

List comprehension

Présentation des « comprehension list »

Un bon développeur cherchera toujours à augmenter sa productivité avec le temps. Il existe des astuces python qui permettent d' optimiser le code .

Une de ces astuces est la compréhension de liste (ou liste en compréhension ou list comprehension).

L'idée est simple: simplifier le code pour le rendre plus lisible et donc plus rapide à écrire et plus simple à maintenir.

Présentation des « comprehension list »

```
cashier = []
for item in cart:
    cashier.append(item)
```

Non-list comprehension

```
cashier = [item for item in cart]
```

List comprehension

```
cashier_3 = []
for item in cart:
    if item % 2 == 0:
        cashier_3.append(item)
```

Non-list comprehension

```
cashier_3 = [item for item in cart if item % 2 == 0]
```

List comprehension

LIST COMPREHENSION

Output

Collection

Condition

[x+1 for x in range(5) if x%2 == 2]

Do this for this collection In this situation

Exemples

```
lst = [1,2,3,4,5,6,7,8,9,10]
a = [x for x in lst]
print(a)
```

output

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
for x in lst:
    a.append(x)
```

ajouter n'importe quel nombre à chaque élément de lst et le stocker dans un

```
a = [x+1 for x in lst]
```

soustraire n'importe quel nombre à tous les éléments de lst et le stocker dans un

```
a = [x-1 for x in lst]
```



Exemples

```
lst = [1,2,3,4,5,6,7,8,9,10]
# avec filtre
c = [x for x in lst if x > 4]
print(c)
```

```
# avec plusieurs if
d = [x for x in lst if x > 4 if x%2 == 0]

# output
[6, 8, 10]
```

```
lst = [1,2,3,4,5,6,7,8,9,10]
# avec if et else
e = [x if x > 4 else 'less than 4' for x in lst]
print(e)
```

```
for x in lst:
    if x > 4:
        a.append(x)
```

```
for x in lst:
    if x > 4:
        if x % 2 == 0:
            a.append(x)
```

```
for x in lst:
    if x > 4:
        d.append(x)
    else:
        d.append('less than 4')
```



Exemples

```
# plusieurs if else conditions
f = ['Two' if x%2 == 0 else "Three" if x%3 == 0 else 'not 2 & 3' for x in lst]
print(f)
```

```
lst = [1,2,3]
lst_rev = [3,2,1]
g = [(x,y) for x in lst for y in lst_rev]
print(g)

#output
[(1, 3), (1, 2), (1, 1), (2, 3), (2, 2), (2, 1), (3, 3), (3, 2), (3, 1)]
```

```
for x in lst:
    if x%2 == 0:
        f.append('Two')
    elif x%3 == 0:
        f.append('Three')
    else:
        f.append('not 2 & 3')
```

```
#equivalent
for x in lst:
    for y in lst_rev:
        f.append((x,y))
```