Programko.sk

Python 2

1. Polia

Základným meniteľným typom v Pythone je **pole**. V porovnaní s poľami v iných programovacích jazykoch, pythonovské pole je v skutočnosti **dynamické pole**, keďže mu môžeme meniť jeho veľkosť (môžeme ho zväčšovať alebo zmenšovať. Niekedy budeme pole zobrazovať ako tabuľku hodnôt, pre ktorú je v každom políčku tabuľky jeden prvok.

VYTVOR POLE

1.1 Príklad

Vytvorte pole obsahujúce čísla 1,3 a 8

Použijeme: pole vytvárame pomocou hranatých zátvoriek “[ ]”

APPEND()

1.2 Príklad

Do vami vytvoreného poľa z úlohy 1.1 pridajte ďalšie (ľubovoľné) číslo

Použijeme príkaz:
append()

Pole môže obsahovať aj prvky rôzneho druhu:

Správne využitie polí:

[1,2,3,100,200]
[“car”, “bus”]
[True, “car”, “bus”]
[1, 1.5, “bus”]

Na pripomenutie

Integer – celočíselné číslo

Float – desatinné číslo

String – reťazec – slovo, veta

Bool – logická hodnota, True alebo False

Základné operácie v pythone pre prácu s poliami sú:

- +zreťazenie [1,2] + [0] = [1,2,0]
- in 6 in [1,3,7], sa rovná false, pretože prvok neobsahuje číslo 6
- len() vráti dĺžku poľa
- sum() vráti súčet prvkov v poli
- min() vráti minimálny prvok v poli
- max() vráti maximálny prvok v poli

SUM()

2.1 Príklad

Vypočítajte aký je súčet prvkov v poli [12,-4,6]

Použijeme funkciu:

sum()

2.2 Príklad

Vypočítajte aký je súčet prvkov v poli [12,-4,6] bez použitia funkcie sum()

Použijeme:

V poli vieme k prvkom pristupovať pomocou indexov(pozícií). Polia indexujeme od nuly. Pracujme s poľom:

array[0] = [5,3,6]

array[0] = 5

array[2] = 6

LEN()

2.3 Príklad

Vytvorte si 2 ľubovoľné polia. Vypíšte, ktoré z nich je väčšie(obsahuje viac prvkov).

Použijeme funkciu:

len()

2.4 Príklad

Pomocou forcyklu vytvorte pole obsahujúce čísla od 1 do 100 vrátane.

2.5 Príklad

Z poľa, ktoré ste si vytvorili v príklade 1.6 vypíšte všetky párne čísla.

Nápoveda:

Budeme potrebovať: %

3. Pokračujeme s poliami

3.1 Príklad

Z poľa [1,3,7,10,2,0,49,81,25] vypíšte všetky čísla deliteľné tromi.

3.2 Príklad

Napíšte koľko čísel väčších a koľko čísel menších ako 15 sa nachádza v poli [1,3,7,10,2,0,49,81,25,100]

Nápoveda:

Potrebujeme 2 pomocné premenné, kde si budeme ukladať aktuálny počet väčších a menších ako 15.

3.3 Príklad

Vypíšte prvok, ktorý sa nachádza v strede poľa [1,3,7,10,2,0,49,81,25,100]

Nápoveda:

Použijeme funkciu **len()**

4. Obvod a obsah geometrických útvarov

Na dnešnej hodine sa budeme venovať týmto geometrickým útvarom:

Štvorec

o = 4 * a
S = a* a

Obdĺžnik

o = 2 *a + 2*b
S = a* b

4.1 Príklad

Vytvoríme funkciu na výpočet obsahu a obvodu štvorca, ktorá bude mať stranu A ako svoj parameter.

4.2 Príklad

Vypočítajte obvod a obsah obdĺžnika. Veľkosť strany A a B nájdete v poli [5,7]

Nápoveda:

Strana A = prvý prvok v poli

Strana B = druhý prvok v poli

4.3 Príklad

Od užívateľa si vypýtate názov jedného športu, ktorému sa venuje. Názov uložíte do vami vytvoreného poľa.

4.4 Príklad

Od užívateľa si vypýtate názov všetkých športov, ktorým sa venuje. Názvy uložíte do vami vytvoreného poľa.

String ako pole

So stringom vieme pracovať rovnako ako s poľom.

4.5 Príklad

Vypíšte prvé a posledné písmeno slova „Python“

5. Príklady

5.1 Príklad

Funkcia vezme ako parameter pole čísel, následne vytvorí a vypíše REVERZ tohto poľa.

Príklad:
[1,2,3] = [3,2,1]

5.2 Príklad

Odstráňte posledný prvok z poľa [0,100.4,6,7]

Poznámka:
Použijeme príkaz **POP()**

5.3 Príklad

Vytvorte si ľubovoľné pole čísel. Funkcia is_duplicate() vráti True ak pole obsahuje duplikát, inak False.

5.4 Príklad

Funkcia príjma ako parameter pole celých čísel. Funkcia upraví dané pole tak, aby bol každý prvok v poli zmenšený

Opakovanie

INPUT() + IF CLAUSE () + ARRAY

Od užívateľa vyžiadame 10 slov, slová, ktorých dĺžka je väčšia ako 5 vložíme do nami vytvoreného poľa.

ÚLOHA PRE SPOLUŽIAKA

Pre spolužiaka vymyslíte úlohu. Úloha bude zameraná na podmienky, forcykly a polia.

6. Cyklus while

Okrem cyklu **for** máme ešte druhý typ cyklu reprezentovaný príkazom **while**. Na rozdiel od for, kde vopred poznáme počet opakovaní, sa while používa, ak cyklus závisí na nejakej podmienke.
Telo cyklu sa opakuje, kým je podmienka splnená.

Vyskúšajte

cyklus sa bude opakovať kým premenná odpoved nebude obsahovať “stop”.

```
1
2   odpoved = input('Povedz stop ')
3
4   while odpoved != 'stop':
5       print('Špatně, zkus to znovu')
6       odpoved = input('Povedz stop! ')
7
```

Cyklus môžeme umelo zastaviť pomocou príkazu “**break**”.
Príkaz break z cyklu “vyskočí” a začnú sa vykonávať príkazy za cyklom.

PRÍKLADY

6.1 Príklad - Myslím si číslo.

Program si od užívateľa bude pýtať číslo dovtedy, kým nezadá číslo 7.

6.2 Príklad

Funkcia dostane ako parameter celé číslo.

Pomocou **while** cyklu číslo znižujte a vypisujte dovtedy, kým sa nerovná 0.

6.3 Príklad

Pomocou while cyklu vypíšte prvky poľa **[“Bratislava”, “New York”, “Paris”, “Moscow”]**.

6.4 Príklad

Funkcia dostane ako parameter celé číslo, nazveme ho number.

Pomocou **while** cyklu vypočítajte aký je výsledok **number // 2**.

6.5 Príklad - Myslím si číslo - updated

Zvoľte si číslo a uložte ho do premennej.

Užívateľ má 5 pokusov na to, aby uhádol vami zvolené číslo. Ak ho uhádne program vypíše **“You won!”**, v opačnom prípade vypíše **“You failed!”**.

NÁHODNÉ ČÍSLA

Pre prácu s náhodnými číslami budeme potrebovať importovať knižnicu random.
Knižnicu importujeme nasledujúcim príkazom.

import random

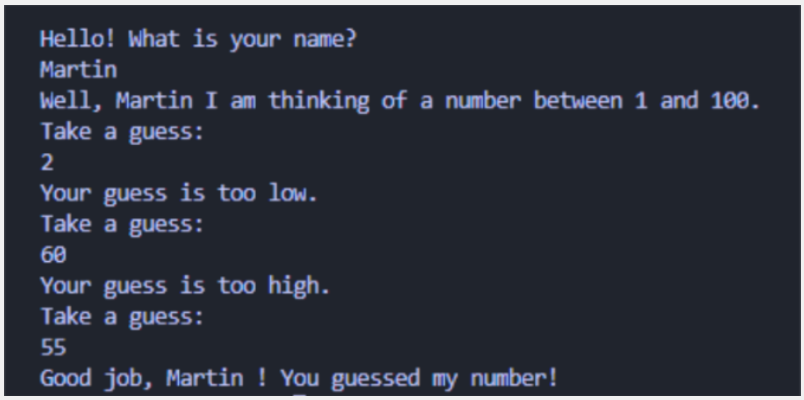
Nasledujúci príkaz nám vráti náhodné číslo, ktoré sa nachádza v intervale 1 <= x <= 9

random.randint (1,9)

6.6 Príklad - Myslím si číslo - final

Do premennej x si uložte náhodné číslo, od 1 do 100.
Vašou úlohou je to číslo uhádnuť.

Hra prebieha následovne:



7. Príklady

7.1 Príklad - Hádzanie kockou

Vytvorte funkciu, ktorá bude simulovať hod kockou. Kocka hádžeme náhodne 20 krát. Vypíšte koľko krát sa nám podarilo vypísať čísla 1,2,3,4,5 a 6.

Pre výber náhodného prvku z poľa môžeme použiť funkciu **random.choice()**

Napríklad:

```
array= [1,2,0,100,4]
random_number = random.choice (array)
```

7.2 Príklad

Uvažujme pole obsahujúce **8** rôznych farieb. Vyberte náhodnu farbz a vypíšte ju.

7.3 Príklad

Vypíšte náhodné číslo, ktoré je deliteľné **7** a **5** zároveň.

7.4 Príklad

Vytvorte program, ktorý vygeneruje pole obsahujúce **5** náhodných párnych čísel.

Funkcia **random.shuffle()** náhodne zamieša prvky v poli.

Napríklad:

```
array = [1,2,3]
random.shuffle(array)
print (array)
```

jeden z možných výstupov: **[1,3,2]**

7.5 Príklad - Kameň, Papier, Nožnice

Vytvorte simuláciu hry kameň papier nožnice. Hra končí v momente keď jeden z hráčov dosiahne 3 body.

7.6 Príklad

Vytvorte funkciu, ktorá bude generovať náhodné čísla od **1** do **100**, kým súčet týchto čísel nebude rovný **1000**.

7.7 Príklad

Vytvorte program, ktorý vypočíta faktoriál náhodného čísla.

Napríklad:
3! = 3 * 2 * 1 = 6

7.8 Príklad

Funkcia berie ako parameter náhodné číslo x - Funkcia vypíše výsledok **1000 // x**.

8. Príklady

8.1 Príklad

Program si vypýta od užívateľa názov všetkých postáv z filmu Harry Potter, ktoré pozná.
Program následne vypíše počet odpovedí užívateľa

8.2 Príklad

Program spočíta počet písmen v slove **PROGRAMKO**.
Následne písmeno „**P**“ nahradí písmenom „**R**”. Nakoniec k slovu pridá „**je mega**“.

8.3 Príklad

Máme dané pole **[10,20,33,44,45]**. Pole vypíše z poľa všetky čísla, ktoré sú deliteľné **3** ALEBO **5**.

8.4 Príklad

Užívateľ zadá svoj rok narodenia. Program vypočíta, koľko rokov bude mať v roku 2155.

8.5 Príklad

Vypočítajte **10!**.
Použite forcyklus alebo while-cyklus.
10! = 10 * 9 * 8 * 7 * 6 * 5* 4 * 3 * 2 * 1

8.6 Príklad

Vytvorte jednoduchú kalkulačku. Kalkulačka bude obsahovať funkcie pre +, -, *, //, !, %. Na vrátenie hodnoty z funkcie použite funkciu **RETURN**.

8.7 Príklad

Máme dané 2 polia.
Pole1 = [10,20,30,45,100]
Pole2 = [49, 104, 1004,55, 26]

Výsledkom programu bude pole, ktoré je súčtom **Pole1** a **Pole2**.
Naším výsledným poľom bude **[59,134,1034,100, 126]**

8.8 Príklad

Vygenerujte 3 celé čísla x, pre ktoré platí **x >= 100** and **x <= 1000**, a zároveň sú čísla deliteľné 7.
Čísla uložte do poľa. Pole vypíšte.

9. Príklady

9.1 Príklad

Vytvorte ľubovoľné pole. Z poľa odstráňte všetky duplikáty.

9.2 Príklad

Program vypočíta koľko písmen “**a**” obsahuje slovo **ANAKONDA**.

9.3 Príklad

Máme číslo x. Určite či je číslo x prvočíslom.

Prvočíslo je číslo, ktoré je deliteľné iba jednotkou alebo sebou samým.

9.4 Príklad

Sú Vianoce ?
Funckia dostane ako parametre deň, mesiac a rok. Program určí či ide o vianočný dátum.

9.5 Príklad

Vytvorte ľubovoľné pole.
Funckia dostane ako parameter celé číslo. Následne funkcia vráti n-té najväčšie číslo v tomto poli.

Nepovinné príklady:

9.6 Príklad

Vypíšte čísla od **1** do **99** vrátane.
Vypíšte slovo “**hello**” 18-krát.

9.7 Príklad

Napíšte program, ktorý si od užívateľa vypýta slovo, uloží ho do premennej, Ak je slovo „voldemort“, vypíše „psst, toto meno nesmieme vyslovovať“.

9.8 Príklad

Napíšte funkciu **poradie(n):**, ktorej pošlete jeden parameter (**integer**). Táto funkcia potom vypíše
Všetky čísla od **1** po **n**, ktoré jej bolo zadané.

9.10 Príklad

Napíšte funkciu **obsahuje_znak(slovo, znak):** , ktorá prejde zadané slovo forcyklom, a ak narazí na zadaný znak, vráti **true**, inak po skončení forcyklu vráti **false**.

9.11 Príklad

Zdvojenie písmen
Napíšte funkciu, ktorá medzi každé 2 písmená daného text vloží dodaný text.
Pre vstup: Python program vráti PPyytthhoonn

9.12 Príklad

Cenzura
Napíšte funkciu, ktorá zcenzuruje string tak, že každý druhý znak nahradí za X.
Pre vstup: Python program vráti Pxtxox

9.13 Príklad

Počet deliteľov
Napíšte funkciu divisors_count(n), ktorá vráti počet deliteľov čísla **n**.

9.14 Príklad

Počet deliteľov
Napíšte funkciu **divisors_count(n)**, ktorá vráti počet deliteľov čísla **n**.

9.15 Príklad

Rozdiel v počte A
Napíšte funkciu príjmajúcu **2** stringy, ktorá vypíše informáciu o tom, ktorý zo stringov obsahuje menej znakov ‘ **a/A** ‘. Následne funkcia vypíše rozdiel v počte znakov ‘ **a/A** ‘ medzi týmito dvoma stringami.

10.1 Príklad

Vytvorte program, ktorému užívateľ zadá reťazec a program vypíše jeho prvé štyri znaky.

10.2 Príklad

Užívateľ zadá slovo. Pomocou for cyklu potom skontrolujte či sú všetky písmená v slove rovnaké a vypíšte či áno alebo nie.

10.3 Príklad

Vytvorte funkciu **sucetAleboSucin**, ktorá dostane dve čísla, tie vynásobí a ak je súčin menší ako **1000**, vráti ho, inak vráti namiesto neho ich súčet.

10.4 Príklad

Vytvorte funkciu **parnePismena**, ktorá dostane jedno slovo, a vypíše iba písmená na párnych pozíciách.

Príklad: slovo = “**Ahoj**“
Výpis: “**Ao**“

10.5 Príklad

Napíšte program, ktorý si od užívateľa pýta **2** čísla, až kým prvé číslo nie väčšie ako prvé

10.6 Príklad

Napíšte program, ktorý si od užívateľa pýta číslo, ktoré následne vo while cykle delíte 10, kým je číslo väčšie ako 0.

11. Slovníky

Slovník, asociatívny zoznam (alebo mapa) mapuje kľúče na hodnoty. Je podobný zoznamu, ale jeho prvky nie sú indexované pomocou postupnosti celých čísel, ale pomocou kľúčov. Kľúčom môže byť napríklad číslo, reťazec alebo iný nemeniteľný objekt.

Napríklad zoznam kľúčom byť nemôže. Pod týmto kľúčom môžme pristupovať k uloženým hodnotám, ak sú v slovníku prítomné.

Príklad slovníku:

```
points = {"Jack":66, "Peter":0, "Denis":0}
```

Prístup k hodnote pomocou kľúča:

```
print(points["Jack"]) # 66
```

Pridanie nového prvku:

```
points["Tom"] = 60 # pridanie novej hodnoty do slovníka
print(points) # {"Jack":66, "Peter":0, "Denis":0, "Tom":60}
```

Prechádzanie kľúčov v slovníku:

```
for name in points:
    print(name)
```

Prechádzanie hodnôt v slovníku:

```
for value in points.values():
print(value)
```

Príklady:

11.1 Príklad

Majme slovník: **{“mesto”:”Bratislava”, “dedina”: Vlkolínec}**. Pridajte do slovníku akúkoľvek hodnotu s kľúčom štát.

11.2 Príklad

Majme slovník: **{“mesto”:”Bratislava”, “dedina”: Vlkolínec}**. Funkcia dostane ako parameter názov slovenského mesta. Funkcia vráti **True** ak slovník obsahuje zadané mesto, **False** inak.

11.3 Príklad

Vytvorte funkciu, ktorá sčíta počet všetkých obyvateľ miest v slovníku:
Obyvatelia = {Bratislava:500 000, Pezinok:25 000, Prešov: 100 000, Humenné: 40 000}

11.4 Príklad

Ktoré mesto obsahuje najväčší počet obyvateľov?
Môžte použiť slovník:
Obyvatelia = {Bratislava:500 000, Pezinok:25 000, Prešov: 100 000, Humenné: 40 000}

11.5 Príklad

Od uzivatela si vypýtajte **20x** ľubovoľné slovo.
Slovník bude obsahovať slovo zadané užívateľom ako kľúč. Hodnotou bude, koľkokrát užívateľ dané slovo zadal

11.6 Príklad

Pomocou nasledujúci polí vytvorte slovník:
keys = [1,2,3,4]
values = [„one“, „two“, „three“, „four“]

11.7 Príklad

Funckia dostane ako parameter string. Do slovník uložte aké písmená string obsahuje a aký je ich počet:

Napríklad:
slovo = “adam”
pismena = {“a”: 2, “d”: 1, “m”: 1}