

Cheatsheet - Základní příkazy

Základní vestavěné funkce

```
len(objekt)          # délka objektu (string, list, tuple...)
type(objekt)         # typ objektu
input(text)          # načtení vstupu od uživatele
print(hodnota)       # výpis na obrazovku
range(start, stop)   # generování číselné posloupnosti
abs(cislo)           # absolutní hodnota
round(cislo, des_mista) # zaokrouhlení
max(a, b, c...)     # maximum
min(a, b, c...)     # minimum
sum(seznam)          # součet prvků
```

Konverze mezi datovými typy

Základní typy

```
# INT - převod na celé číslo
int("123")           # "123" → 123 ✓
int(45.9)            # 45.9 → 45 (ořízne desetinnou část)
int("12.5")          # CHYBA! Nelze převést string s desetinnou čárkou
int("ahoj")          # CHYBA! Nelze převést text na číslo

# FLOAT - převod na desetinné číslo
float("12.5")        # "12.5" → 12.5 ✓
float(10)             # 10 → 10.0 ✓
float("ahoj")         # CHYBA! Nelze převést text na číslo

# STR - převod na řetězec (funguje téměř vždy)
str(123)              # 123 → "123" ✓
str(45.9)             # 45.9 → "45.9" ✓
str(True)              # True → "True" ✓
str([1, 2, 3])         # [1, 2, 3] → "[1, 2, 3]" ✓

# BOOL - převod na boolean (True/False)
bool(1)                # 1 → True ✓
bool(0)                # 0 → False ✓
bool("text")            # "text" → True ✓
bool("")                # "" (prázdný string) → False ✓
bool("False")           # "False" → True (!! neprázdný string je vždy True
bool([])                # [] (prázdný seznam) → False ✓
bool([1, 2])             # [1, 2] → True ✓
```

Složitější typy

```
list("ahoj")      # převod stringu → ['a', 'h', 'o', 'j']
list((1, 2, 3))   # převod tuple → [1, 2, 3]

tuple("ahoj")     # převod stringu → ('a', 'h', 'o', 'j')
tuple([1, 2, 3])  # převod listu → (1, 2, 3)

set([1, 2, 2, 3]) # převod listu → {1, 2, 3} (odstraní duplicitu)
set("ahoj")       # převod stringu → {'a', 'h', 'o', 'j'}

dict()            # vytvoření prázdného slovníku
```

Číselné operace

```
+  # sčítání
-  # odčítání
*  # násobení
/  # dělení (výsledek vždy float)
// # celočíselné dělení
%  # modulo (zbytek po dělení)
** # mocnina
```

Operace s řetězci (string)

```
retezec.upper()      # převod na velká písmena
retezec.lower()      # převod na malá písmena
retezec.strip()       # odstranění mezer na začátku/konci
retezec.split(sep)    # rozdělení na seznam
retezec.replace(old, new) # nahrazení části řetězce
retezec.find(podretezec) # najde pozici podřetězce (-1 pokud nenajde)
retezec.count(znak)   # počet výskytů znaku
retezec.startswith(text) # začíná řetězec daným textem?
retezec.endswith(text) # končí řetězec daným textem?
retezec.isdigit()     # jsou všechny znaky číslice?
retezec + retezec2    # spojení řetězců
retezec * n           # opakování řetězce n-krát
podretezec in retezec # je podřetězec obsažen v řetězci?
```

Seznam (list)

```
seznam = [1, 2, 3]
seznam.append(prvek)      # přidá prvek na konec
seznam.insert(index, prvek) # vloží prvek na danou pozici
seznam.remove(prvek)       # odstraní první výskyt prvku
seznam.pop()               # odstraní a vrátí poslední prvek
seznam.pop(index)          # odstraní a vrátí prvek na pozici
seznam.clear()              # vymaže všechny prvky
del seznam[index]          # smaže prvek na pozici
seznam[index]                # přístup k prvku
prvek in seznam             # je prvek obsažen v seznamu?
```

Tuple (n-tice)

```
ntice = (1, 2, 3)
ntice[index]           # přístup k prvku
prvek in ntice        # je prvek obsažen v n-tici?
# Tuple nelze měnit po vytvoření!
```

Množina (set)

```
mnozina = {1, 2, 3}
mnozina.add(prvek)      # přidá prvek
mnozina.remove(prvek)   # odstraní prvek (vyhodí chybu pokud neexistuje)
mnozina.discard(prvek) # odstraní prvek (bez chyby pokud neexistuje)
prvek in mnozina       # je prvek obsažen v množině?
```

Slovník (dictionary)

```
slovnik = {"klic": "hodnota"}
slovnik[klic]           # přístup k hodnotě
slovnik[klic] = hodnota # přidání/změna hodnoty
del slovnik[klic]       # smazání klíče
klic in slovnik         # je klíč obsažen ve slovníku?
slovnik.keys()          # všechny klíče
slovnik.values()         # všechny hodnoty
slovnik.items()          # páry klíč-hodnota
```

Podmínky (if)

```
if podmínka:
    # kód pokud je podmínka True
elif další_podmínka:
    # kód pokud je další_podmínka True
else:
    # kód pokud žádná podmínka není True
```

Porovnávací operátory: == , != , < , > , <= , >=

Logické operátory: and , or , not

Cyklus for

```
# Iterace přes seznam/tuple/řetězec
for prvek in seznam:
    # kód pro každý prvek

# Iterace s indexem
for i in range(10):
    # kód pro i = 0, 1, 2, ..., 9

# Range s krokem
for i in range(0, 10, 2):
    # kód pro i = 0, 2, 4, 6, 8
```

Cyklus while

```
while podmínka:
    # kód dokud je podmínka True
```

Tip: Nezapomeň na odsazení! používá odsazení (4 mezery nebo Tab) pro bloky kódu.