

Les Avantages de la Répétition en Programmation

```
export PATH=/opt/homebrew/opt/ruby/bin:/opt/homebrew/opt/openjdk/bin:/opt/homebrew/bin:/opt/homebrew/sb
```

Dans ce cas, évitez d'utiliser la variable :

```
"$HOME/Library/Python/3.9/bin"
```

```
export PATH=/opt/homebrew/opt/ruby/bin:/opt/homebrew/opt/openjdk/bin:/opt/homebrew/bin:/opt/homebrew/sb
```

La deuxième version semble un peu répétitive avec le répertoire `/Users/lzwjava` apparaissant plusieurs fois, mais elle est beaucoup plus claire et plus facile à lire.

La répétition en programmation peut en réalité être bénéfique. Nous n'avons pas toujours besoin de tout abstraire ou simplifier avec des variables ou des fonctions. Dans ce cas, le répertoire personnel `/Users/lzwjava` a peu de chances de changer fréquemment, il est donc tout à fait acceptable de le répéter.

Cela ressemble à la façon dont nous faisons référence aux noms des personnes : si nous utilisons trop de pronoms comme "il", "elle" ou "ils", il devient difficile de savoir de qui nous parlons. Les noms eux-mêmes sont peu susceptibles de changer, et les utiliser directement peut apporter de la clarté.

L'abstraction et la simplification sont importantes, mais dans certains cas, elles introduisent une complexité inutile. Parfois, la répétition est plus simple et plus compréhensible.