

# Bash

- [How to use correctly error-handling](#)
- [Bash profile](#)

# How to use correctly error-handling

```
set -euo pipefail
```

JULIA EVANS  
@b0rk

## errors

by default, bash will continue after errors

oh, was that an error? who cares, let's keep running!!!

uh is that really the best choice?

bash programmer

set -e stops the script on errors

```
set -e  
unzip file.zip
```

typo! script stops here!

this makes your scripts WAY more predictable

by default, unset variables don't error

```
rm -r "$HOME/$SOMEPTH"
```

\$SOMEPTH doesn't exist? no problem, i'll just use an empty string!

OH NOOOO

bash

set -u stops the script on unset variables

```
set -u  
echo $UNSET
```

oops!

I've never heard of that variable! STOP EVERYTHING!!!

bash

by default, a command failing doesn't fail the whole pipeline

```
curl 404.com | grep 'panda'
```

curl failed but grep succeeded so it's fine! success!

bash

set -o pipefail prevents this

all of these are why I like to put set -euo pipefail at the beginning of all my scripts!

# Bash profile

# Bash profile customization

<https://github.com/ohmybash/oh-my-bash>

Run this :

```
bash -c "$(curl -fsSL https://raw.githubusercontent.com/ohmybash/oh-my-bash/master/tools/install.sh)"
```