

The Benefits of Repetition in Programming

```
export PATH=/opt/homebrew/opt/ruby/bin:/opt/homebrew/opt/openjdk/bin:/opt/homebrew/bin:/opt/homebrew/$HOME/Library/Python/3.9/bin:/Library/TeX/texbin:/Users/lzwjava/bin:/Users/lzwjava/platform-tools:/Users/lzwjava/Downloads/google-cloud-sdk/bin
```

In this case, avoid using the variable:

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

```
export PATH=/opt/homebrew/opt/ruby/bin:/opt/homebrew/opt/openjdk/bin:/opt/homebrew/bin:/opt/homebrew/$HOME/Library/Python/3.9/bin:/Library/TeX/texbin:/Users/lzwjava/bin:/Users/lzwjava/platform-tools:/Users/lzwjava/Downloads/google-cloud-sdk/bin
```

The second version seems a bit repetitive with the `/Users/lzwjava` directory appearing multiple times, but it's much cleaner and easier to read.

Repetition in programming can actually be beneficial. We don't always need to abstract or simplify everything with variables or functions. In this case, the home directory `/Users/lzwjava` is unlikely to change frequently, so it's perfectly acceptable to repeat it.

This is similar to how we refer to people's names: If we overuse pronouns like "he", "she", or "they," it becomes unclear who we're talking about. Names themselves are unlikely to change, and using them directly can add clarity.

Abstraction and simplification are important, but in some cases, they introduce unnecessary complexity. Sometimes, repetition is simpler and more understandable.