

# Queste de savoir

La variable CDPATH dans votre shell

---

4 février 2021



# Table des matières

Récemment j'ai découvert la variable shell `CDPATH`.

Usuellement lorsque l'on utilise la commande `cd` c'est pour changer de répertoire par rapport au répertoire courant.

Il existe bien des raccourcis pour se rendre rapidement dans son *home*, avec `cd ~` (équivalent à `cd $HOME`) ou encore pour se rendre rapidement dans le répertoire précédent avec `cd -` (équivalent à `cd $OLDDIR`).

La variable `CDPATH` permet d'étendre le champ de déplacement à d'autres répertoires.

Faisons un exemple.

Par défaut la variable `CDPATH` est vide.

```
1 $ echo $CDPATH
```

Imaginons que je sois un développeur Go et que j'utilise un *workspace* qui est `$HOME/prg/go` (`$GOPATH=$HOME/prg/go`). Les projets que je développe en Go seront dans le répertoire `$HOME/prg/go/src/github.com/keftcha`.

*i*

Il n'y a pas d'obligation de savoir ou de comprendre comment fonctionnent les *workspace* de Go. Si vous voulez en savoir plus je vous invite à lire [ceci](#) ↗ .

Mes projets se situent donc dans le répertoire `$HOME/prg/go/src/github.com/keftcha`. Ce chemin est plutôt long. Pour pouvoir m'y rendre rapidement je peux créer un alias de la sorte :

```
1 alias projects="cd $HOME/prg/go/src/github.com/keftcha"
```

Cependant faire un alias ne nous permet pas d'aller directement dans le répertoire d'un projet. Pour pallier ce problème on peut faire le choix d'utiliser une variable.

```
1 export $PROJECTS="$HOME/prg/go/src/github.com/keftcha"
```

Avec cette méthode, je peux me déplacer directement dans un projet.

```
1 $ cd $PROJECTS/super_projet
```

Cette méthode a l'inconvénient de devoir toujours commencer par le nom de notre variable. C'est maintenant que la variable `CDPATH` entre en scène!

Je définis la variable comme étant le répertoire de mes projets.

```
1 export CDPATH="~/prg/go/src/github.com/keftcha"
```

À présent, en étant dans mon répertoire *home*, je peux directement me rendre dans mon super projet.

```
1 $ pwd
2 /home/keftcha
3 $ cd super_projet
4 ~/prg/go/src/github.com/keftcha/super_projet
5 $ pwd
6 ~/prg/go/src/github.com/keftcha/super_projet
```

Comme on peut le voir, lorsque je souhaite changer de répertoire, la commande `cd` cherche dans un premier temps si le répertoire se trouve dans le répertoire courant **puis ensuite s'il n'est pas trouvé** dans les répertoires spécifiés par la variable `CDPATH`.



Tout comme la variable `PATH`, `CDPATH` peut avoir plusieurs répertoires.  
Ex : `CDPATH="~/prg/python:~/prg/go/src/github.com/keftcha"`

---

Le shell utilise le répertoire courant comme étant le répertoire de base pour tous les chemins relatifs. En utilisant la variable `CDPATH` nous pouvons utiliser d'autres répertoires comme étant une base relative.