CIS241

System-Level Programming and Utilities

bash Arguments and Variables

Erik Fredericks, frederer@gvsu.edu

Fall 2025

> **Based on material provided by Erin Carrier, Austin Ferguson, and Katherine Bowers**

Arguments (positional parameters)

Modify behavior of program!

- **E.g., `cat file` --> `file` is an argument!**
- **Reference by `${n}` - `n` is the position**
  - What was `$0` again?
- **`$1`, `$2`, 1st and 2nd (respectively) arguments on command line**
  - Need braces for 10 or more: `${10}`

Arguments (more of)

- `$#` - total number of arguments
- `$@` - array of arguments
- `$*` - string with arguments separated by space

WHY?


To Be Continued

Variables

Assignment:

```
age=42
```

```
filename=erik.txt
```

```
message="This is a test"
```

SPACING IS IMPORTANT

```
echo "I am $age$ years old
```

Variables

You can use `$(cmd)` to grab its input

Combining with variables:

```
num_lines=$(grep "dog" file.txt | wc - l)
```

Single vs. double quotes

Yes it matters!

Single quotes: everything treated literally

Double quotes: expand variables inside

Try:

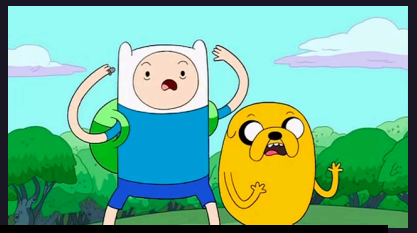
```
echo '$message'
echo "$message"
```

MORE

Variables can be accessed like `$var` or `${var}`

- Disambiguation!
  - `age=102` : `echo "I am $ageyears"` vs. `echo "I am ${age}years"`

Mathematical



What you expect won't work

Want to do 1 + 1?

- That's `$((1+1))`
- `$((num1+num2))`

Operators available:

- `+`, `-`, `/`, `*`, `%`, `++`, `--`, `**`

(Can go deeper with bracket expansion or the `expr` keyword but that way lies pain)

- https://www.namehero.com/blog/how-to-do-bash-math-and-when-to-use-it/