Overview

- Written by Brian Fox for the GNU Project in 1989
- BASH stands for Bourne Again Shell
- BASH is a shell scripting language, perfect for writing command line programs
- Huge amount of online support
- Used to easily automate complex series of commands for easy reuse

Features

- No explicit types
- Supports arrays: no size declaration required
- Redirect stdin and stdout to files
- Flexible parameter passing with functions
- Extensive string manipulation
  - `tr` command
- Tight integration with operating system
  - Commands executed on the command line can be executed in the shell script
- Variables global unless declared otherwise
- Read and write to sockets
- Process substitution
- Multifunctional test command
- Debugging: `#!/bin/bash`
- Can execute most Bourne shell scripts without modification
- Doesn’t support floating point math
- Only supports 1-D arrays

Cool Stuff

- Variables global unless declared otherwise
- Read and write to sockets
- Process substitution
- Multifunctional test command
- Debugging: `#!/bin/bash`
- Can execute most Bourne shell scripts without modification
- Doesn’t support floating point math
- Only supports 1-D arrays

Gotchas

- Use of whitespace in variable assignments
- Mixing up `–eq` and `=`
- Assuming uninitialized variables are zero

Code Examples

```
#!/bin/bash

echo -n "Which fibonacci number do you want to see? "
read serial

count=0
fibonacci_number=1

while [ $count -lt $serial ]
do
  fibonacci_number=`$fibonacci_number + $count`
  count=`$count + 1`

done

echo "Fibonacci number $serial = $fibonacci_number"
```

```
#!/bin/bash

if [ $file1 -nt $file2 ]
then
  echo "Have you been modified more recently than file2"
else
  echo "Has not been modified more recently than file2"
fi
```

```
diff <(find dir1) <(find dir2)
```

```
if [ $file1 -nt $file2 ]
then
  echo "Have you been modified more recently than file2"
else
  echo "Has not been modified more recently than file2"
fi
```