The Bash programming language can do very basic arithmetic, which we'll demonstrate in this section. Now that you have math.sh open in your preferred text editor type the following into your text editor:

```
#!/usr/bin/env bash
# File: math.sh

expr 5 + 2
expr 5 - 2
expr 5 \* 2
expr 5 / 2
```

Save math.sh and then run this script in your shell:

```
bash math.sh

## 7
## 3
## 10
## 2

```

Let's break down what's going on in the Bash script you just created. Bash executes programs in order from the fist line in your file to the last line. The expr command can be used to evaluate Bash expressions. An expression is just a valid string of Bash code that, when run, produces a result. The arithmetic operators that you're already familiar with for addition (+), subtraction (-), and multiplication (*) work like you would expect them to. Notice that when doing multiplication you need to escape the star character, otherwise Bash thinks you're trying to create a regular expression! The division operator (/) does not work as you might expect it to since 5 / 2 = 2.5. Bash does integer division, which means that the result of dividing one number by another is always rounded down to the nearest integer. Let's take a look at a few examples on the command line:

```
expr 1 / 3
expr 10 / 3
expr 40 / 21
expr 40 / 20

## 0
## 3
## 1
## 2

```

The other numerical operator you should be aware of that you might not be familiar with is the modulus operator (%). The modulus operator returns the remainder after integer division. In integer division if A / B = C, and A % B = D, then B * C + D = A. Let's take a look at some examples on the command line:

```
expr 1 % 3
expr 10 % 3
expr 40 % 21
expr 40 % 20

## 1
## 1
## 19
## 0

```

Notice that when one number is completely divisible by another number then the result of the modulus is zero.

If you want to do more complex math, for example math with fractions and numbers with decimals then I highly suggest combining echo and the **b**ench **c**alculator program called bc. Open up a new file called bigmath.sh and type in the following:

```
#!/usr/bin/env bash
# File: bigmath.sh

echo "22 / 7" | bc -l
echo "4.2 * 9.15" | bc -l
echo "(6.5 / 0.5) + (6 * 2.2)" | bc -l
```

Save bigmath.sh and then run this script in your shell:

```
bash bigmath.sh

## 3.14285714285714285714
## 38.430
## 26.2

```

You can pipe any mathematical string to bc with the -l flag in order to use decimal numbers in your calculations.

Summary

- Bash programs are executed in order from the first line in a file until the last line.
- Anything written after a pound sign (#) is a comment and is not executed by Bash.
- You can do simple arithmetic with the expr command.
- Perform more complicated arithmetic by piping a string expression into bc using echo.

Mark as completed