

# nbChiffres

Documents d'accompagnement  
Algorithmique et programmation  
nouveau programme du lycée 2019

## 1 Nombre de chiffres d'un nombre donné

### 1.1 Présentation de l'activité

- **Niveau de classe :**
- Classe de seconde générale et technologique
- **Référence au programme :**
- Seconde: *Déterminer la première puissance d'un nombre positif donné supérieure ou inférieure à une valeur donnée.*
- **Description :** activité de décompte du nombre de chiffres d'un nombre fixé permettant de faire travailler la notion de puissance et de faire réfléchir sur la notion de base.

### 1.2 Cas général

Dans cette partie, on cherche à écrire un programme qui détermine le nombre de chiffres de l'écriture décimale d'un entier naturel. Pour cela, on remarque que  $10^n$  a exactement  $n + 1$  chiffres.

Suggestions pédagogiques

- **Mathématiques débranchées**

De combien de chiffres est constituée l'écriture d'un nombre entier  $a$  tel que  $10^4 \leq a < 10^5$  ?  
tel que  $10^n \leq a < 10^{n+1}$  ?

La fonction `nombreChiffres` prend en paramètre un entier naturel  $a$  non nul et renvoie l'exposant de la plus petite puissance de 10 ( $\geq 1$ ) strictement supérieure au nombre  $a$  (cela correspond au nombre de chiffres de  $a$ ).

---

```
def nombreChiffres(a):  
    n = 0  
    while 10**n <= a:  
        n = n+1  
    return n  
puissance(109)
```

---

Out [6]: 3

Suggestions pédagogiques

- **Expliquer un programme**

À quelle condition s'arrête la boucle `while` ? À quoi servent les lignes 2 et 4 ?

- **Compléter un programme**

Le programme précédent étant fourni en remplaçant les lignes 2, 3 et 4 par `n=...`, `while ...` et `n=...`, demander aux élèves de compléter les lignes 2, 3 et 4.

- **Écrire un programme**

Écrire une fonction prenant en entrée un entier  $a$  et renvoyant l'exposant de la plus petite puissance de 10 strictement supérieure à  $a$ .

- **Remarque** On peut ajouter une ligne au début de la fonction pour vérifier que  $a$  est un entier strictement positif: `assert type(a)==int and a>0, 'Attention a doit etre un entier positif'`

### 1.3 Cas des puissances de 2

Comme application, on cherche maintenant à connaître le nombre de chiffres des puissances de 2. Suggestions pédagogiques

- **Tester un programme**

En utilisant le programme `nombreChiffres`, déterminer le nombre de chiffres des nombres  $2^{10}$ ,  $2^{64}$  et  $2^{100}$ .

---

```
nombreChiffres(2**10), nombreChiffres(2**64), nombreChiffres(2**100)
```

---

Out [7]: (4, 20, 31)

Suggestions pédagogiques

- **Tester un programme**

Calculer  $2^{10}$  et dire si le dernier résultat est cohérent.

Réponse

$2^{10} \simeq 10^3$ , on peut donc avoir l'idée que  $2^{100} = (2^{10})^{10}$  sera estimé par  $(10^3)^{10} = 10^{30}$ . On retrouve les 30 chiffres annoncés.

### 1.4 Représentation du nombre de chiffres des puissances de 2

Dans cette partie, nous cherchons à représenter l'évolution du nombre de chiffres des puissances de 2.

---

```
%matplotlib inline
import matplotlib.pyplot as plt
```

---

Afin de représenter l'évolution du nombre de chiffres des puissances de 2, nous allons placer au fur et à mesure les points  $(i, 2^i)$  pour  $i$  parcourant les  $n$  premiers entiers ( $n$  variables fixées).

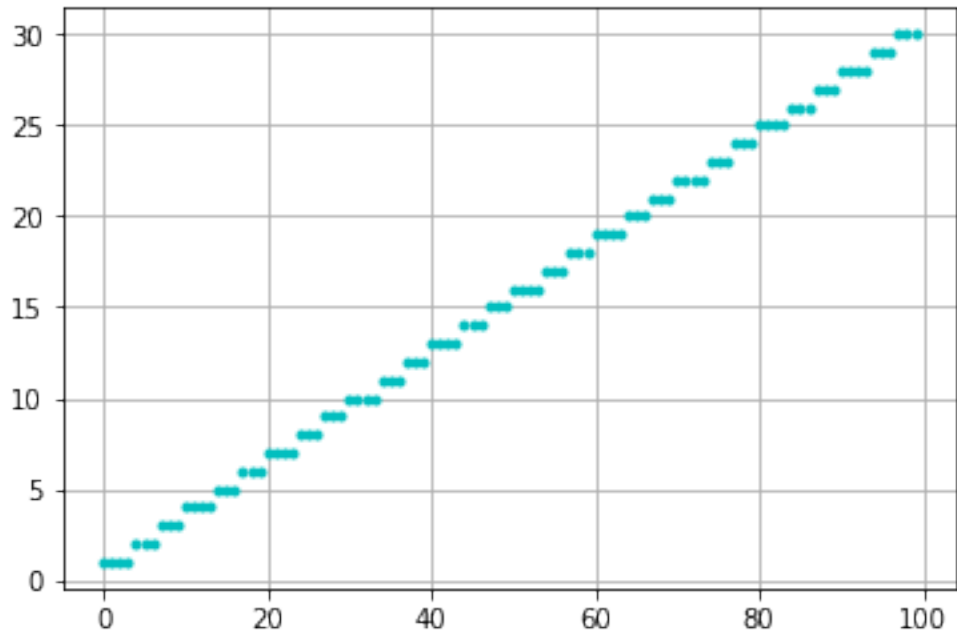
---

```
n = 100
```

```
for i in range(n):  
    plt.plot(i, nombreChiffres(2**i), 'c.')
```

```
plt.grid(True)  
plt.show()
```

---



Suggestions pédagogiques

- **Décrire**

Qu'est-ce qui explique la forme en escalier de cette représentation graphique ? Quelle est la forme approximative de cette courbe ?

- **Compléter un programme**

Le programme précédent étant fourni en remplaçant la ligne 4 par `plt.plot(i, ..., 'c.')`, demander aux élèves de compléter la ligne 4.