Working with Variables and Constants

Nigel Poulton
Author & Trainer

@nigelpoulton nigelpoulton.com

Overview

Declaring Variables

Variable Types & Type Conversion

Short Declaration

Values and Pointers

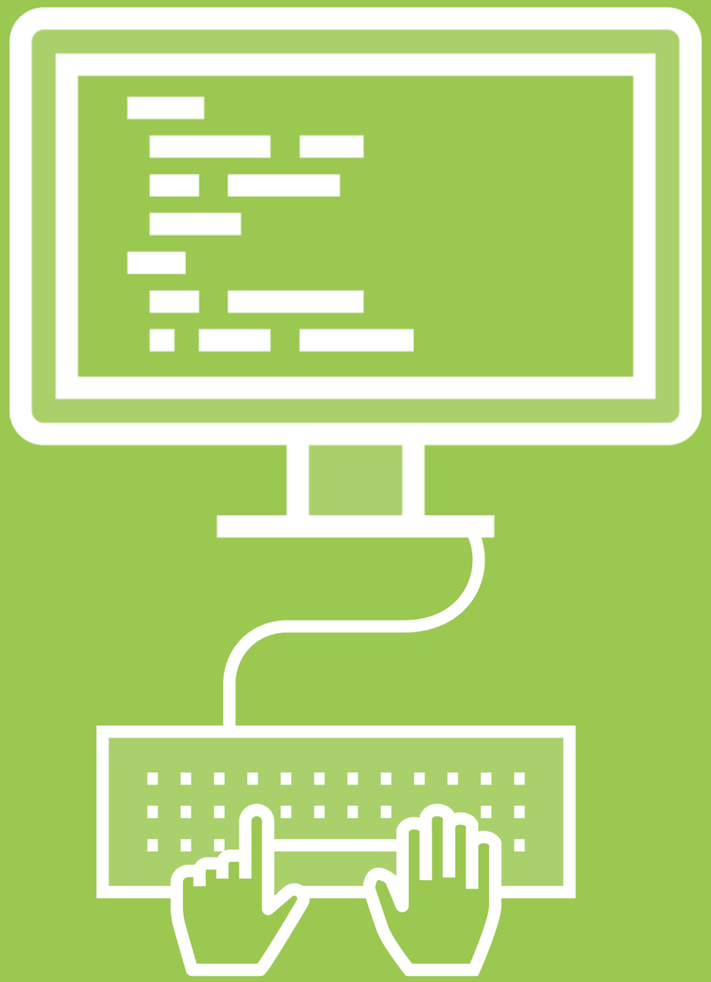
Passing by Value

Passing by Reference

Constants

Accessing Environment Variables

Recap



Sample App

For demo purposes only.
`github.com/nigelpoulton/ps-go/variables/`

Declaring Variables

```
break     default      func    interface   select
case      defer        go      map         struct
chan      else         goto    package     switch
const     fallthrough  if      range       type
continue  for          import  return      var
```

Variable names:

- Start with a letter
- Can’t be Go keyword
- Can’t contain spaces
- Can’t contain special characters

Good practice:

- Use meaningful names
- Keep names short
- Use camelCase

Getting Started with Kubernetes

0xaa

“course”

Getting Started with Kubernetes

0xbb

“course”

0xaa

“course”

0xbb

“copy”

Getting Started with Kubernetes

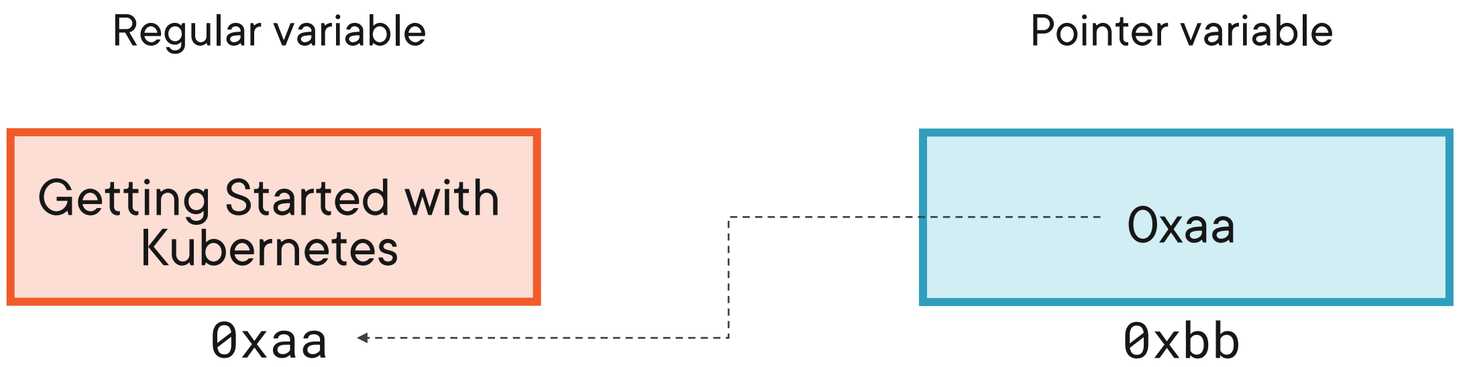
0xaa

“course”

Docker & Kubernetes The Big Picture

0xbb

“copy”


Regular variable
Pointer variable
Getting Started with Kubernetes
0xaa
0xaa
0xbb

Recap

Recap

At package level

`var <name> <type>`

At package level variables are *global*

Inside functions

`<name> := <value>`

Go supports *type inference*

Go passes variables *by value* (copy)

Pointers let you pass *by reference*

Constants are immutable variables

Go can access and manipulate environment variables

Up Next:
Working with Functions