Suite TP2 - Itérables et boucles for

Comment faire n tours de boucle ?

Python propose un type itérable qui permet de représenter la séquence des valeurs entières contenues dans un intervalle. Ce type s'appelle range.

Pour créer une telle séquence, on fournit :

- la borne inférieure de l'intervalle (cette borne est *incluse* dans la séquence)
- la borne supérieure de l'intervalle (cette borne est *exclue* de la séquence).

On peut ne pas fournir la borne inférieure, qui vaut alors par défaut 0.

Par exemple :

- range(2, 6) contient la séquence des valeurs entières 2, 3, 4, 5
- range(6) contient la séquence des valeurs entières 0, 1, 2, 3, 4, 5

On a bien :

```
>>> type(range(6))
<class 'range'>
```

Comme un objet de type range est itérable, on peut parcourir ses valeurs avec une boucle for.

Récupérer sur le portail le fichier activite_range.py.

1. Dans ce fichier, compléter la signature et le code de la procédure afiche_range() en supprimant le mot pass.
2. Utiliser des appels à cette fonction pour afficher dans la console les valeurs associées à :

- range(2, 6)
- range(6)

On constate que range(n) contient n valeurs, pour n positif ou nul. Itérer sur range(n) exécutera donc n itérations.

Pour demander n saisies à l'utilisateur, ou encore générer une liste contenant n valeurs, on veut justement itérer n fois.

Comme on n'utilisera alors pas les valeurs contenues dans l'intervalle range(n), on pourra utiliser un souligné _[1] à la place de la variable d'itération.

3. Observer le code de la fonction compte_iterations() et exécuter ses tests.
4. Observer le code de la fonction saisie_caracteres() et compléter sa documentation.

Vous pouvez revenir au TP6 de programmation ! Pour le moment vous ne devez utiliser range que pour l'usage "faire n tours de boucles". On verra plus tard dans quels autres algorithmes les range peuvent servir.

[1] Du 8, pas du 6 !