

Accueil > Cours > Reprenez le contrôle à l'aide de Linux ! > Les fonctions

# Reprenez le contrôle à l'aide de Linux !

🕒 30 heures 📶 Facile

Mis à jour le 29/06/2021



## Le contenu de ce cours n'est plus à jour

Nous avons archivé ce cours et n'actualiserons plus son contenu.

Accédez au contenu le plus récent en découvrant ce cours :



### SYSTÈMES & RÉSEAUX

#### Initiez-vous à Linux

📶 Easy 🕒 8 heures

Dans ce cours débutant, découvrez Linux : un système d'exploitation gratuit et fascinant qui vous donnera un contrôle sans précédent sur votre ordinateur ! Créé par des passionnés d'informatique, Linux est un vecteur important de la philosophie du libre et l'alternative parfaite à Windows ou macOS.

VOIR LE NOUVEAU COURS

# Les fonctions

Dans ce chapitre, je vous propose de rendre notre programme plus agréable grâce **aux fonctions**. Et oui, en Bash aussi on peut faire des fonctions ! En fait, vous pouvez en faire dans quasiment tous les langages de programmation.

Commençons très simplement.

## Qu'est-ce qu'une fonction ?



Mais, c'est quoi une fonction ? 😞

Une fonction, c'est un ensemble d'instructions, permettant d'effectuer plusieurs tâches avec des **paramètres d'entrée différents**.

Son utilisation vous permettra de rendre votre programme plus lisible et structuré. Ainsi, il facilitera le développement de votre programme.



En plus, vous pouvez appeler votre fonction autant de fois que vous le souhaitez dans votre script.



Comment savoir si je peux faire une ou plusieurs fonction(s) dans mon code ? 💡

Pas si vite ! Avant de vous lancer dans la création de vos fonctions ou de votre script, vous devrez vous poser un maximum de questions.

Mais avant toute chose, il y a quelques points importants à connaître.



1. Vous n'êtes **pas limité** dans le nombre de fonction. Par contre, elles ne doivent pas avoir le même nom.
2. Vous pouvez **réutiliser** les variables globales initialisées dans votre script à l'intérieur ou à l'extérieur de votre fonction.
3. Vous pouvez déclarer des **variables locales** à vos fonctions.
4. Une bonne fonction, c'est une fonction qui traite d'un **élément spécifique et récurrent** dans votre script.

## Comment déclarer une fonction ?



En Bash, il y a deux manières pour déclarer une fonction :

```
# déclaration méthode 1
maFonction ()

{
  bloc d'instructions
}

#appel de ma fonction

maFonction
```

ou

```
# déclaration méthode 2

function maFonction

{
  bloc d'instructions
}

#appel de la fonction

maFonction
```

Vous voyez, ce n'est pas plus compliqué ! 😊

Maintenant, passons à la pratique et sauvegardez votre répertoire personnel sous Linux.

Posez-vous les bonnes questions...



Mais quelles questions dois-je me poser à part, comment sauvegarder mon répertoire personnel ?

Cherchez bien, il y en a plein !

Par exemple :

- Comment vais-je sauvegarder le répertoire ? (tar, cp , rysnc ...)

- Que faut-il sauvegarder dans mon répertoire personnel ? (tout ?, .bashrc, ....)
- Ou vais-je sauvegarder le répertoire ? (/Repertoiredebackup, sur mon autre serveur distant, dans le cloud...) ?
- Comment être averti et enregistrer un historique du bon ou du mauvais fonctionnement de mon script ? ( par MAIL, par fax ?, par téléphone ? dans un fichier, sur un site web, les deux ... ? )
- À quelle fréquence ?

Je suis sûr que vous en trouverez plein d'autres 😊

Lorsque vous ne vous posez plus de questions, c'est qu'il est temps d'ordonner toutes les informations.



Super ! Chaque question que je me pose correspond donc à une fonction ? 😊

Pas tout à fait ! Mais presque.

Vous pourriez par exemple avoir trois fonctions :

- une fonction **sauvegarde** ;
- une fonction **log** ;
- une fonction **avertissement**.



Tu as dit que l'on pouvait appeler notre fonction autant de fois qu'on le souhaite. Quel est l'intérêt dans ce script puisque les fonctions sont appelées une seule fois, NON ?

C'est vrai quand le programme est relativement court. Mais si vous souhaitez étoffer votre script et sauvegarder aussi le répertoire d'un membre de votre famille, vous pourrez rappeler chacune des fonctions **sauvegarde**, **log** et **avertissement** en passant les bons paramètres.



Je ne comprends toujours pas 😞, la fonction a été développée pour fonctionner avec le script qui sauvegarde mon répertoire personnel, NON ?

Justement NON ! En plus de rendre notre programme plus lisible, la fonction doit rester **le plus générique** possible, car le principe c'est aussi de réutiliser les fonctions sans coder un nouveau bloc d'instruction et faire des doublons.



Et oui, la magie de la fonction c'est que l'on peut lui passer des paramètres comme les paramètres que vous passez à votre script. 😊

Souvenez-vous **\$1 \$2 \$3 \$n** (\$0 pour le nom du script) et (**\$#** le nombre de paramètre passé au script) ...

## En résumé :

- Une fonction comprend plusieurs instructions.
- Vous pouvez l'appeler autant de fois que vous voulez dans votre code : laissez donc votre fonction assez générique et faites-la varier grâce aux paramètres que vous pouvez préciser en l'appelant.
- Vous pouvez déclarer une fonction Bash avec `maFonction ()` ou `function maFonction` .
- Vous faites ensuite appel à votre fonction `maFonction()` en tapant simplement dans votre code `maFonction` .

J'AI TERMINÉ CE CHAPITRE ET JE PASSE AU SUIVANT

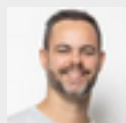


LES BOUCLES

TP : GÉNÉRATEUR DE GALERIE  
D'IMAGES



## Le professeur



**Mathieu Nebra**

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)

## Découvrez aussi ce cours en...



Livre



PDF

---

## OPENCLASSROOMS

Qui sommes-nous ?

Financements

Expérience de formation

Forum


Blog 

Presse 

---

## OPPORTUNITÉS

Nous rejoindre 

Devenir mentor 

Devenir coach carrière 

---

## AIDE



FAQ

---

## POUR LES ENTREPRISES

Former et recruter

---

## EN PLUS

Boutique 

Mentions légales

Conditions générales d'utilisation

Politique de protection des données personnelles

[Cookies](#)

[Accessibilité](#)

 Français ▼

---

