

TP1 Python - Correction des exercices

Exercice 1

1. Il y a 64 cases sur un échiquier. Le nombre de grains de riz sur chaque case forme une suite géométrique de raison 2, avec $u_1 = 1$, donc $u_{64} = u_1 \times 2^{64-1} = 2^{63}$. Ainsi, il y aura

$2^{63} = 9223372036854775808$ grain de riz sur la dernière case.

2. Formule générale : $\sum_{k=n}^m q^k = \frac{q^n - q^{m+1}}{1-q}$. Ici $q=2$, $n=0$, $m=63$.

On obtient $1+2+2^2+\dots+2^{63} = \sum_{k=0}^{63} 2^k = \frac{1-2^{64}}{1-2} = 2^{64}-1$
 $= 18446744073709551615$ grains de riz

au total sur l'échiquier.

3. Il suffit de multiplier : $(2^{64}-1) \times 0,04 \approx 7,379 \times 10^{17}$ g de riz

4. la production annuelle est de $699 \times 10^6 \times 10^6 = 6,99 \times 10^{14}$ g de riz.

On calcule alors : $\frac{(2^{64}-1) \times 0,04}{6,99 \times 10^{14}} \approx 1055,6$ ans nécessaires pour donner sa récompense à Sissa!

Exercice 2 - $4560 // 7 = 651$

$$4560 \% 7 = 3$$

Ainsi, $4560 = 7 \times 651 + 3$. Dans 4560 jours, il se sera passé 651 semaines et 3 jours. Le TP d'info est un mercredi : nous serons un samedi, et 651 dimanches seront passés.

Exercice 3 -

$\sin(\pi) \rightarrow \approx 1,225 \times 10^{-16}$ au lieu de 0 } calcul avec des flottants
 $\cos(\pi/2) \rightarrow \approx 6,123 \times 10^{-17}$ au lieu de 0 } approximations

$$\log(e) \rightarrow 1,0$$

$$\text{sqr}(2) \rightarrow \approx 1,41421$$

$$\exp(0) \rightarrow 1,0$$

$\text{sqr}(-3)$ renvoie l'erreur "math domain error" : on ne peut pas prendre la racine carrée d'un nombre négatif.

$$\text{int}(-3.3) \rightarrow -3$$

tronqué

$$\text{floor}(-3.3) \rightarrow -4$$

plus grand entier $\leq -3,3$.

Exercice 4

$\left\{ \begin{array}{l} \text{floor}(x) \text{ renvoie le plus grand entier } n \text{ tel que } n \leq x \\ \text{ceil}(x) \text{ renvoie le plus petit entier } n \text{ tel que } x \leq n \\ \text{round}(x) \text{ arrondit } x \text{ à l'entier le plus proche} \\ \text{round}(x, n) \text{ donne un arrondi de } x \text{ avec une précision à } 10^{-n} \text{ près.} \end{array} \right.$

Ainsi: $\text{floor}(-3.5) = \underline{-4}$; $\text{ceil}(-3.5) = \underline{-3}$; $\text{round}(3.5) = \underline{3}$;
 $\text{round}(3.557, 1) = \underline{3.6}$ (1 chiffre après la virgule, 10^{-1}).
 $\text{round}(3516.2, -2) = \underline{3500}$ (2 chiffres avant la virgule, 10^2).

Exercice 5

$\log(x, b)$ renvoie $\log_b(x) = \frac{\ln(x)}{\ln(b)}$. $\log(x)$ renvoie simplement $\ln(x)$.

Ainsi: x se calcule par $\log(16, 2)$: $x = \underline{4.0}$

y se calcule par $\log(16, 4)$: $y = \underline{2.0}$

z s'obtient par $\log(100, 10)$: $z = \underline{2.0}$

On a $2^x = 16$, $4^y = 16$, $10^z = 100$: on peut conjecturer que
 $b^{\log_b(x)} = x$ pour tout b et x réels strictement positifs.

$\ln(10)$ se calcule simplement avec $\log(10)$. Le logarithme népérien correspond à la base e : $\forall x > 0$, $\log_e(x) = \frac{\ln(x)}{\ln(e)} = \frac{\ln(x)}{1} = \ln(x)$.

Exercice 6 On commence par: `from random import *`

1. `random() * 70`

2. `random() * 70 - 20`

3. `random() * 5.5 + 3`

Exercice 7

- Le 1^{er} script affiche 1 puis renvoie une erreur car `A` n'est pas défini.
- Le 2^e script affiche 3
- Le 3^e script renvoie une erreur car la variable `bonjour` n'est pas définie.

Exercice 8

1. On vient de calculer l'effectif de la population 1 après 1 an, au 1^{er} janvier 2016.

2. On exécute 7 fois l'instruction $\text{eff1} = \text{eff1} \times 1.05$
eff1 est alors de 1407 (arrondi à l'unité).

Pour eff2, on exécute $\text{eff2} = 700$, puis 7 fois l'instruction:

$$\text{eff2} = \text{eff2} \times 1.07$$

On obtient eff2 à 1124 (arrondi à l'unité).

Ainsi, le 1^{er} janvier 2022, la population 1 sera encore la plus nombreuse.

Exercice 9 - Il faut avoir fait: `from random import *`

`de1 = randint(1, 6)`

`de2 = randint(1, 6)`

`de3 = randint(1, 6)`

`print(de1 + de2 + de3)`