

LIS-5364

Programming concepts, web admin, and basic
networking with Bash

“Singular” Data types

(scalars, primitives)

- Booleans
- Strings
- Integers
- Floats

Booleans

True/False.

Often, 0/1

....but.

Null

Watch out for “Null” - depending on the language it might be 0, or it might be something else.

- consider how this can go wrong?

(weird metaphysical way to think about it - blind people don't see “black,” they see nothing.)

Strings

Text - “Hello world!”

Sometimes, numbers can still be treated as
“text”

Data “Types”: Bash

Bash basically only has two scalar/singulars:

“Text”

and

“Integers”

For multiples: Files and “delimited text”

(bash does have arrays)

Integers

Whole numbers, can be negative.

Usually pretty easy to deal with.

Note: some languages default to this, as in

$5 / 2 = 2$ - including Bash.

(for math in bash, use “bc”)

Floats (Floating Point Numbers)

Decimals.

Can be very tricky, be careful.

(Excel found a floating point error over 10 years later. Messed a lot of people up)

“Multiple” data types (holders/containers/lists)

Indexed list of things.

Color[0] = “red”

Color[1] = “orange”

etc.

Usually – ALL THE SAME DATA TYPE.

Some languages have different types of arrays. With funny names like “tuples.” or ones that kind of make sense (“dictionaries”)

Arrays

Usually – arrays of strings, or numbers, etc.

Depending on the language, these can be mixed.

Also, nested. As in, an array of arrays.

(and now you can do a grid.)

“Hello Worlds”

Way to get a tiny flavor of a language.

Variables

Indicator/Identifier of an instance of a data type.

```
greeting = "Hey everybody"
```

```
cost = "19.99"
```

```
numberarray = [0,4,23,29]
```

Loops/Conditionals

So, GOTO is bad; because if something goes wrong, there might not be a way to tell how you got there.

10

20 If explosion, then GOTO 50

30 do stuff

40 do some more stuff

50 Here we are? But how did we get here?

Functions

- If there isn't a command that does what you want? Make one up!
- just like "math" function, e.g. --x goes in, y goes out:

$$Y = 2x + 30.$$

Some languages are only functions:

(those who do not know lisp are doomed to repeat it)

Loops

So whenever there is some sort of repetition or decision, put it in a loop, of some kind.

If house is on fire

 get water

Else

 chill out

do next thing

Loops

These tend to be pretty good english words +
the condition you test for +
what to do

While

Do

If

Unless

For

“For” is the only one that's a little weird: but it's most often used as a count

For 1 to 100

Count

Variables

Indicator/Identifier of an instance of a data type.

```
greeting = "Hey everybody"
```

```
cost = "19.99"
```

```
numberarray = [0,4,23,29]
```