

Cheatsheet pro výuku v Pythonu

Co je to Python?

Python je oblíbený programovací jazyk s intuitivní syntaxí a čitelným kódem, vhodný pro mnoho různých aplikací. Kromě toho, díky svým knihovnám pro matematické a statistické výpočty, jako jsou NumPy, Pandas a SciPy, je Python také velmi užitečný **pro zpracování výsledků a analýzu dat**. To z něj dělá ideální nástroj pro vědce, analytiky dat a vývojáře, kteří hledají snadný způsob, jak pracovat s velkými datovými soubory. Navíc, díky velké podpoře ze strany komunity, je Python **zdarma**, a je **ideální pro začátečníky**, kteří chtějí začít psát kód bez učení se složitých nízkoúrovňových detailů.

Základní matematické operace

```
sčítání:          2 + 3          # 5
násobení:        2 * 5          # 10
desetinné dělení: 10 / 3        # 3.3333333333333335
celočíslné dělení: 10 // 3      # 3
modus:           10 % 3         # 1
závorkování:     (5 - 3) * (7 - 4) # 6
mocniny:          5**2          # 25
```

Proměnné

V Pythonu můžete vytvářet proměnné a přiřazovat jim hodnoty pomocí znaménka "=" (rovnítko). Například, pokud chcete vytvořit proměnnou "x" a přiřadit jí hodnotu 5, napíšete "x = 5". Poté můžete proměnnou "x" použít v kódu pro různé výpočty nebo operace.

```
x = 2           # do proměnné x přiřad číslo 2
y = 5           # do promenne y priřad cislo 5
x + y           # x + y = 2 + 5 = 7; výsledek je tedy 7
```

Řetězce (string)

Řetězec je sekvence znaků uzavřená v uvozovkách, která umožňuje práci s textem v kódu. Ukázky práce s textem:

```
a = "Hello, World! " # přiřazení řetězce do proměnné
a[1]                 # vypsání znaku na dané pozici; e
a*2                  # násobení řetězců; Hello, World! Hello, World!
len(a)               # zjištění délky řetězce; 14
"ello" in a          # jestli daný řetězec obsahuje jiný řetězec; True
```

Cykly

Cykly jsou v Pythonu struktury, které umožňují opakování kódu. *For* cyklus v Pythonu slouží k procházení prvků v sekvenci, jako jsou seznamy, řetězce nebo *n*-tice, a opakování kódu pro každý prvek. Například, pro výpis všech prvků v seznamu můžeme použít *for* cyklus:

```
my_list = [1, 2, 3, 4, 5]
for number in my_list:
    print(number)
```

While cyklus v Pythonu opakuje kód, dokud platí určitá podmínka. Například, dokud je proměnná "i" menší než 5, cyklus bude opakovat kód v těle *while* cyklu:

```
i = 0
while i < 5:
    print(i)
    i += 1
```

Funkce

Funkce jsou bloky kódu, které lze vyvolat z jiných částí kódu a které provádí určitou úlohu nebo vrací určitou hodnotu. Funkce v Pythonu se definují pomocí klíčového slova "def" a mohou přijímat argumenty (vstupní hodnoty) a vracet hodnoty pomocí klíčového slova "return". Například, následující funkce vypočítá druhou mocninu zadaného čísla:

```
def square(x):
    return x**2
```

Tuto funkci lze poté použít z jiné části kódu s nějakým vstupním argumentem:

```
result = square(5)
print(result) # vypíše 25
```

Funkce jsou užitečné pro opakovaně používaný kód, modularitu a snadnější čitelnost a údržbu kódu.

Knihovny

Knihovny v Pythonu jsou soubory kódu, které obsahují funkce a metody, které mohou být importovány do jiných částí kódu. Tyto knihovny usnadňují programování tím, že nabízejí hotová řešení pro běžné úlohy a operace. Například použití knihovny „math“:

```
import math
x = math.sin(math.radians(45))
print(x) # vypíše 0.7071067811865475
```