

Liste des fonctions natives Python (en français)

`abs(x)` : Retourne la valeur absolue de `x`.

`all(iterable)` : Retourne `True` si tous les éléments de l'itérable sont vrais.

`any(iterable)` : Retourne `True` si au moins un élément de l'itérable est vrai.

`ascii(obj)` : Retourne une version imprimable de l'objet en ASCII.

`bin(x)` : Convertit un entier en chaîne binaire.

`bool(x)` : Convertit une valeur en booléen (`True` ou `False`).

`bytearray([source[, encodage[, erreurs]])` : Retourne un tableau d'octets mutable.

`bytes([source[, encodage[, erreurs]])` : Retourne un objet d'octets immuable.

`callable(obj)` : Retourne `True` si l'objet peut être appelé comme une fonction.

`chr(i)` : Retourne le caractère Unicode correspondant à l'entier `i`.

`classmethod(méthode)` : Transforme une méthode en méthode de classe.

`compile(source, nom_fichier, mode)` : Compile du code source en objet exécutable.

`complex(reel, imag)` : Crée un nombre complexe.

`delattr(obj, nom)` : Supprime un attribut d'un objet.

`dict()` : Crée un dictionnaire.

`dir([objet])` : Retourne la liste des attributs et méthodes d'un objet.

`divmod(a, b)` : Retourne le couple `(a // b, a % b)`.

`enumerate(iterable, start=0)` : Retourne un itérateur énumérant les éléments avec leur indice.

`eval(expr[, globals[, locals]])` : Évalue une chaîne comme une expression Python.

`exec(objet[, globals[, locals]])` : Exécute du code Python dynamiquement.

`filter(fonction, iterable)` : Filtre les éléments d'un itérable selon une fonction.

`float(x)` : Convertit `x` en nombre flottant.

`format(valeur[, format_spec])` : Formate une valeur en chaîne.

`frozenset([iterable])` : Retourne un ensemble immuable.

`getattr(obj, nom[, défaut])` : Retourne la valeur de l'attribut d'un objet.

`globals()` : Retourne le dictionnaire des variables globales.

`hasattr(obj, nom)` : Vérifie si un objet possède un attribut.

`hash(obj)` : Retourne le hachage d'un objet.

`help([objet])` : Affiche l'aide interactive de Python.

`hex(x)` : Retourne la représentation hexadécimale d'un entier.

`id(obj)` : Retourne l'identifiant unique d'un objet.

`input([invite])` : Lit une entrée de l'utilisateur.

`int(x)` : Convertit `x` en entier.

`isinstance(obj, classe)` : Vérifie si un objet est instance d'une classe.

`issubclass(classe, classe_info)` : Vérifie si une classe hérite d'une autre.

`iter(obj)` : Retourne un itérateur à partir d'un objet.

`len(obj)` : Retourne la taille d ' un objet.

`list([iterable])` : Crée une liste.

`locals()` : Retourne le dictionnaire des variables locales.

`map(fonction, iterable)` : Applique une fonction à chaque élément d ' un itérable.

`max(iterable)` : Retourne le plus grand élément.

`min(iterable)` : Retourne le plus petit élément.

`next(itérateur[, défaut])` : Retourne l ' élément suivant d ' un itérateur.

`object()` : Retourne une nouvelle instance d ' objet de base.

`oct(x)` : Retourne une chaîne en base octale.

`open(fichier, mode)` : Ouvre un fichier et retourne un objet fichier.

`ord(c)` : Retourne le code Unicode d ' un caractère.

`pow(x, y[, z])` : Retourne $(x ** y) \% z$ si z est fourni.

`print(*objets)` : Affiche les objets.

`property(fget, fset, fdel)` : Retourne une propriété d ' un objet.

`range(stop)` : Retourne un itérateur de nombres.

`repr(obj)` : Retourne la représentation en chaîne d ' un objet.

`reversed(seq)` : Retourne un itérateur inversé.

`round(x[, n])` : Retourne x arrondi à n décimales.

`set([iterable])` : Retourne un ensemble mutable.

`setattr(obj, nom, valeur)` : Assigne un attribut à un objet.

`slice(start, stop[, step])` : Retourne un objet slice.

`sorted(iterable)` : Retourne une liste triée.

`staticmethod(méthode)` : Transforme une méthode en méthode statique.

`str(obj)` : Retourne une chaîne de caractères.

`sum(iterable)` : Retourne la somme des éléments.

`super([type[, obj]])` : Retourne un objet proxy pour déléguer des appels à la classe parente.

`tuple([iterable])` : Crée un tuple.

`type(obj)` : Retourne le type d ' un objet.

`vars([obj])` : Retourne `__dict__` d ' un objet.

`zip(*iterables)` : Retourne un itérateur regroupant les éléments.

`__import__(nom)` : Importe un module.