

ancora-su-stringhe-e-numeri

June 5, 2023

0.1 Ancora su stringhe e numeri

Abbiamo costruito espressioni con numeri o con stringhe, utilizzando vari operatori. Naturalmente ci sono molti altri operatori per manipolare i numeri.

0.1.1 Esponenziazione

L'elevazione a potenza si esprime con due asterischi attaccati e funziona con esponenti positivi e negativi, interi o frazionari.

```
[ ]: 12**(4+3)
```

```
[ ]: 12**(0.5)
```

```
[ ]: -4**12
```

```
[ ]: 4**(-2)
```

0.1.2 Divisione intera e resto

Abbiamo visto che l'operatore `/` è la divisione tra numeri. Se viene usata con un divisore che con divide esattamente il dividendo, allora produce un un numero con parte decimale. È possibile avere però la **divisione intera** ovvero la divisione che produce sempre un risultato intero, ma che implica un resto. Per avere questo resto si usa invece l'operatore `%`.

```
[ ]: 5 / 3
```

```
[ ]: 10 / 4
```

```
[ ]: 5 // 3
```

```
[ ]: 10 // 4
```

```
[ ]: 5 % 3
```

```
[ ]: 10 % 4
```

```
[ ]: (12 * 7 + 5) % 7
```

Esercizio: cosa succede ad usare l'operatore di divisione intera con dei numeri non interi?

0.1.3 Funzioni (matematiche?)

È possibile applicare delle funzioni ai valori. Le funzioni sono identificate da un nome e si utilizzano in un'espressione con la sintassi

```
[ ]: nomefunzione(param1,param2,param3,...)
```

dove il numero di parametri dipende dalla funzione utilizzata, e per alcune funzioni il numero di parametri è libero.

Vediamo ad esempio - **abs** prende la versione positiva del numero - **min,max** abbastanza autoesplicativi

```
[ ]: abs(-2.1)
```

```
[ ]: abs(4.23)
```

```
[ ]: min(23, 5, 3, 1, 7, 8, 784, 2.6, 4-5, 6)
```

```
[ ]: max(23, 5, 3, 1, 7, 8, 784, 2.6, 4-5, 6)
```

0.1.4 Costruzione delle espressioni

Tutte queste operazioni e funzioni sono componibili a piacimento. Ed è possibile scrivere espressioni arbitrariamente complicate. Vediamo ad esempio:

```
[ ]: abs(-3.5) + max( 32 % 3, 12 // 7 , 12.5 - abs(13.2) ) + 83 / 4
```

Esercizio: riuscite a calcolare il valore dell'espressione senza usare python? Dopo che ci avete provato per un po', verificate nella console sottostante.

0.1.5 Precedenza degli operatori

Per quanto complessa sia un'espressione, il suo valore è sempre ben definito. Questo perché esiste un ordine preciso con cui le operazioni vengono effettuate, grazie alle regole di precedenza degli operatori. L'ordine di precedenza è

- parentesi e funzioni
- esponenziazioni (da destra a sinistra)
- divisione e moltiplicazioni (da sinistra a destra)
- somme e sottrazioni (da sinistra a destra)

```
[ ]: 3**3**3
```

```
[ ]: 3**(3**3)
```

```
[ ]: (3**3)**3
```

```
[ ]: 4 * 12 + 4
```

```
[ ]: (4 * 12) + 4
```

```
[ ]: 4 * (12 + 4)
```

0.1.6 E per le stringhe?

Naturalmente in python esistono funzioni di tutti i tipi. E ce ne sono anche per le stringhe, tuttavia per le stringhe sono più comuni i cosiddetti **metodi**, che sono delle funzioni con delle particolarità aggiuntive. Vediamo la funzione `len` che calcola la lunghezza di una stringa, e i metodi `upper` e `lower`.

```
[ ]: len('la vispa teresa')
```

```
[ ]: len("cacio" + "cavallo")
```

```
[ ]: len('  gli  spazi  contano  ')
```

```
[ ]: len('l\'apice è un carattere solo')
```

```
[ ]: 5 + len("aldo rossi") / 2
```

```
[ ]: "Ciao".upper()
```

```
[ ]: "Arrivederci Roma".lower()
```

```
[ ]: ("Buona" + " " + "Giornata").lower()
```

Osservazione: poiché `len` riceve una stringa come parametro ma produce un intero, l'espressione `len("aldo rossi")` può essere utilizzata insieme ad altri numeri, mentre l'espressione `5 + "aldo rossi" / 3` causa un errore.

0.1.7 Differenza tra funzioni e metodi

Una funzione è chiamata così

```
funzione(param1,param2,...)
```

un metodo invece si richiama così

```
expr.metodo(param1,param2,...)
```

In particolare esiste una sola funzione `len`, mentre potete pensare che si sia un metodo `upper` associato ad ogni stringa, e in effetti nonostante `upper` non abbia parametri, il risultato che produce **dipende** dall'oggetto su a cui è associato.

Pensate ad una funzione come un macchinario nel quale mettete dei dati, mentre metodo invece è una specie di gadget che è associato al dato stesso.

0.1.8 Riassumendo

Abbiamo visto altre operazioni sui numeri - esponenziazione - divisione intera e resto - l'uso di funzioni `abs`, `min`, `max` - precedenza tra operatori - funzione `len` e metodi `upper` e `lower` - differenza tra metodi e funzioni.