

Les fonctions : Rappels

Les fonctions : une recette générique

Paramètres

Recette générique : `bouilli(aliment, durée)`

Remplir à moitié une casserole d'eau froide. Placer **l'aliment** dans l'eau. Placer la casserole sur le feu, Lorsque l'eau bout, tournez lentement et faites bouillir pendant **durée**. Enlever **l'aliment**.

Ainsi, maintenant nous pouvons dire :

- `bouilli(oeuf, 3 minutes)`
- `bouilli(patate, 20 minutes)`
- `bouilli(XXX, Y minutes)`

Deux sortes de fonctions

1. Procédures

(Écrire une fonction qui **affiche...**)

- **Ne retourne pas** de valeur
- **Affiche** quelque chose en sortie
- Leur nom commence par un verbe: `affiche_table`, `affiche_resultat`
- **Appelée de cette manière :**
`affiche_table(10)`
`affiche_resultat(data)`

2. Fonctions (véritables)

(Écrire une fonction qui **retourne...**)

- **Retourne** une valeur
- **N' affiche pas** quelque chose en sortie
- Leur nom sont des noms: `moyenne_note`, `max_temp...`
- **Appelée de cette manière :**
`moyenne = moyenne_note(data)`
`print(max_temp(temps))`

Utilité des fonctions

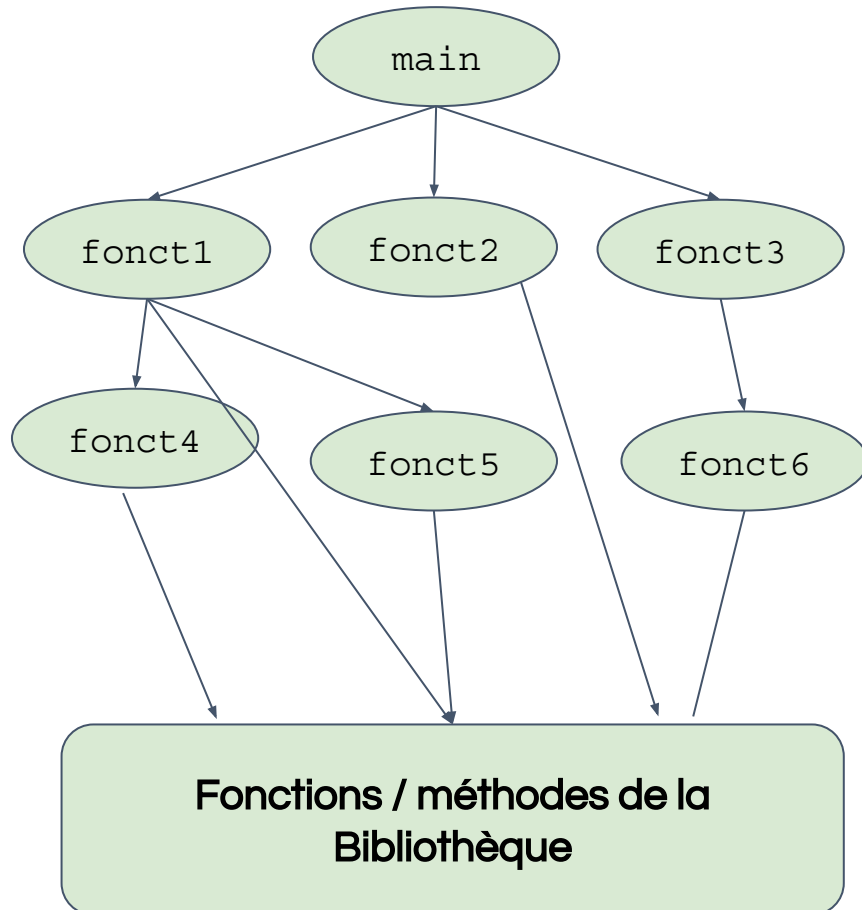
Les deux raisons principales pour écrire des fonctions

- Pour faire un code plus facile à comprendre
 - Découper un gros problème en plus petites parties (nommées)
 - Qui peuvent contenir aussi des sous problèmes
 - Travailler un seul sous problème à la fois
- Pour faire du code réutilisable
 - Dans un même module
 - Dans un autre module

Décomposition d'un programme en fonction

- Les bons programmes consistent en **un ensemble de fonctions**, chacune accomplissant une tâche bien définie.
- Afin d'accomplir leurs tâches en quelques lignes (au plus 40), **les fonctions de niveau supérieur appellent des fonctions de niveau inférieur** et ainsi de suite.
- Le processus de décomposition d'une tâche complexe en fonctions plus petites est appelé **décomposition fonctionnelle**.

Fonctions principale et intermédiaires



} Fonction principale

} Fonctions intermédiaires

Lorsque vous commencez à écrire un programme à partir de zéro :

- **les seules fonctions disponibles** sont les fonctions / méthodes de bibliothèque intégrées de niveau inférieur (par exemple, toutes les méthodes des chaînes de caractères).
- Votre fonction du premier niveau s'appelle **fonction principale** : `main`.
- Vous devez inventer toutes les **fonctions intermédiaires** entre la fonction principale et la bibliothèque.

Invention et mise en place de fonction

Avant d'écrire le code : Avec le top-down design

Par exemple :

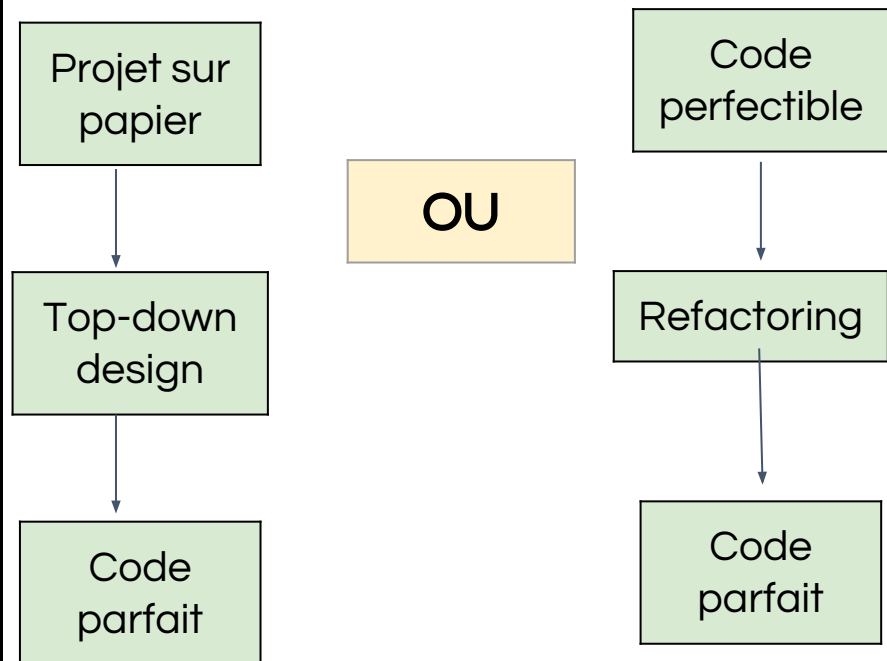
- on part avec l'idée de `lit_et_analyse_data`
- on le décompose en deux fonction
 - `lit_data`
 - `et_analyse_data`.
- On pourra le décomposer plus tard en
 - `prend_info_site`,
 - `rend_mesure_pluie`,
 - `calcule_moyenne`,
 - *etc*

Après avoir écrit le code : Pendant le 'refactoring'

En essayant de nettoyer le code

- Extraction (ou factorisation) de fonction
- Par la découverte :
 - Par exemple, une séquence particulière semble se répéter "Ah, ça me fait penser à une fonction..."

Dans l'idéal :



Dans la vraie vie

