

Les fonctions natives

Dès son lancement, l'interpréteur Python met à disposition du développeur plusieurs fonctions natives (*built-in functions* en anglais), comme la fonction `print()`. Ce ne sont pas des mots-clés du langage. Nous présentons les fonctions les plus couramment utilisées.

1. Les fonctions natives de conversion

Nous les avons déjà rencontrées, il s'agit des fonctions permettant de convertir un type en un autre type : `int()`, `float()`, `str()`, et des fonctions permettant de passer d'un code Unicode au caractère associé : `chr()` et `ord()`.

Il existe deux fonctions permettant de convertir un nombre entier dans sa représentation binaire (suite de 0 et de 1) ou hexadécimale (nombre écrit en base 16). Il s'agit des fonctions `bin()` et `hex()` retournant toutes deux une chaîne de caractères. Ainsi, `bin(128)` retourne `'0b10000000'` et `hex(128)` retourne `'0x80'`.

2. Les fonctions natives mathématiques

Nous trouvons des fonctions très utiles comme le calcul du maximum ou de la valeur absolue. Nous les présentons ci-après :

- `abs(-3) → 3` : calcule la valeur absolue.
- `max(4, 5) → 5`.
- `max(4, 5, 6) → 6` : calcule le maximum entre trois valeurs.
- `min(4, 5, 6) → 4`.
- `round(7.2) → 7` et `round(7.8) → 8` : arrondit à l'entier le plus proche.
- `round(7.123456, 3) → 7.123` : arrondit à trois chiffres après la virgule.



Hormis la fonction `round()`, les autres fonctions s'adaptent automatiquement au type des données en entrée pour "choisir" leur type de retour. Par exemple, lorsque la fonction `max()` reçoit deux entiers, elle retourne un nombre entier. Ainsi, `max(1, 4)` retourne 4. Lorsqu'on lui passe deux nombres de type float, la valeur retournée est de type float. Ainsi, `max(4.3, 2.1)` retourne 4.3. Si cela vous semble une évidence, sachez que dans d'autres langages, les choses ne sont pas aussi aisées.

3. La fonction native input pour les entrées-clavier

Il est possible de demander à l'utilisateur d'entrer une réponse au clavier, ceci grâce à la fonction `input()`. Celle-ci lit les touches-clavier, les affiche à l'écran jusqu'à ce que la touche [Entrée] soit appuyée. Après cela, la saisie est terminée et la fonction `input()` retourne une chaîne de caractères. Ainsi, même si vous tapez un nombre comme 123, le retour de la fonction est de type string : `"123"`. Si vous voulez traiter le nombre associé, il faut alors convertir la chaîne en nombre entier avec la fonction `int()`. Voici un exemple de programme qui calcule le carré d'un nombre entré au clavier :

```
print("Entrez un nombre : ")
reponse = input()
print(type(reponse))
valeur = int(reponse)
print("Voici ce nombre au carré :", valeur*valeur)
```

```
>> Entrez un nombre :  
>> 50  
>> <class 'str'>  
>> Voici ce nombre au carré : 2500
```

4. Les fonctions natives avancées

Il existe une cinquantaine de fonctions natives, certaines répondent à des besoins très pointus et dépassent largement le cadre de cet ouvrage. Nous en citons quelques-unes à titre d'exemple :

- `type ()` : retourne le type d'une variable. Très utile lorsque vous avez un doute sur le type de retour d'une fonction, elle permet de lever l'ambiguïté.
- `del ()` : détruit explicitement une variable, son nom n'est plus accessible.