

Format des nombres en Python

1/ Lors d'une saisie au clavier

Lors de la saisie d'un nombre entier au clavier, plusieurs formats sont acceptés en langage Python

1.1/ Pour les entiers

Un nombre entier peut être saisi sous différents formats :

| Format de saisie | Exemple |
|---|--------------------|
| Format décimal : seuls les chiffres sont saisis. | >>> 1234 1234 |
| Format binaire : les chiffres sont précédés du préfixe « 0b ». | >>> 0b11101 29 |
| Format hexadécimal : les chiffres sont précédés du préfixe « 0x » (majuscule ou minuscule indifférente) | >>> 0X12bA 4794 |

1.1/ Pour les nombres à virgule flottante

Un nombre saisi au clavier sera reconnu comme un nombre à virgule flottante si il contient un point « . » ou s'il contient « e » ou « E ».

| Format de saisie | Exemple |
|---|--|
| Ecriture naturelle : la virgule est située à droite du chiffre des unités (virgule fixe). | >>> 1234. 1234.0 |
| Ecriture scientifique : la virgule est située à droite du premier chiffre, le nombre est pondéré par une puissance de 10, noté « e » ou « E » : $1,234 \cdot 10^3$ | >>> 1.234e3 1234.0 >>> 1.234E3 1234.0 |

2/ Pour l'affichage

Il est aussi possible d'afficher un nombre selon un format désiré en utilisant la fonction `format(valeur, 'option')` qui crée une chaîne de caractères contenant la valeur écrite selon un format précisé dans 'option'.

2.1/ Pour les entiers

| Type | Signification | Exemple |
|------|--|--|
| 'b' | Format binaire. Affiche le nombre en base 2. | >>> format(1234, 'b') '10011010010' |
| 'x' | Format hexadécimal. Affiche le nombre en base 16 en utilisant les lettres minuscules pour les chiffres au-dessus de 9. | >>> format(1234, 'x') '4d2' |
| 'X' | Format hexadécimal. Affiche le nombre en base 16 en utilisant les lettres majuscules pour les chiffres au-dessus de 9. | >>> format(1234, 'X') '4D2' |
| 'c' | Caractère. Convertit l'entier en le caractère unicode associé avant de l'afficher. | >>> format(1234, 'c') 'Ä' |

2.2/ Pour les nombres à virgule flottante

Dans ce cas, les options doivent indiquer la précision et/ou le format choisi pour le nombre à virgule, sous la forme '`.précisionformat`' :

- la précision correspond au nombre de chiffres affichés après la virgule ;
- le format est à choisir parmi les options ci-dessous :

La précision par défaut est 6.

| Type | Signification | Exemple |
|------|---|--|
| 'f' | Virgule fixe. Affiche le nombre comme un nombre à virgule fixe. | >>> format(12345.67890, '.3f') '12345.679' |
| 'e' | Affiche le nombre dans sa notation scientifique en utilisant la lettre "e" pour indiquer l'exposant. | >>> format(12345.67890, '.2e') '1.23e+04' |
| 'E' | Notation par exposant. Pareil que 'e' sauf l'utilisation de la lettre majuscule "E" comme séparateur. | >>> format(12345.67890, '.2E') '1.23E+04' |
| 'g' | Format général. Python choisit le format le plus ergonomique, en virgule fixe ou en notation scientifique, en fonction de la magnitude. | >>> format(12345.67890, 'g') '12345.7' |
| '%' | Pourcentage. Multiplie le nombre par 100 et l'affiche en virgule fixe ('f'), suivi d'un symbole pourcent '%' | >>> format(12345.67890, '.1%') '1234567.9%' |

La fonction `format()` possède des usages et options plus larges, mais non développés ici.