prog_datasci_2_python_exercicisResolts

September 25, 2020

1 Fonaments de Programació

1.1 Unitat 2: Breu introducció a la programació en Python 3 - Exercicis resolts

En aquest Notebook hi trobareu un conjunt d'exercicis per practicar. Aquests exercicis no puntuen per a la PAC, però us recomanem que els intenteu resoldre com a part del procés d'aprenentatge. Trobareu exemples de possibles solucions als exercicis al propi notebook, però és important que intenteu resoldre'ls vosaltres abans de consultar les solucions. Les solucions us han de permetre validar les vostres respostes, així com veure alternatives de resolució de les activitats. També us animem a preguntar qualsevol dubte que us sorgeixi sobre la resolució de les activitats per practicar al fòrum de l'aula.

1.2 Preguntes i exercicis per a practicar

1.2.1 Pregunta 1

Quin és el valor final d'a, b i c (atenció als tipus de nombre decimal i enter)?

a = 5 * 11

b = a / 2.

c, a = b * 2, b + 1

Resposta

1.2.2 Pregunta 2

Identifiqueu tres tipus de dades de Python i expliqueu breument en què es diferencien. Expliqueu la diferència entre el tipus llista i el tipus tupla.

Resposta

1.2.3 Pregunta 3

Quin és el resultat de l'operació lógica `'poma'>'casa'`? Explica per què.

Resposta

1.2.4 Exercici 1

Escriviu un programa que seleccioni la cadena "analista" a partir de la cadena: "Ana és bona analista de dades.". Mostra-la per pantalla.

[1]:
```
# Resposta
```

1.2.5 Exercici 2

Quina expressió en Python necessitem per aconseguir l'*string* "nohtyPythonython" utilitzant només la paraula "Python"? **(1 punt)**

[2]:
```
# Resposta
```

1.2.6 Exercici 3

Escriviu un programa que assigni dos valors enters qualsevol (trieu vosaltres un nombre enter aleatori) a dues variables amb nom `a` i `b`. Utilitzeu les variables definides anteriorment per evaluar la següent expressió matemàtica:

(a2 + b2)2

[3]:
```
# Resposta
```

1.2.7 Exercici 4

Escriviu un programa que calculi el volum d'una piràmide quadrangular amb una base de 4x5 metres de longitud i amplada respectivament, i amb una altura de 7,5 m. Recorda que la fórmula per calcular el volum és

on l i w són la longitud i l'amplada de la base, i h és l'altura.

[4]:
```
# Resposta
```

1.2.8 Exercici 5

Escriviu un programa que defineixi una llista amb el nom de tots els mesos de l'any. Feu que mostri els mesos que corresponen a la tardor.

[5]:
```
# Resposta
```

1.2.9 Exercici 6

Escriviu un programa que a partir de la llista de nombres donada mostri per pantalla una cadena de caràcters amb tots els números, separats entre ells per un guió, i duplicant el primer i l'últim element de la llista.

Així, per exemple, per a la llista ['1', '2', '3'], el programa hauria de mostrar la cadena: '1-1-2-3-3'.

[6]:
```
# Resposta
```

1.2.10 Exercici 7

Ordena la següent llista de cadenes de caràcters:

1. Invertint l'ordre original
2. En ordre alfabètic invers

Nota: Pots consultar la documentació oficial de la funció `sorted()` per veure quins paràmetres pots fer servir per resoldre la segona part de l'activitat.

[7]:
```
st_chars = ["Benjamin Sisko", "Kira Nerys", "Odo", "Quark", "Jadzia Dax"]
```

[8]:
```
# Resposta
```

1.2.11 Exercici 8

A partir de la següent llista,

`A_list = [42, 7.5," Answer to the Ultimate Question "," Dave ", 7.5]`

proporciona expressions que retornin:

1. El nombre de vegades que apareix l'element `7.5` a la llista.
2. La posició de la primera aparició de la valor `7.5`
3. La mateixa llista sense l'últim element

Nota: Al notebook de teoria hem vist què són les llistes i algunes operacions sobre elles. Per fer l'activitat, necessitareu investigar algunes operacions addicionals que podem realitzar sobre llistes. Per a això podeu consultar la documentació oficial de Python sobre llistes (intro i més sobre llistes).

[9]:
```
# Resposta
```

1.2.12 Exercici 9

Quina expressió en Python necessitem per aconseguir la *string* `Learning Python` utilitzant només les variables `str1` i `str2` definides?

[10]:
```
str1 = "I love Python, it's great!"
str2 = "Learning"

# Resposta
```

1.3 Solucions als exercicis per a practicar

1.3.1 Pregunta 1

Quin és el valor final d'`a`, `b` i `c` (atenció als tipus de nombre decimal i enter)? **(1 punt)**

a = 5 * 11

b = a / 2.

c, a = b * 2, b + 1

Anirem per passos:

A la primera línia trobem que la variable `a` contindrà el valor de 5 * 11. (el primer un nombre enter i el segon també un nombre enter, pel que el resultat serà enter), **55**. La variable `b`, al contrari, tindrà el valor de 55 (enter) dividit per 2. (decimal), per tant, **27.5** (decimal).

A la tercera línia tenim una expressió on s'assignen dues variables alhora (en realitat això és una assignació entre tuples atès que ',' és el constructor de tuples en Python). Primer avaluem la part de la dreta:

- `b \* 2`, el resultat és decimal donat que també ho era abans: 27.5 * 2 = 55.0
- `b + 1`, el resultat és també decimal, 27.5 + 1 = 28.5

Ara assignem els resultats a la part de l'esquerra de l'expressió:

- `c = b \* 2` (l'expressió que ja s'havia avaluat) = 55.0
- `a = b + 1` = 28.5

Per tant, tenim:

a = 28.5, b = 27.5, c = 55.0

Executem el codi en Python per veure que efectivament aquest és el resultat:

[11]:
```
a = 5 * 11
b = a / 2.
c, a = b * 2, b + 1

print (a, b, c)
```

```
28.5 27.5 55.0
```

1.3.2 Pregunta 2

Identifiqueu tres tipus de dades de Python i expliqueu breument en què es diferencien. Expliqueu la diferència entre el tipus llista i el tipus tupla. **(1 punt)** NM

Resposta

Integers: Fa referència a valors numèrics enters. A Python 3, no hi ha cap límit en quant a la mida d'un valor enter. Per descomptat, està limitat per la quantitat de memòria que té el sistema.

***Floats* (*Floating-Point Numbers*)**: A diferència del tipus *integer* o enter, el tipus *float* de Python designa valors flotants o decimals que s'especifiquen amb un punt.

Strings: Les *strings* (o cadenes) són seqüències de dades de caràcters. El tipus cadena a Python s'anomena `str`. Els literals de cadena es poden delimitar mitjançant cometes simples o dobles. Una cadena a Python pot contenir tants caràcters com es desitgen. L'únic límit són els recursos de memòria de la màquina.

A Python, les **llistes** i les **tuples** es declaren de diferents maneres. Es crea una llista mitjançant claudàtors `[ ]`, mentre que la tupla es declara amb parèntesi `( )`. Hi ha diferències notables entre els dos, essent la principal diferència que les llistes són mutables mentre que els tuples són immutables. Una llista té una mida variable mentre que una tupla té una mida fixa. A més, les operacions sobre tuples es poden executar més ràpidament en comparació amb les de les llistes.

1.3.3 Pregunta 3

Quin és el resultat de l'operació lógica `'poma'>'casa'`? Explica per què.

Resposta

El resultat de l'operació es *True*. La comparació de cadenes Python es realitza mitjançant l'evaluació dels caràcters de les dues cadenes, un per un. Quan es troben diferents caràcters, es compara el seu valor Unicode. El caràcter amb un valor Unicode inferior es considera més petit. En aquest cas "c" és menor si es compara amb "p" a causa dels seus valors Unicode corresponents. A continuació utilitzem la funció ord() per mostrar el valor del codi Unicode de cada un dels caràcters:

[12]:
```
print('poma' > 'casa')
print('El valor unicode de p és', ord('p'), 'i el de c és ', ord('c'))
```

```
True
El valor unicode de p és 112 i el de c és  99
```

1.3.4 Exercici 1

Escriviu un programa que seleccioni la cadena "analista" a partir de la cadena: "Ana és bona analista de dades.". Mostra-la per pantalla.

[13]:
```
# Resposta
frase = "Ana és bona analista de dades."

paraula = frase[12:20]

print(paraula)
```

```
analista
```

1.3.5 Exercici 2

Quina expressió en Python necessitem per aconseguir l'*string* "nohtyPythonython" utilitzant només la paraula "Python"?

Com podem veure, per construir la cadena de text necessitem dos blocs diferenciats:

- la sub-cadena *"nothyP"* que correspon a la paraula Python en ordre invers
- la sub-cadena *"ython"* repetida 2 vegades

Podem crear aquests blocs de forma separada i al final concatenar-los per obtenir el resultat final.

[14]:
```
# Resposta

# Primer definim la nostra cadena base
python = "Python"

# Accedim als caràcters de python en ordre invers
bloc_on = python[::-1]

# Accedim als 5 caràcters finals i els repetim 2 vegades
bloc_Py = 2* python[1:]

# Finalment, concatenem els blocs anteriors
resultat = bloc_on + bloc_Py
print (resultat)
```

```
nohtyPythonython
```

1.3.6 Exercici 3

Escriviu un programa que assigni dos valors enters qualsevol (trieu vosaltres un nombre enter aleatori) a dues variables amb nom `a` i `b`. Utilitzeu les variables definides anteriorment per evaluar la següent expressió matemàtica:

(a2 + b2) 2

[15]:
```
# Escollim dos nombres enters a l'atzar
a = 6
```

```
b = 5

# Calculem els quadrats de les variables anteriors utilitzant l'operador **
quadrat_a = a ** 2
quadrat_b = b ** 2

# Calculem la suma dels quadrats
suma_quadrats = quadrat_a + quadrat_b

# Finalment, calculem el quadrat de la suma
resultat = suma_quadrats ** 2
print(resultat)

# També podem escriure aquesta mateixa expressió en una sóla línia
resultat = (a**2 + b**2)**2
print(resultat)
```

```
3721
3721
```

1.3.7 Exercici 4

Escriviu un programa que calculi el volum d'una piràmide quadrangular amb una base de 4x5 metres de longitud i amplada respectivament, i amb una altura de 7,5 m. Recorda que la fórmula per calcular el volum és

on l i w són la longitud i l'amplada de la base, i h és l'altura.

[16]:
```
# Resposta

# Assignem els valors del radi i l'altura
base_long = 4
base_amplada = 5
altura = 7.5

# A continuació calculem el volum utilitzant la següent fòrmula:
volum = 1/3 * base_long * base_amplada* altura

print('El volum de la piràmide és de ', volum, 'metres cúbics.')
```

```
El volum de la piràmide és de  49.99999999999999 metres cúbics.
```

1.3.8 Exercici 5

Escriviu un programa que defineixi una llista amb el nom de tots els mesos de l'any. Feu que mostri els mesos que corresponen a la tardor. **(1 punt)**

[17]:
```
# Definim la llista amb els mesos de l'any
mesos = ["Gener", "Febrer", "Març", "Abril", "Maig", "Juny", "Juliol", "Agost",␣
 ↪"Setembre", "Octubre", "Novembre", "Desembre"]

# Els mesos corresponents a la tardor son els mesos 9, 10 i 11
print (mesos[8:11])
```

```
['Setembre', 'Octubre', 'Novembre']
```

Fixeu-vos que en Python el primer element d'una llista o array el trobarem a la posició **0**. Per tant, si volem accedir al nové element de la llista utilitzarem l'index 8, és a dir, escriurem `mesos[8]`.

D'altra banda quan especifiquem un rang, `8:11`, l'element final no s'inclou en el resultat final. Es a dir, `mesos[8:11]` retornará els elements corresponents a les posicions 8, 9 i 10.

1.3.9 Exercici 6

Escriviu un programa que a partir de la llista de nombres donada mostri per pantalla una cadena de caràcters amb tots els números, separats entre ells per un guió, i duplicant el primer i l'últim element de la llista.

Així, per exemple, per a la llista ['1', '2', '3'], el programa hauria de mostrar la cadena: '1-1-2-3-3'.

[18]:
```
# Definim la llista de nombres d'entrada
nombres = ['6', '3', '45', '23', '2', '3', '17', '80']

# Resposta
nombres = [nombres[0]] + nombres + [nombres[-1]]
print('-'.join(nombres))
```

```
6-6-3-45-23-2-3-17-80-80
```

1.3.10 Exercici 7

Ordena la següent llista de cadenes de caràcters:

1. Invertint l'ordre original
2. En ordre alfabètic invers

Nota: Pots consultar la documentació oficial de la funció [`sorted`](https://docs.python.org/3/library/functions.html#sorted) per veure quins paràmetres pots fer servir per resoldre la segona part de l'activitat.

[19]:
```
st_chars = ["Benjamin Sisko", "Kira Nerys", "Odo", "Quark", "Jadzia Dax"]

# Hem vist l'ús dels dos punts (":") per accedir a múltiples elements d'una␣
 ↪llista.
# Python tracta l'expressió de la següent manera:
```

```
# - Si no hi ha cap índex abans de el primer ":" assumeix que ha de començar␣
 ↪pel principi de la llista.
# - Si no hi ha cap índex després de el primer ":" assumeix que s'ha de␣
 ↪posicionar al final de la llista.
# - L'últim número ens indica els increments en l'índex que s'han de fer a␣
 ↪mesura que recorrem la llista.

# (En aquest cas sempre "sumem" -1 a l'índex i per tant, obtenim -1, -2, -3,␣
 ↪etc.)
print(st_chars[::-1])
```

```
['Jadzia Dax', 'Quark', 'Odo', 'Kira Nerys', 'Benjamin Sisko']
```

Per al següent exercici, busquem la documentació de la funció `sorted()` i veiem que aquesta ens serveix per ordenar qualsevol seqüència (llista, tupla) i devuelverá una llista amb els elements de manera ordenada, sense modificar la seqüència original. El paràmetre *reverse* per defecte s'estableix com *fals*, però si ho establim com *True*, llavors el iterable s'ordenaria en ordre invers (descendent):

[20]:
```
print(sorted(st_chars, reverse=True))
```

```
['Quark', 'Odo', 'Kira Nerys', 'Jadzia Dax', 'Benjamin Sisko']
```

1.3.11 Exercici 8

A partir de la següent llista,

```
A_list = [42, 7.5,"Answer to the Ultimate Question","Dave", 7.5]
```

proporciona expressions que retornin:

1. El nombre de vegades que apareix l'element `7.5` a la llista.
2. La posició de la primera aparició de la valor `7.5`
3. La mateixa llista sense l'últim element

Nota: Al notebook de teoria hem vist què són les llistes i algunes operacions sobre elles. Per fer l'activitat, necessitareu investigar algunes operacions addicionals que podem realitzar sobre llistes. Per a això podeu consultar la documentació oficial de Python sobre llistes (intro i més sobre llistes).

[21]:
```
a_list = [42, 7.5, "Answer to the Ultimate Question", "Dave", 7.5]

# El mètode count() compta quantes vegades apareix un valor en una llista i el␣
 ↪retorna.
print (a_list.count (7.5))

# Farem servir el mètode index(), aquest troba el valor donat en una llista i␣
 ↪retorna la posició de l'element.
# Si el mateix valor apareix més d'una vegada, el mètode index () retorna la␣
 ↪posició de la seva primera aparició.
print (a_list.index (7.5))
```

```
# Indiquem que volem excloure l'últim element indexat de la llista usant␣
↪l'operador []:
print (a_list [: - 1])
```

```
2
1
[42, 7.5, 'Answer to the Ultimate Question', 'Dave']
```

1.3.12 Exercici 9

Què expressió en Python necessitem per aconseguir la *string* `Learning Python` utilitzant només les variables `str1` `istr2` definides?

[22]:
```
str1 = "I love Python, it's great!"
str2 = "Learning"
```

Com podem veure, per construir la cadena de text necessitem aïllar la sub-cadena "Python", que trobem a la tercera paraula de la cadena `str1` i correspon amb els caràcters 7 a 13 de la cadena. Podrem crear aquesta nova cadena de forma separada i a la fi concatenarla amb la cadena `str2` utilitzant l'operador+:

[23]:
```
# Accedim als caràcters que ens interessen de la cadena str1, incloent l'espai␣
↪previ a la paraula 'Python'
subcadena = str1 [6:13]

# Finalment, concatenamos la nova cadena (subcadena 'Python') i la concatenamos␣
↪amb str2
resultat = str2 + subcadena

# Mostrem el resultat
print (resultat)
```

```
Learning Python
```

[24]:
```
# També podem afegir l'espai en blanc entre paraules en el moment d'unirles,␣
↪vigileu de no afegir dos espais en blanc:
subcadena = str1 [7:13]
resultat = str2 + ' ' + subcadena
print (resultat)
```

```
Learning Python
```