

Documentations

a

Fonctions natives

<code>all(iterable: Iterable)</code> -> bool	Renvoie 'VRAI' si toutes les éléments de l'objet itéré valent 'VRAI'.
<code>any(iterable: Iterable)</code> -> bool	Renvoie 'VRAI' si au moins un élément de l'objet itéré vaut 'VRAI'.
<code>breakpoint()</code>	Place un point d'arrêt sur la ligne où est appelé la fonction puis instancie un débogueur.
<code>dir(object: Optional[Any])</code> -> list[str]	Retourne la liste des noms des attributs de l'objet.
<code>enumerate(iterable: Iterable, start: int = 0)</code> -> Iterator[list[int, Any]]	Retourne un couple (<code>index</code> , <code>objet[index]</code>) de la liste. pour chaque élément.
<code>filter(function: Optional[Callable], iterable: Iterable)</code> -> Iterator	Retourne les éléments de l'objet itéré qui lesquels la fonction renvoie 'VRAI'.
<code>help(object: Any)</code>	Affiche la documentation de l'objet.
<code>isinstance(object: Any, type: [list[Type], Type])</code> -> bool	Retourne 'VRAI' si l'objet est du type spécifié ou un sous-type.
<code>map(function: Callable, *iterables: list[Iterable])</code> -> Iterator	Retourne les éléments de l'objet itéré après application de la transformation décrite dans la fonction spécifiée.
<code>range(start: int, stop: Optional[int], step: Optional[int])</code> -> Iterator	Retourne une séquence qui débute par <code>start</code> jusqu'à <code>stop</code> exclu avec un pas de <code>step</code> .
<code>type(object: Any)</code> -> Type	Retourne le type de l'objet.
<code>zip(*iterables: list[Iterable], strict: bool = False)</code> -> Iterator	Retourne une liste de tuple qui contient le i-ème élément de chaque liste.

Anti-Patterns

Mauvaise pratique

```
name_list = ["alice", "bob", "caroline", "david"]
for i in range( len (na me_ list)):
    pri nt( nam e_l ist[i])
```

Récupérer les éléments d'une liste.

Mauvaise pratique

```
name_list = ["alice", "bob", "caroline", "david"]
for i in range( len (na me_ list)):
    pri nt(i, name_l ist[i])
```

Récupérer les éléments d'une liste et son index.

Bonne pratique

```
name_list = ["alice", "bob", "caroline", "david"]
for name in name_list:
    pri nt( name)
```

Récupérer les éléments d'une liste.

Bonne pratique

```
name_list = ["alice", "bob", "caroline", "david"]
for i, name in enumer ate (na me_ list):
    pri nt(i, name)
```

Récupérer les éléments d'une liste et son index.



By **bobby777**
cheatography.com/bobby777/

Not published yet.
Last updated 18th October, 2023.
Page 2 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>