Practice Exercise 1\_1 (expressions)

Noteer van elke expressie wat de uitkomst en het type is!

|  |  |  |
| --- | --- | --- |
|  | uitkomst | type |
| 5 | 5 | integer |
| 5.0 | 5.0 | float |
| 5 % 2 | 1 | integer |
| 5 > 1 | True | boolean |
| '5' | 5 | integer |
| 5 \* 2 | 10 | INTEGER |
| '5' \* 2 | 55 | integer |
| '5' + '2' | 52 | INTEGER |
| 5 / 2 | 2.5 | Float |
| 5 // 2 | 2 | INTEGER |
| [5, 2, 1] | [5,2,1] | string |
| 5 in [1, 4, 6] | true | bool |

Practice Exercise 1\_2 (strings)

Schrijf en evalueer Python expressies die de volgende vragen beantwoorden:

1. Hoeveel letters zijn er in 'Supercalifragilisticexpialidocious'?

print(len(**'Supercalifragilisticexpialidocious'**)) 🡪 34

1. Komt in 'Supercalifragilisticexpialidocious' de tekst 'ice' voor?

print(**'ice' in 'Supercalifragilisticexpialidocious'**)🡪 true

1. Is het woord 'Antidisestablishmentarianism' langer dan 'Honorificabilitudinitatibus'?

print(len(**'Antidisestablishmentarianism'**>len(**'Honorificabilitudinitatibus'**)) 🡪 true

1. Welke componist komt in alfabetische volgorde het eerst: 'Berlioz', 'Borodin', 'Brian',

'Bartok', 'Bellini', 'Buxtehude', 'Bernstein'? Welke het laatst?

list =[ **'Berlioz'**, **'Borodin'**, **'Brian'**,**'Bartok'**, **'Bellini'**, **'Buxtehude'**, **'Bernstein'**]  
print(min(list), max(list)) 🡪 Bartok Buxtehude

Practice Exercise 1\_3 (statements)

Schrijf Python statements die het volgende doen:

1. Ken de waarde 6 toe aan variabele a, en waarde 7 aan variabele b.

a = 6; b = 7

1. Ken aan variabele c als waarde het gemiddelde van a en b toe.

c = (a + b)/2

1. Ken aan variabele inventaris de een lijst van strings toe: ‘papier’, ‘nietjes’, en ‘pennen’.

inventaris = [**'papier','nietjes','pennen'**]

1. Ken aan variabelen voornaam, tussenvoegsel en achternaam je eigen naamgegevens toe.

voornaam = **'Wesley'**tussenvoegsel =**''**achternaam =**'Ackah'**

1. Ken aan variabele mijnnaam de variabelen van opdracht 4 (met spaties er tussen) toe.

mijnNaam = [voornaam, tussenvoegsel, achternaam]

Practice Exercise 1\_4 (boolean expressions)

Schrijf booleaanse expressies die van de variabelen van Practice Exercise 1\_3 evalueren of:

1. 6.75 groter is dan a en kleiner b.

print( a < 6.75 **and** 6.75 < b)🡪 true

1. de lengte van inventaris meer dan 5 keer zo groot is als de lengte van variabele mijnnaam.

print(len(inventaris) > 5 \*len(mijnNaam)) 🡪 false

1. de lijst inventaris leeg is, of juist meer dan 10 items bevat. print(len(inventaris)==0 **or** len(inventaris) >10) 🡪 false

Practice Exercise 1\_5 (lists)

We gaan een lijst bijhouden met je favoriete artiesten. We gaan de lijst eerst creëren met 1 artiest en

dan uitbreiden. Schrijf per stap een expressie om:

1. een nieuwe list met 1 artiest aan te maken met de naam favorieten.

favorieten = [**'kendrick lamar'**]

1. de lijst uit te breiden met een tweede artiest.

favorieten.append(**'j cole'**)

1. de tweede artiest te vervangen door een andere naam.

favorieten[1]=(**'kanye west'**)

Practice Exercise 1\_6 (lists)

Het bereik van een lijst is het verschil tussen het grootste en het kleinste getal. Schrijf een Python

expressie die het bereik van een lijst berekent. Als bijvoorbeeld variabele list bestaat uit de getallen 3,7, -2 en 12, dan moet de expressie evalueren naar 14 (verschil tussen 12 en -2). Zorg dat de expressiealtijd werkt, ook al bestaat de lijst uit andere waarden.

lst = [3,7,12,-2]  
print(abs(min(lst))+max(lst))

Practice Exercise 2\_1 (tuples)

De tuple letters kan in willekeurige volgorde de letters A, B en C bevatten. Bijvoorbeeld:

letters = ('A', 'C', 'B', 'B', 'C', 'A', 'C', 'C', 'B')

Maak een nieuw bestand, bijvoorbeeld pe2\_1.py, en schrijf een programma dat een nieuwe lijst maakt(en print) met het aantal voorkomens van de letters in alfabetische volgorde. Tuple letters bevat 4 keer ‘A’, 3 keer ‘B’ en 5 keer ‘C’. De lijst die dit programma maakt (en print) is dan: [2, 3, 4].

letters = (**'A'**, **'C'**, **'B'**, **'B'**, **'C'**, **'A'**, **'C'**, **'C'**, **'B'**)  
telling = [letters.count(**'A'**), letters.count(**'B'**), letters.count(**'C'**)]  
print(telling)

Practice Exercise 2\_2 (getallen, strings en conversie)

De Hogeschool Utrecht wil studenten financieel ondersteunen bij hun studie, afhankelijk van de cijfers die een student haalt. Voor elk cijferpunt krijg je € 30,-. Voor een 1,0 krijgt je dus 30 euro, voor een 2,5 krijgt je 75 euro en voor een 10,0 beloont de HU een student met € 300,-. Maak variabelen cijferICOR, cijferPROG en cijferCSN. Geef ze alle drie de waarde die jij verwacht dat je voor de betreffende vakken in blok 1 zult gaan halen. Maak nu vervolgens ook de volgende variabelen aan, en bereken de bijbehorende waarden m.b.v. een Python expressie:

Gemiddelde: het gemiddelde van de variabelen cijferICOR, cijferPROG en cijferCSN.

Beloning: de totale beloning voor deze drie cursussen.

Overzicht: een string met een tekstuele omschrijving het gemiddelde en de beloning:

'Mijn cijfers (gemiddeld een 7.5) leveren een beloning van € 675.0 op!'

Print tot slot variabele overzicht! Schrijf dit programma in een nieuw bestand, bijvoorbeeld pe2\_2.py.

cijferICOR = 7  
cijferPROG = 6.5  
cijferCSN = 6  
gemiddelde = (cijferCSN +cijferPROG + cijferICOR)/3  
beloning = 3\*gemiddelde\*30  
overzicht = **f"Mijn cijfers (gemiddeld een {gemiddelde}) leveren een beloning van € {beloning} op!"**print(overzicht)

Practice Exercise 2\_3 (operator voorrang) Voeg haakjes toe aan de volgende expressies zodat ze naar True evalueren. Print het resultaat!

1. 0 == 1 == 2

2. 2 + 3 == 4 + 5 == 7

3. 1 < -1 == 3 > 4

Practice Exercise 2\_4 (Input/Output) Schrijf een programma dat de gebruiker vraagt om zijn uurloon, het aantal uur dat hij of zij gewerkt heeft en dat daarna het salaris uitprint.

salaris = eval(input(**'Wat is je salaris per uur?'**))  
tijd = eval(input(**'Hoeveel uur heb je gewerkt?'**))  
Geld = salaris\*tijd  
print(**f'{tijd} uur werken levert {Geld} Euro op'**)

Final Assignment: Git & GitHub

Alle stappen uitgevoerd.

Github naam is wesackah.

Alle py bestanden staan daar ook.