

# Tahák základů Pythonu

## Hlavní datové typy

**boolean** = True / False ... pravda/nepravda  
**integer** = 10 ... celé číslo  
**float** = 10.01 ... desetinné číslo  
**string** = "123abc" ... řetězec  
**list** = [ value1, value2, ... ] ... seznam  
**dictionary** = { key1:value1, key2:value2, ... } ... slovník

## Číselné operátory

+ sčítání  
- odčítání  
\* násobení  
/ dělení  
\*\* mocnina  
% zbytek po dělení  
// celočíselné dělení

## Porovnávací operátory

== rovno  
!= nerovno  
> větší  
< menší  
>= větší nebo rovno  
<= menší nebo rovno

## Operace s řetězci

**string[i]** - získá znak na pozici i  
**string[-1]** - získá poslední znak  
**string[i:j]** - získá znaky v rozsahu od i do j

## Metody pro řetězce

**string.count(x)** - spočítá, kolikrát se x objeví  
**string.find(x)** - pozice prvního výskytu x  
**string.format(x)** - vrátí řetězec, který obsahuje hodnotu x  
**string.join(L)** - vrátí řetězec s hodnotami L spojenými řetězcem  
**string.lower()** - převede na malá písmena  
**string.replace(x,y)** - nahradí znak x za y  
**string.split(x)** - vrátí seznam hodnot oddělených x  
**string.strip(x)** - odstraní bílé znaky na začátku a na konci  
**string.upper()** - převede na velká písmena

## Operace se seznamy

**list = []** - definuje prázdný seznam  
**list[i] = x** - uloží x s indexem i  
**list[i]** - získá prvek s indexem i  
**list[-1]** - získá poslední prvek  
**list[i:j]** - získá prvky v rozsahu od i do j  
**del list[i]** - odstraní prvek s indexem i

## Operace se slovníky

**dict = {}** - definuje prázdný slovník  
**dict[k] = x** - uloží hodnotu x spojenou s klíčem k  
**dict[k]** - získá hodnotu prvku s klíčem k  
**del dict[k]** - odstraní prvek s klíčem k

## Metody pro seznamy

**list.append(x)** - přidá x na konec seznamu  
**list.clear()** - odstraní všechny prvky ze seznamu  
**list.copy()** - vrátí kopii seznamu  
**list.count(x)** - spočítá, kolikrát se x objevuje v seznamu  
**list.extend(L)** - přidá na konec seznamu seznam L  
**list.index(x)** - vrátí index prvního výskytu x  
**list.insert(i,x)** - vloží x na pozici i  
**list.pop(i)** - odstraní prvek na pozici i a vrátí jeho hodnotu  
**list.remove(x)** - odstraní první prvek v seznamu, jehož hodnota je x  
**list.reverse()** - obrátí pořadí prvků v seznamu  
**list.sort()** - seřadí prvky seznamu

## Metody pro slovníky

**dict.clear()** - odstraní všechny klíče a hodnoty ze slovníku  
**dict.copy()** - vrátí kopii slovníku  
**dict.get(k)** - vrátí hodnotu spojenou s klíčem k  
**dict.items()** - vrátí seznam dvojic (klíč, hodnota)  
**dict.keys()** - vrátí seznam klíčů  
**dict.pop(k)** - odstraní prvek spojený s klíčem a vrátí jeho hodnotu  
**dict.update(D)** - přidá klíče a hodnoty, nový slovník (D) do slovníku  
**dict.values()** - vrátí seznam hodnot

**Legenda:** x,y reprezentují jakoukoli hodnotu dat, s značí řetězec, n číslo, L seznam, kde i,j jsou indexy seznamu, D značí slovník a k je klíč slovníku.

## Ošetření výjimek

**try:**  
    <kód> # Kód, který může vyvolat výjimku  
**except <chyba>:**  
    <kód> # Kód pro ošetření konkrétní výjimky  
**else:**  
    <kód> # Kód pokud nebyla vyvolána žádná výjimka

## Vestavěné funkce

**abs(n)** - vrátí absolutní hodnotu n  
**float(x)** - převede x na desetinné číslo  
**help(object)** - vytiskne nápovědu o objektu  
**input(s)** - vytiskne s a čeká na vstup, který bude vrácen  
**int(x)** - převede x na celé číslo  
**len(x)** - vrátí délku x (s, L nebo D)  
**list(x)** - převede x na seznam  
**map(function, L)** - aplikuje funkci na hodnoty v L  
**max(L)** - vrátí maximální hodnotu v L  
**min(L)** - vrátí minimální hodnotu v L  
**print(x, sep='y')** - vytiskne objekty x oddělené y  
**range(n1,n2,n)** - vrátí sekvenci čísel od n1 do n2 s krokem n  
**round(n1,n)** - zaokrouhlí číslo n1 na n desetinných míst  
**sorted(L)** - vrátí seřazený seznam  
**str(x)** - převede x na řetězec  
**sum(L)** - vrátí součet hodnot v L  
**type(x)** - vrátí typ x (řetězec, desetinné číslo, seznam, slovník ...)

## Třídy

**class Třída:**

**def \_\_init\_\_(self, <parametry>):** # Konstruktor třídy  
self.atribut = hodnota

**def metoda(self, <parametry>):** # Metoda třídy  
    <kód> # Kód metody  
    return <data>

**objekt = Třída(<parametry>)** # Vytvoření objektu třídy

**výsledek = objekt.metoda(<parametry>)** # Volání metody objektu

**hodnota\_atributu = objekt.atribut** # Přístup k atributu objektu

**class Potomek(Třída):** # Dědění  
    **def metoda(self, <parametry>):** # Přetížená metoda rodičovské třídy  
        <kód> # Kód metody  
        return <data>

## Import modulů

**import module**                      **from module import \***  
**module.function()**                **function()**

## Podmínkové příkazy

**if <podmínka>:**

    <kód> # Provede se, pokud je podmínka splněna

**elif <podmínka>:**

    <kód> # Provede se, pokud je druhá podmínka splněna a první ne

...

**else:**

    <kód> # Provede se, pokud žádná z předchozích podmínek nebyla splněna

**if <hodnota> in <seznam>:**

    <kód> # Provede se, pokud je hodnota v seznamu

## “While” cyklus

**while <podmínka>:**

    <kód> # Kód, který se bude opakovat, dokud je podmínka splněna

## “For” cyklus

**for <proměnná> in <seznam>:**

    <kód> # Kód, který se bude opakovat pro každou hodnotu v seznamu

**for <proměnná> in range(začátek, konec, krok):**

    <kód> # Kód, který se bude opakovat pro každou hodnotu v rozmezí

**for index, hodnota in enumerate(seznam):**

    <kód> # Kód, který se bude opakovat pro každou hodnotu v seznamu s jejím indexem

**for klíč, hodnota in slovník.items():**

    <kód> # Kód, který se bude opakovat pro každý pár klíč-hodnota ve slovníku

## Ovládací příkazy smyčky

**break** - ukončí provádění smyčky

**continue** - přeskočí na další iteraci

**pass** - nedělá nic

## Funkce

**def funkce(<parametry>):**

    <kód> # Tělo funkce

    return <data> # Vrací určitá data z funkce

Vaše cesta k programování  
začíná zde!  
[www.umimpython.cz](http://www.umimpython.cz)

