

Espressioni e numeri

- Espressioni
- Valutazione
- Numeri (int e float)
 - i numeri sono **oggetti**
 - Tipo (o classe) di un numero (funzione type)
- La divisione restituisce sempre un numero con la virgola, le altre dipende, se sono tutti interi restituisce un intero, altrimenti un numero con la virgola
- Operatori aritmetici (+, -, *, /)
- Regole di precedenza
- Parentesi tonde
- Altri operatori (**, //, %)
- Regole di precedenza di questi nuovi operatori (** precedenza massima, divisione intera e resto come il *)

Esercizi

Area del trapezio

Sapendo che l'area del trapezio si calcola in questo modo:

$$\frac{(base_{maggiore} + base_{minore}) \cdot altezza}{2}$$

calcola l'area del trapezio avente le seguenti misure:

$$base_{maggiore} = 14.8$$

$$base_{minore} = 10.9$$

$$altezza = 6.1$$

Quanti mesi hai?

Calcola quanti mesi di vita hai vissuto

Tradurre le seguenti espressioni aritmetiche

1. Moltiplica per 4 la differenza fra il quadrato di 5 e il quadrato di 3
2. Eleva al quadrato la somma del cubo di 2 e del quadrato di 1, poi dividi il risultato per 27

Espressioni che non resituiscono numeri

- Le espressioni aritmetiche resituiscono oggetti di tipo numerico (int e float)
- Ma ci sono anche espressioni che restituiscono oggetti di tipo diverso
- Confrontiamo due valori verificando che uno sia maggiore dell'altro
- Il tipo **bool**
- Operatori di confronto ($>$, $<$, $==$, $<=$, $>=$, $!=$)
- Si possono confrontare valori, ma anche espressioni:
 - $(4 * 5 + 3) == (7 + 16)$

Esercizi

- Scrivere un'espressione che restituisca **True**
- Scrivere un'espressione che restituisca **False**

Variabili

- Necessità di salvare i dati da qualche parte
- variabili ben formate:
 - nome (lettere, numeri, _)
 - simboli ammessi
 - parole riservate
 - lunghezza
 - convenzioni
- operatore di assegnamento (=)
 - tramite questo operatore associamo un oggetto (un numero o *qualunque* altro oggetto) a una variabile
 - assegnazioni multiple
- cosa possiamo assegnare ad una variabile?
- l'assegnamento avviene da destra a sinistra, prima python *valuta* l'espressione a destra, poi assegna il valore valutato alla variabile
- in python non esistono variabili propriamente dette, ma **etichette**
- il valore associato a una variabile può cambiare, il suo nome no!
- operazioni fra variabili
- stampa di un valore tramite **print()**

Esercizi

assegnamento e operazioni fra variabili

Assegna a 2 variabili chiamate **base** e **altezza** i seguenti valori: 12.8 e 13.6 poi calcola l'area del rettangolo e assegnala alla variabile **area**. Stampa il risultato con la funzione **print()**

Area del Trapezio

Calcola l'area del trapezio dell'esercizio precedente assegnando i valori alle variabili **base_maggiore**, **base_minore**, **altezza** e infine il calcolo dell'area alla variabile **area**

Usa le variabili per risolvere il seguente problema

Se un violino costa 500 euro, quanto lo pagherò se ottengo uno sconto del 30%?

Le Stringhe

- Cosa sono le stringhe?
- il tipo **str**
- Tipi di delimitatori (",')
- Includere un delimitatore in una stringa
- operatori sulle stringhe (*, +)
- usare print() con le stringhe
- caratteri speciali (\n, \t)

Esercizi

Scrivi le istruzioni per stampare 5 volte la stringa "Buon Compleanno"

Scrivi la sequenza di istruzioni per ottenere il perimetro e l'area di un rettangolo che abbia i lati di cm 9 e cm 5 e infine stampa la seguente stringa: "Dato un rettangolo con lati 10 e 5, il perimetro sarà uguale a ... e l'area a ...". Naturalmente sostituisci i puntini con i giusti valori

Funzioni sulle stringhe

- **len()**: lunghezza

Introspezione

Python consente di fare introspezione, cioè di ottenere informazioni sugli oggetti presenti in un programma.

Importanti funzioni di introspezione:

- identità di un oggetto: **id()**
- tipo di un oggetto: **type()**
- elenco degli attributi: **dir()**
- descrizione di una funzione o di un attributo: **help()**

Ogni tipo definisce un insieme di attributi, alcuni dei quali sono dati *accessorii* dell'oggetto, altri sono *chiamabili*, cioè sono delle vere e proprie operazioni da compiere sull'oggetto. Questi attributi *operazionali* sono detti **metodi**. Per sapere se un attributo di un oggetto è un metodo, si può chiamare l'attributo con la funzione *callable*:

```
callable(str.capitalize)
```

che resituirà **True** o **False**.

Metodi importanti delle stringhe

Alcuni metodi restituiscono informazioni su una stringa:

- **upper()**: restituisce una stringa tutta maiuscola
- **lower()**: restituisce una stringa tutta minuscola
- **replace()**: sostituisce uno o più caratteri con altri
- **count()**: conta il numero di occorrenze di un caratteri o di una sequenza di caratteri
- **isdigit()**: restituisce **True** se la stringa contiene delle cifre

esempi

```
# str.upper()
stringa = "sheldon cooper"
print(stringa.upper())
>>> SHELDON COOPER
```

```
# str.lower()
stringa = "ShElD0n cooPer"
print(stringa.lower())
>>> sheldon cooper
```

```
# str.replace()
stringa = "sheldon cooper"
print(stringa.replace('o','a'))
>>> sheldan caaper
```

```
# str.count()
stringa = "sheldon cooper"
print(stringa.count('o'))
>>> 3
```

```
# str.isdigit()
stringa = "sheldon cooper"
print(stringa.isdigit())
>>> False
```

```
# str.isdigit()
stringa = "23"
print(stringa.isdigit())
>>> True
```

I metodi **NON** modificano la stringa originale, ma creano una **NUOVA** stringa modificata. Gli oggetti di tipo **str** infatti sono **IMMUTABILI**, come del resto gli **int** e i **float**.

Funzioni importanti delle stringhe

input()

Permette di *catturare* una sequenza di caratteri immessi da tastiera. Permette di stampare una stringa di descrizione del testo da immettere e restituisce a sua volta la stringa immessa da tastiera.

```
input("scrivi il tuo nome: ")
```

len()

Restituisce la lunghezza di una stringa, cioè la quantità di caratteri che la compongono.

Conversione

La conversione di *tipo* permette di convertire un oggetto di un tipo in un oggetto di un altro tipo. Ad esempio un numero di tipo *float* può essere convertito in un oggetto di tipo *int*:

```
>>> numero = 13.8
>>> numero_convertito = int(numero)
>>> print(numero_convertito)
>>> 13
```

Le stringhe possono essere convertite in numeri. Se una stringa contiene delle cifre, possiamo convertirla in numero intero con **int()**. Se invece contiene cifre separate da un punto, possiamo convertirla in **float**.

```
>>> stringa = '13'
>>> print(type(stringa))
>>> <class 'str'>
>>> numero = int(stringa)
>>> print(type(numero))
>>> <class 'int'>
```

Il meccanismo della conversione consente di ottenere valori numerici da tastiera tramite la funzione `input()` e di convertirli con `int()` e `float()` in un valore di tipo numerico.

esempio:

```
>>> anno_di_nascita = input("quando sei nato? ")
>>> anno_attuale = 2021
>>> eta = anno_attuale - anno_di_nascita
>>> print("Hai", eta, "anni!")
```