Python 3: Cheat Sheet Pratica in Italiano

Operazioni Numeriche

```
2 + 2     # addizione
4 - 3     # sottrazione
5 * 5     # moltiplicazione
6 / 3     # divisione
5 ** 3    # calcolo esponenziale
30 // 7   # quoziente
30 % 7    # resto
```

Tipi di Dato Principali

```
1, 22, 35                 # Intero
3.14, 5.55, 4.20          # Float
True, False               # Booleano
"spam", 'eggs'            # Stringa
[8, "elon", 3.14]         # Lista
(1, 1)                    # Tupla
{1: "Mark", 2: "Steve"}   # Dizionario
{0, 4, 2, 1}              # Set
None                      # None
```

Variabili

Le variabili sono dei "contenitori" che usiamo per salvare e richiamare valori. Per dichiarare una variabile non è necessario alcun comando speciale, e possiamo assegnare a queste qualsiasi tipo di dato:

```
name = "Python"
pi = 3.14
```

Operatori di Confronto

```
==  # Uguale a
!=  # Non uguale a
<   # Minore di
>   # Maggiore di
<=  # Minore o uguale a
>=  # Maggiore o uguale a
```

Operatori Booleani

```
and
2 > 1 and 5 != 7

or
4 == 5 or 5 == 6

not
not 3 == 3
```

Indentazione

In Python l'indentazione è usata per definire i blocchi di codice dei nostri programmi.

Convenzionalmente, un livello di indentazione è rappresentato da un TAB, o 4 spazi: dai uno sguardo alla sezione sul Controllo di Flusso.
NB: In Python le parentesi graffe non sono usate!

Controllo di Flusso: If, Elif, Else

Queste istruzioni vengono usate per stabilire le condizioni necessarie all'esecuzione di un blocco di codice

```
if espressione_di_controllo_uno:
    # codice da eseguire
elif espressione_di_controllo_due:
    # codice da eseguire
else:
    # codice da eseguire
```

```
age = 18
patente = False

if age >= 18 and patente == True:
    print('Puoi noleggiare una Ferrari!')
elif age >= 18 and patente == False:
    print('Senza patente niente Ferrari!')
else:
    print('Torna tra qualche anno...')
```

Funzioni da Conoscere

```
print()
# manda in output un valore passato

input()
# accetta input dall'utente

type()
# restituisce il tipo di dato
corrispondente al parametro passato

len()
# restituisce la lunghezza di una lista
o stringa

str() | int() | float() | list()
# usate per convertire valori
rispettivamente in stringa, intero,
float e lista
```

Scrivi e Usa le Tue Funzioni

```
def nome_funzione(param1, param2):
    return param1 + param2

---

def somma(a, b):
    risultato = a + b
    return risultato


>>> somma(2, 3)
5
```

Il Ciclo While

Il codice nel ciclo while viene eseguito finché la condizione di controllo resta True

```
while condizione_di_controllo:
    # codice da eseguire

---

contatore = 0
while contatore <= 10:
    print(contatore)
    contatore = contatore + 1
```

Il Ciclo For

Il codice nel ciclo for viene eseguito per un numero esatto di cicli.
Possiamo usare la funzione range() per definire il numero di cicli o fare tanti cicli quanti elementi sono presenti in un oggetto iterabile

```
for numero in range(11):
    print(numero)

0
1
…
10

---

for element in ["spam", "bacon”, 15]:
    print(element)
```

L’Istruzione break

L'istruzione break serve per terminare un ciclo (while o for) prematuramente

```
contatore = 0
while True:
    print(contatore)
    contatore += 1
    if contatore > 10:
        print('Sto uscendo dal loop!')
        break
```

L’Istruzione continue

L'istruzione continue serve per saltare un loop del ciclo (while o for)

```
contatore = 0
while contatore < 10:
    contatore += 1
    if contatore == 3:
        print('saltato')
        continue
    print(contatore)
```

Principali Metodi delle Liste

```
nome_lista.metodo(eventuali_parametri)

.append()  # per aggiungere elementi
.remove()  # per rimuovere elementi
.sort()    # per ordinare una lista
.extend()  # per "unire" due liste
.index()   # ottieni l'indice di un elem.
.reverse() # inverte l'ordine degli elem.
.insert()  # agg. elem. a un dato indice
```

Gestione degli Errori

```
try:
    # codice da provare a eseguire
except Exception as e:
    # codice per gestire l'errore
finally:
    # blocco eseguito in ogni caso
```

Le Classi

```
class Persona:
    def __init__(self, nome, cognome):
        # metodo inizializzatore
        self.nome = nome
        self.cognome = cognome

    def profilo(self):
        # un esempio di metodo di classe
        print("Nome: " + self.nome)
        print("Cognome: " + self.cognome)

>>> p = Persona("Mario", "Rossi")
>>> p.profilo()

Nome: Mario
Cognome: Rossi
```

Principali Metodi delle Stringhe

```
.join()  # per unire assieme più stringhe
.split() # divide la stringa in più parti
.startswith() # verifica inizio stringa
.endswith()   # verifica fine stringa
.isalpha() # True se stringa di solo lettere
.isdecimal() # True se stringa di solo numeri
.isalnum() # True se stringa alfanumerica

.upper() # restituisce la versione in
maiuscolo della stringa
.lower() # restituisce la versione in
minuscolo della stringa
```

La Standard Library

La Standard Library è una raccolta di moduli inclusi nell'installazione di Python che mette a disposizione funzioni estremamente utili in vari contesti.

Elenco Completo: docs.python.org/3/library

Per importare un modulo usiamo import, seguito dal nome del modulo, che ci da accesso a tutte le funzioni e classi del modulo stesso:

```
>>> import math
>>> math.sqrt(25)
5.0
```

Per importare una classe o funzione specifica usiamo from:

```
>>> from math import sqrt
>>> sqrt(25)
5.0
```

Principali Metodi dei Dizionari

```
.keys() | .values() | .items()
# usati per ottenere rispettivamente gli
elenchi delle chiavi, dei valori, o delle
coppie chiave-valore del dizionario

.get(chiave, messaggio)
# restituisce il valore associato alla
chiave passata se questa esiste, o il
messaggio passato come secondo parametro

.setdefault(chiave, valore)
# restituisce il valore associata alla
chiave passata se questa esiste, oppure
crea la nuova coppia chiave-valore usando
il valore passato come secondo parametro
```

Appunti e Note Extra Personali

.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.

Appunti e Note Extra Personali

.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.

STAMPA QUESTE PAGINE E TIENILE A PORTATA DI MANO MENTRE IMPARI A PROGRAMMARE IN PYTHON! Se vuoi, condividi pure la versione digitale o cartacea di questo documento con i tuoi amici.

Offerto da:

www.ProgrammareInPython.it

Il Sito Web Italiano dedicato al Linguaggio di Programmazione Python