un contenitore che memorizza un valore in memoria.
In Python non è necessario dichiarare il tipo: viene assegnato automaticamente.

Esempi


```
price = 2.50
product = "Milk"
available = True

print(price)
print(product)
print(available)
```

◆ 2. Variable Names (Nomi delle Variabili)

Descrizione

I nomi delle variabili devono:

- iniziare con una lettera o 
- non contenere spazi
- non usare parole chiave Python
- essere **descrittivi**

Python usa la convenzione **snake_case**.

Corretto

```
total_price = 15.75  
user_name = "Alice"
```

Errato

```
2price = 10      # inizia con numero  
user-name = "Bob" # trattino non consentito
```

◆ 3. Strings (Stringhe)

Descrizione

Una **stringa** è una sequenza di caratteri racchiusa tra:

- `" "`
- `' '`

Esempi

```
name = "Apple"  
category = 'Fruit'  
  
print(name)  
print(category)
```

◆ 4. Escape Sequences

Descrizione

Le **sequenze di escape** permettono di inserire caratteri speciali nelle stringhe.

Sequenza	Significato
<code>\n</code>	Nuova riga
<code>\t</code>	Tab
<code>\"</code>	Virgolette
<code>\\</code>	Backslash

Esempi

```
print("Hello\nWorld")  
print("Item:\tMilk")  
print("She said: \"Hello\"")
```


◆ 5. Formatted Strings (f-strings)

Descrizione

Le **f-string** permettono di inserire variabili direttamente nelle stringhe. Sono più leggibili ed efficienti.

Esempi

```
item = "Bread"  
price = 1.20  
  
print(f"The price of {item} is ${price}")
```

◆ 6. String Methods (Metodi delle Stringhe)

Descrizione

Le stringhe hanno **metodi integrati** per manipolarle.

Metodo	Funzione
<code>.upper()</code>	Maiuscolo
<code>.lower()</code>	Minuscolo
<code>.title()</code>	Prima lettera maiuscola
<code>.find()</code>	Trova posizione
<code>.replace()</code>	Sostituisce testo
<code>.strip()</code>	Rimuove spazi

Esempi

```
text = "  apple pie  "

print(text.upper())
print(text.strip())
print(text.replace("apple", "cherry"))
```

◆ 7. Numbers (Numeri)

📌 Descrizione

Python supporta diversi tipi numerici:

Tipo	Esempio
<code>int</code>	10
<code>float</code>	3.14
<code>complex</code>	2 + 3j

Esempi

```
quantity = 10
price = 2.75

total = quantity * price
print(total)
```

◆ 8. Working with Numbers (Operazioni sui Numeri)

Descrizione

Python consente operazioni matematiche standard:

Operatore	Significato
<div>+</div>	somma
<div>-</div>	sottrazione
<div>*</div>	moltiplicazione
<div>/</div>	divisione
<div>//</div>	divisione intera
<div>%</div>	resto
<div>**</div>	potenza



Esempi

```
x = 10
y = 3

print(x + y)
print(x // y)
print(x ** y)
```

◆ 9. Type Conversion (Conversione di Tipo)

Descrizione

La **type conversion** permette di trasformare un valore da un tipo a un altro.

Funzione	Converte in
<code>int()</code>	intero
<code>float()</code>	decimale
<code>str()</code>	stringa

Esempi

```
price = "3.99"  
quantity = "4"  
  
total = float(price) * int(quantity)  
print(total)
```

Riepilogo Competenze

- ✓ Uso corretto delle variabili
- ✓ Gestione delle stringhe
- ✓ Output formattato
- ✓ Operazioni numeriche
- ✓ Conversione dei tipi



Esercizi Python — Fondamenti

◆ 1. Variables (Variabili)

Esercizi

1. Crea una variabile `store_name` con valore `"Fresh Market"` e stampala.
2. Assegna un numero alla variabile `items_in_stock` e stampala.
3. Cambia il valore di una variabile dopo la sua creazione.
4. Crea due variabili numeriche e stampane la somma.
5. Assegna un valore booleano a una variabile e stampala.

◆ 2. Variable Names (Nomi delle Variabili)

Esercizi

1. Individua quale nome di variabile è corretto:

- `2price` , `total_price` , `total-price`

2. Correggi i nomi errati:

```
user Name = "Mario"
```

3. Rinomina una variabile per renderla più descrittiva.

4. Scrivi 3 variabili usando lo stile `snake_case` .

5. Spiega perché `class = 10` non è valido.

◆ 3. Strings (Stringhe)

Esercizi

1. Crea una stringa con il nome di un prodotto.
2. Stampa una stringa usando virgolette singole.
3. Unisci due stringhe usando `+`.
4. Stampa la lunghezza di una stringa con `len()`.
5. Assegna una frase a una variabile e stampala.

◆ 4. Escape Sequences

Esercizi

1. Stampa una frase su due righe usando `\n`.
2. Stampa una tabulazione tra due parole.
3. Stampa una frase con virgolette al suo interno.
4. Stampa un percorso file usando `\\`.
5. Scrivi una frase che contenga `\n` e `\t`.

◆ 5. Formatted Strings (f-strings)

Esercizi

1. Usa una f-string per stampare nome e prezzo di un prodotto.
2. Stampa un totale usando due variabili numeriche.
3. Crea una frase che includa un valore booleano.
4. Usa una f-string con un'espressione matematica.
5. Stampa una frase con una variabile stringa e una numerica.

◆ 6. String Methods (Metodi delle Stringhe)

Esercizi

1. Converti una stringa in maiuscolo.
2. Rimuovi spazi all'inizio e alla fine di una stringa.
3. Sostituisci una parola in una frase.
4. Trova la posizione di una lettera in una stringa.
5. Trasforma una stringa in formato titolo.

◆ 7. Numbers (Numeri)

Esercizi

1. Crea una variabile `int` e una `float` .
2. Moltiplica due numeri e stampa il risultato.
3. Sottrai due numeri.
4. Stampa il tipo di una variabile usando `type()` .
5. Assegna il risultato di un calcolo a una variabile.

◆ 8. Working with Numbers (Operazioni Numeriche)

Esercizi

1. Calcola il resto di una divisione.
2. Usa l'operatore `**` per una potenza.
3. Usa la divisione intera `//`.
4. Calcola il totale di un acquisto.
5. Scrivi un'espressione che usi almeno 3 operatori.

◆ 9. Type Conversion (Conversione di Tipo)

Esercizi

1. Converti una stringa numerica in `int`.
2. Converti un numero in stringa.
3. Moltiplica una stringa convertita in numero.
4. Usa `input()` e converti il valore in `int`.
5. Spiega cosa succede se converti `"abc"` in `int`.

Sfida Finale (Opzionale)

Crea un mini programma che:

- definisce il nome di un prodotto
- chiede la quantità all'utente
- converte l'input in numero
- calcola il totale
- stampa il risultato con una f-string