Ruby | *for MacAdmins*

Horvitz Hall, Kenyon College

Nic Scott

Senior Apple Admin | Kenyon College

Professional Interest: ruby, scripting, automation

Personal Enjoyments: HBO, dry humor, IPA's

Slack: @nic.scott

Twitter: @niclscott

Github: https://github.com/nlscott

Email: scottnl@kenyon.edu

Blog: https://redlinetech.wordpress.com

Why Ruby?

Overview

1. Ruby Setup
2. Basic Ruby Concepts
3. Working Plist
4. Things Admins Can Do

Setup

Is Ruby installed?
What version do I have?
How do I run a script?

```
$ ruby -v
ruby 2.0.0p648
```

$which ruby

/usr/bin/ruby

```
$irb

irb(main):001:0>
```

```
irb(main):001:0> puts "Hello"
Hello
=> nil
```

```
#!/usr/bin/ruby

puts "Hello"
```

Save file to Desktop as hello.rb

Open Terminal, type:

ruby /path_to_hello.rb

or

chmod +x /path_to_hello.rb

```
[nscott@robot:~$ /Users/nscott/Desktop/hello.rb
 hello
```

Strings & Integers

a string in ruby is simply characters enclosed by quotes

a integer in ruby is whole number

```
> puts "Hello Mac Admins"

Hello Mac Admins
```

```
#convert all text to lowercase
puts "Hello Mac Admins”.downcase
```

```
hello mac admins
```

```
#convert all text to uppercase
puts "Hello Mac Admins”.upcase
```

```
HELLO MAC ADMINS
```

```
#concatenate strings with +
puts "Hello " + "Mac " + "Admins "
```

```
Hello Mac Admins
```

```
#anything inside quotes is a string
puts "3"
```

```
3
```

#.class will tell you what kind of object you have

```
puts "3".class
```

String

```
#list of methods for class
puts String.instance_methods
```

eql?
hash
casecmp
+
*
%
[]
[]=
insert
length
size
bytesize
empty?
=~
match
succ
succ!
next
next!
upto
index
rindex
replace
clear
chr
getbyte

setbyte
byteslice
to_i
to_f
to_s
to_str
inspect
dump
upcase
downcase
capitalize
swapcase
upcase!
downcase!
capitalize!
swapcase!
hex
oct
split
lines
bytes
chars
codepoints
reverse
reverse!
concat

<<
prepend
crypt
intern
to_sym
ord
include?
start_with?
end_with?
scan
ljust
rjust
center
sub
gsub
chop
chomp
strip
lstrip
rstrip
sub!
gsub!
chop!
chomp!
strip!
lstrip!

rstrip!
tr
tr_s
delete
squeeze
count
tr!
tr_s!
delete!
squeeze!
each_line
each_byte
each_char
each_code
point
sum
slice
slice!
partition
rpartition
encoding
force_enco
ding
b
valid_encod
ing?

ascii_only?
unpack
encode
encode!
to_r
to_c
>
>=
<
<=
between?
nil?
!~
class
singleton_cl
ass
clone
dup
taint
tainted?
untaint
untrust
untrusted?
trust
freeze
frozen?

methods
singleton_m
ethods
protected_
methods
private_met
hods
public_met
hods
instance_va
riables
instance_va
riable_get
instance_va
riable_set
instance_va
riable_defin
ed?
remove_inst
ance_varia
ble
instance_of
?
kind_of?
is_a?
tap

send
public_sen
d
respond_to
?
extend
display
method
public_met
hod
define_singl
eton_metho
d
object_id
to_enum
enum_for
equal?
!
!=
instance_ev
al
instance_ex
ec
__send__
__id__
=> nil

Ruby Strings & Integers

```
#integers do not need quotes
puts 3
```

```
3
```

```
puts 3 + 5
```

```
8
```

#.class will tell you what kind of object you have

```
puts 3.class
```

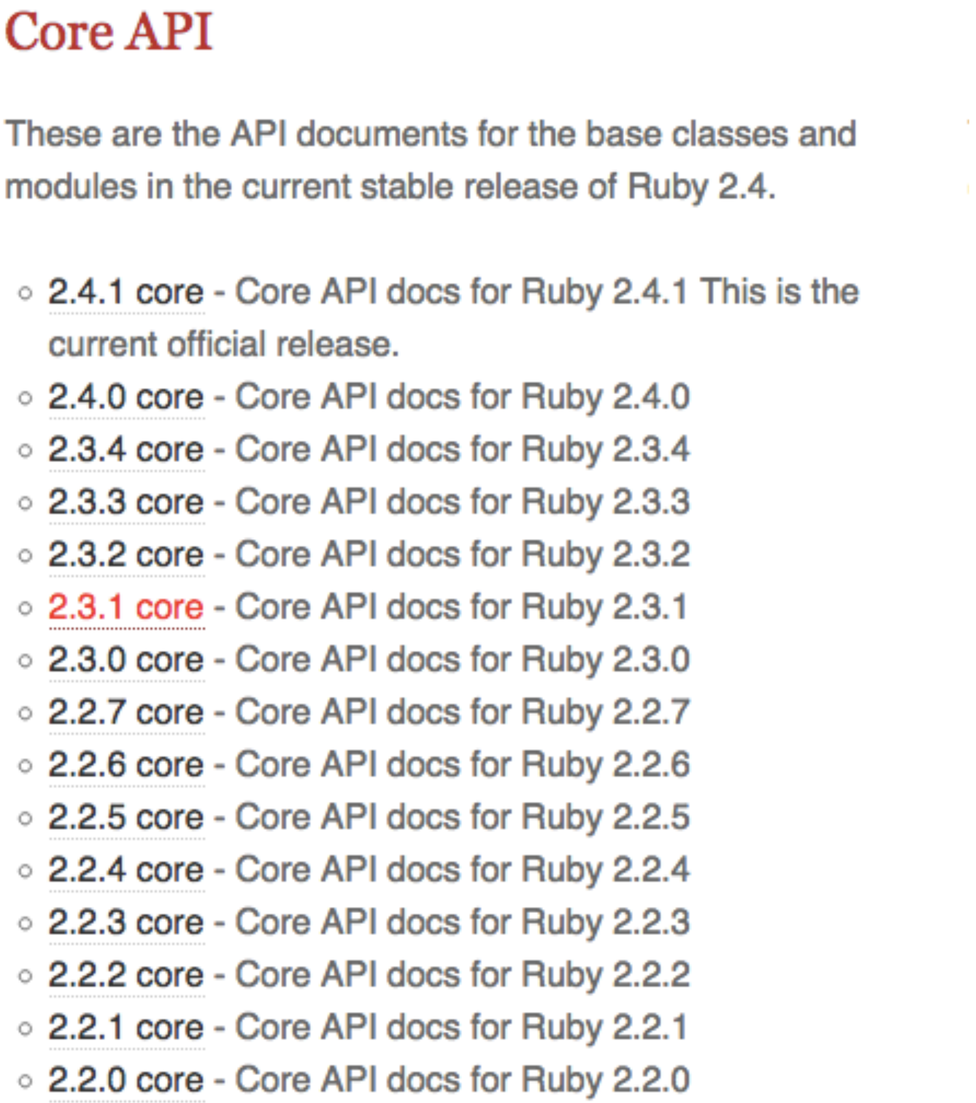
Fixnum

```
#list of methods for class
puts Fixnum.instance_methods
```

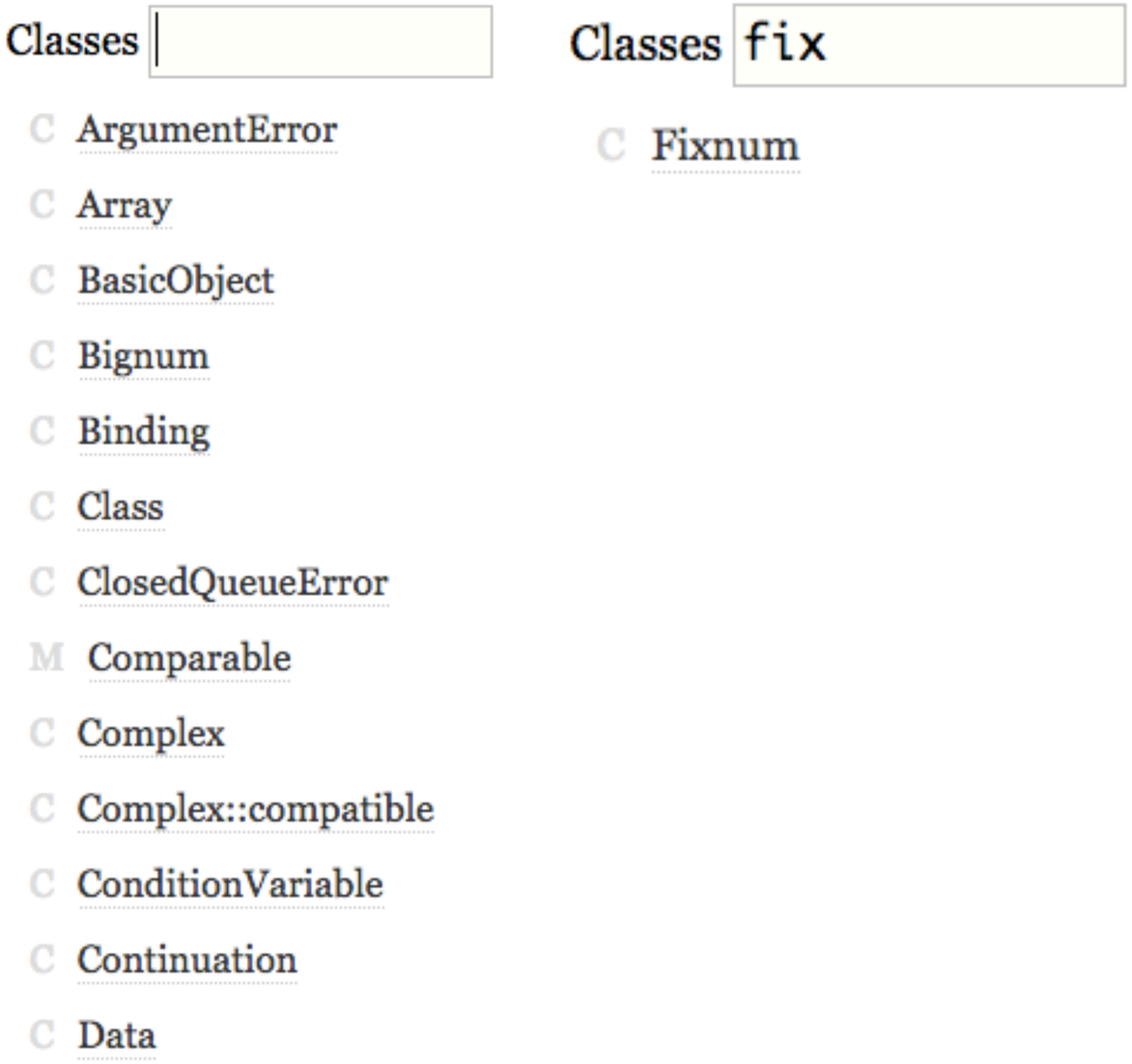
#to see examples

https://ruby-doc.org

- under "Core API", click on your version of ruby



- Scroll down to "Classes"
- manually find
- search box



Home Classes Methods

In Files

numeric.c

Parent

Integer

Methods

#%
#&
#*
#**
#+
#-
#-@
#/
#<
#<<
#<=
#<=>
#==
#===
#>

Fixnum

Holds Integer values that can be represented in a native machine word (minus 1 bit). If any operation on a Fixnum exceeds this range, the value is automatically converted to a Bignum.

Fixnum objects have immediate value. This means that when they are assigned or passed as parameters, the actual object is passed, rather than a reference to that object.

Assignment does not alias Fixnum objects. There is effectively only one Fixnum object instance for any given integer value, so, for example, you cannot add a singleton method to a Fixnum. Any attempt to add a singleton method to a Fixnum object will raise a TypeError.

Public Instance Methods

fix % other → real

Returns `fix` modulo `other`.

See Numeric#divmod for more information.

Variables

Variables are used to store information

Local Variables
Constants
Global Variables
Class Variables
Instance Variables

Local Variables:
should start with a lower case letter or an underscore

```
name = "Joe"
_name = "Bob"
```

Constants:
Should be used with all caps and meant for items that will not change

APPFOLDER = "/Applications"

Global Variables:

begins with '$' and has a global scope; meaning it can be accessed from anywhere within the program during runtime

$serial = "C02M50AGF6T6"

Instance Variables:

An instance variable has a name beginning with @, and its scope is confined to whatever object self refers to

@printer = "Ricoh 3005"

Class Variables:

A class variable is shared by all instances of a class and begins with '@@'

```
@@number_of_users = 20
```

Flow Control

Using conditional statements and loops

If Statement

```ruby
x = 3

if x == 3
  puts "x is 3"
end

```

```
x is 3
[Finished in 0.3s]
```

If/Else Statement

```ruby
x = 3

if x == 2
  puts "x is 2"
else
  puts "x is NOT 2"
end

```

```
x is NOT 2
[Finished in 0.3s]
```

Case Statement

```ruby

a = 5

case a
when 5
    puts "a is 5"
when 6
    puts "a is 6"
else
    puts "a is neither 5, nor 6"
end
```

```
a is 5
[Finished in 0.5s]
```

Until Loop

```ruby
x = 5
until x == 0
    x = x- 1
    puts "This loop runs #{x} more times"
end
```

```
This loop runs 4 more times
This loop runs 3 more times
This loop runs 2 more times
This loop runs 1 more times
This loop runs 0 more times
[Finished in 0.3s]
```

While Loop

```ruby
x = 5
while x > 0
    x = x- 1
    puts "This loop runs #{x} more times"
end
```

```
This loop runs 4 more times
This loop runs 3 more times
This loop runs 2 more times
This loop runs 1 more times
This loop runs 0 more times
[Finished in 0.1s]
```

For Loop

```
1
2 for x in 1..10
3     puts x
4 end
```

```
 1
 2 1
 3 2
 4 3
 5 4
 6 5
 7 6
 8 7
 9 8
10 9
11 10
```

Each

```
2 numbers = [1,2,3,4,5]
3
4 numbers.each do |x|
5     puts x
6 end
```

```
2 1
3 2
4 3
5 4
6 5
7 [Finished in 0.3s]
```

Arrays

Arrays are ordered, integer-indexed collections of any object

Print objects in array

```
1
2 fruit = ["apple", "orange", "banana"]
3
4 puts fruit
5
6
```

```
1
2 apple
3 orange
4 banana
5 [Finished in 0.1s]
```

Print objects in array by index

```
1
2 fruit = ["apple", "orange", "banana"]
3
4 puts fruit[0]
5
```

```
1
2 apple
3 [Finished in 0.3s]
```

Print objects index location

```
1
2 fruit = ["apple", "orange", "banana"]
3
4 puts fruit.index("banana")
5
6
```

```
1
2 2
3 [Finished in 0.3s]
```

Add an object to an array

```
1
2 fruit = ["apple", "orange", "banana"]
3
4 fruit.push("kiwi")
5
6 puts fruit
```

```
1
2 apple
3 orange
4 banana
5 kiwi
6 [Finished in 0.3s]
```

Delete an object from an array

```
1
2 fruit = ["apple", "orange", "banana", "kiwi"]
3
4 fruit.delete("kiwi")
5
6 puts fruit
```

```
1
2 apple
3 orange
4 banana
5 [Finished in 0.4s]
```

Print array in alphabetical order

```
1
2 fruit = ["apple", "orange", "banana"]
3
4 puts fruit.sort
5
```

```
1
2 apple
3 banana
4 orange
5 [Finished in 0.3s]
```

Print number of objects in an array

```ruby

fruit = ["apple", "orange", "banana"]

puts fruit.length

```

```
3
[Finished in 0.3s]
```

Print only unique items in an array

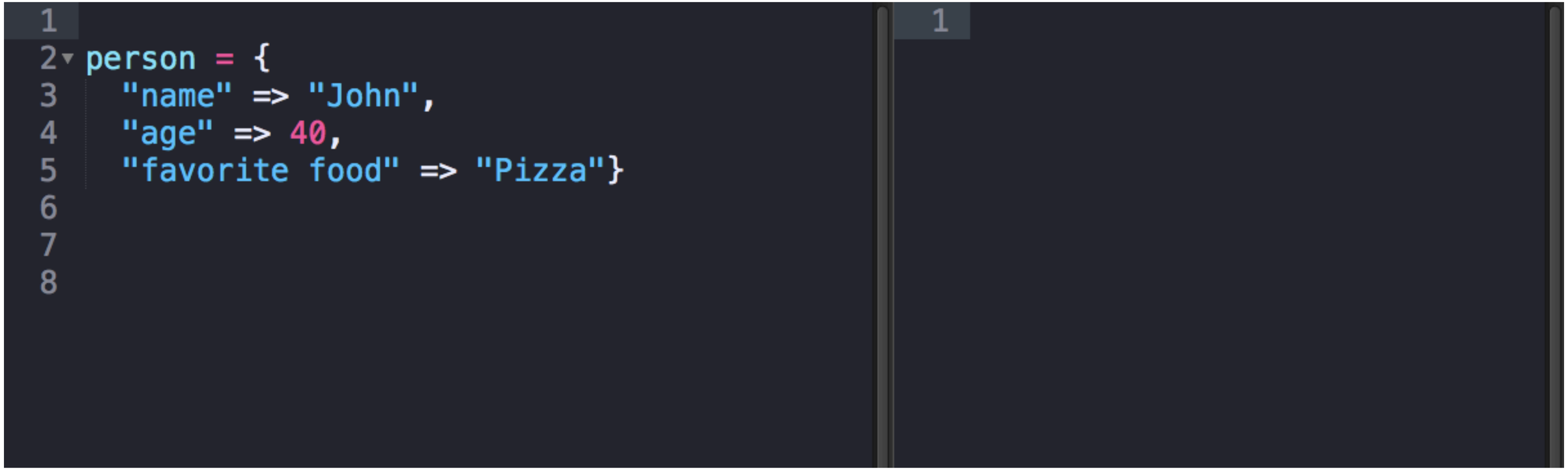
```
1
2 fruit = [
3     "apple", "orange",
4     "banana", "apple"]
5
6 puts fruit.uniq
7
```

```
1
2 apple
3 orange
4 banana
5 [Finished in 0.1s]
```

Hashes

A Hash is a collection of key-value pairs

Hash example



Puppet example

```
1
2  user { 'test':
3    ensure     => 'present',
4    comment    => 'test',
5    gid        => '20',
6    groups     => ['admin'],
7    home       => '/Users/test',
8    iterations => '40000',
9    shell      => '/bin/bash',
10   uid        => '502',
11 }
```

```
1
```

Hash

Print hash key and values

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

oses.each do |key, value|
    puts key + ": " + value
end


```

```
10.12: Sierra
10.11: El Capitan
10.10: Yosemite
10.9: Mavericks
[Finished in 0.3s]
```

Print hash key and values

```ruby
oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

oses.each do | puppy_dogs, icecream_cones|
    puts puppy_dogs + ":" + icecream_cones
end
```

```
10.12:Sierra
10.11:El Capitan
10.10:Yosemite
10.9:Mavericks
[Finished in 0.3s]
```

Check if a hash contains a specific key

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

puts oses.has_key? "Mountian Lion"
```

```
false
[Finished in 0.3s]
```

Check if a hash contains a specific key

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

puts oses.has_key? "10.9"
```

```
true
[Finished in 0.3s]
```

Print value for a key

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

puts oses["10.9"]


```

```
Mavericks
[Finished in 0.3s]
```

Add a key, value pair to a hash

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

oses["10.7"] = "Lion"

puts oses
```

```
{"10.12"=>"Sierra",
"10.11"=>"El Capitan",
"10.10"=>"Yosemite",
"10.9"=>"Mavericks",
"10.7"=>"Lion"}
[Finished in 0.1s]
```

Delete a key, value pair to a hash

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks",
    "10.7" => "Lion"}

oses.delete("10.7")

puts oses
```

```
{"10.12"=>"Sierra",
"10.11"=>"El Capitan",
"10.10"=>"Yosemite",
"10.9"=>"Mavericks"}
[Finished in 0.1s]
```

Hash methods

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

puts oses.empty?
puts oses.length
puts oses.keys
puts oses.values

```

```
false
4
10.12
10.11
10.10
10.9
Sierra
El Capitan
Yosemite
Mavericks
[Finished in 0.3s]
```

Hash methods

```ruby

oses ={
    "10.12" => "Sierra",
    "10.11" => "El Capitan",
    "10.10" => "Yosemite",
    "10.9" => "Mavericks"}

puts oses.class
puts Hash.instance_methods

```

```
Hash
<
>
<=
>=
==
[]
[]=
empty?
eql?
inspect
length
```

Methods

methods are used to bundle one or more repeatable statements

methods are very similar to functions in any other languages

should begin with a lowercase letter

should be defined before calling them

Method format

```ruby

def method_name
    code to execute…..
end


```

Method example

```ruby

puts "Hello MacAdmins"

def psumac
    puts "Hello MacAdmins"
end

psumac
```

```
Hello MacAdmins
Hello MacAdmins
[Finished in 0.1s]
```

Method example

```
1
2 def add
3     puts 3 + 3
4 end
5
6 add
7
```

```
1
2 6
3 [Finished in 0.2s]
```

Method arguments

```ruby

def double(x)
    puts x * 2
end

double(5)
```

```
10
[Finished in 0.3s]
```

Loops or iterators inside methods

```ruby

def list_of_apps
    Dir.entries("/Applications").each do |app|
        if !app.start_with?(".")
            puts app
        end
    end
end

list_of_apps
```

```
Adapter.app
Adobe InDesign CC 2015
Adobe Photoshop CC 2015.5
Atom.app
Audacity.app
BitBar.app
BlueGriffon.app
BlueStacks.app
Boom 2.app.zip
Brackets.app
Calculator.app

```

Working With Plist

CFPropertyList
a module to read, write, and manipulate both binary and XML property lists as defined by apple.

https://rubygems.org/gems/CFPropertyList

Checking for CFPropertyList

#open terminal and type

gem list

CFPropertyList (2.3.5)

Install CFPropertyList

```
#to install
sudo gem install CFPropertyList

#to update
sudo gem update CFPropertyList
```

Reading a plist with CFPropertyList

```ruby
require 'cfpropertylist'

# loads plist into an hash named data
plistfile = "/Library/Preferences/com.apple.loginwindow.plist"
plist = CFPropertyList::List.new(:file => "#{plistfile}")
data = CFPropertyList.native_types(plist.value)

#print  vaule
puts data['LoginwindowText']
```

```
For support contact Helpline x5700
[Finished in 0.2s]
```

#same as

defaults read "/Library/Preferences/com.apple.loginwindow.plist" LoginwindowText

Shelling out in Ruby

```
`defaults read "/Library/Preferences/com.apple.loginwindow.plist" LoginwindowText`
```

```
cmd = 'defaults read "/Library/Preferences/com.apple.loginwindow.plist" LoginwindowText'
system(cmd)
```

Modify a plist with CFPropertyList

```
1
2  require 'cfpropertylist'
3
4  # loads plist into an hash named data
5  plistfile = "/Library/Preferences/com.apple.loginwindow.plist"
6  plist = CFPropertyList::List.new(:file => "#{plistfile}")
7  data = CFPropertyList.native_types(plist.value)
8
9  # modify hash value
10 data['LoginwindowText'] = "Call the Helpdesk for support"
11
12 # save data back to file
13 plist.value = CFPropertyList.guess(data)
14 plist.save("#{plistfile}", CFPropertyList::List::FORMAT_BINARY
```

#same as

sudo defaults write /Library/Preferences/com.apple.loginwindow.plist LoginwindowText "Call the Helpdesk for support"

What Can MacAdmins Do With Ruby?

List users on machine

```ruby
#!/usr/bin/ruby

def list_of_users
    users=[]
    for username in Dir.entries("/Users")
        if !username.start_with?(".")
            users.push(username)
        end
    end
    return users
end

puts "Total Users: #{list_of_users.length}"
puts list_of_users

```

```
Total Users: 5
faculty
jsmith
nscott
Shared
student
```

List users in local admin group

```ruby

#!/usr/bin/ruby

def local_admins
    list_of_users_tocheck = []
    for x in Dir.entries("/Users")
        if !x.start_with?(".")
            list_of_users_tocheck.push(x)
        end
    end

    localadmins = []
    for x in list_of_users_tocheck
        if `dsmemberutil checkmembership -U #{x} -G admin 2>/dev/null`.chomp == "user is a member of the group"
        localadmins.push(x)
        end
    end
    puts "Local Admins:"
    return localadmins
end


def listoflocaladmins
    listoflocaladmins =`dscacheutil -q group -a name admin`.split(":")
    listoflocaladmins=listoflocaladmins[4].strip.split(" ")
    puts "Local Admins:"
    return listoflocaladmins
end

puts listoflocaladmins
```

```
Local Admins:
root
nscott
faculty

```

List of apps user installed after deployment

```ruby
#!/usr/bin/ruby

base_apps = [ "App Store.app", "Automator.app", "Calculator.app",
"Calendar.app", "Chess.app", "Contacts.app", "Dashboard.app",
"Dictionary.app", "DVD Player.app", "FaceTime.app", "Font Book.app",
"Game Center.app", "Google Chrome.app", "iBooks.app", "Image Capture.app",
"iTunes.app", "Launchpad.app", "Mail.app", "Maps.app", "Messages.app",
"Mission Control.app", "Notes.app", "Photo Booth.app", "Photos.app",
"Preview.app", "QuickTime Player.app", "Reminders.app", "Safari.app",
"Stickies.app", "System Preferences.app", "TextEdit.app", "Time Machine.app",
"Utilities", "VLC.app" ]

users_apps = []

#=============================================================================
#add apps installed by user to user_apps

Dir.entries("/Applications").each do | appname |
    if !appname.start_with?(".")
        if !base_apps.include?(appname)
            users_apps.push(appname)
        end
    end
end

puts "Number of Apps Installed: #{users_apps.length}"
puts users_apps

```

```
Number of Apps Installed: 52
Adapter.app
Atom.app
Audacity.app
BitBar.app
BlueGriffon.app
BlueStacks.app
Boom 2.app.zip
Camtasia 2.app
Dash.app
Dropbox.app
Firefox.app
GarageBand.app
GeekTool.app
GitHub Desktop.app
Google Drive.app
Keynote.app
LastPass.app
Microsoft Excel.app
Microsoft OneNote.app
Microsoft Word.app
MusicManager.app
Numbers.app
OverSight.app
Packages.app
Pages.app
Platypus.app
Skype.app
Slack.app
SourceTree.app
Sublime Text.app
Suspicious Package.app

```

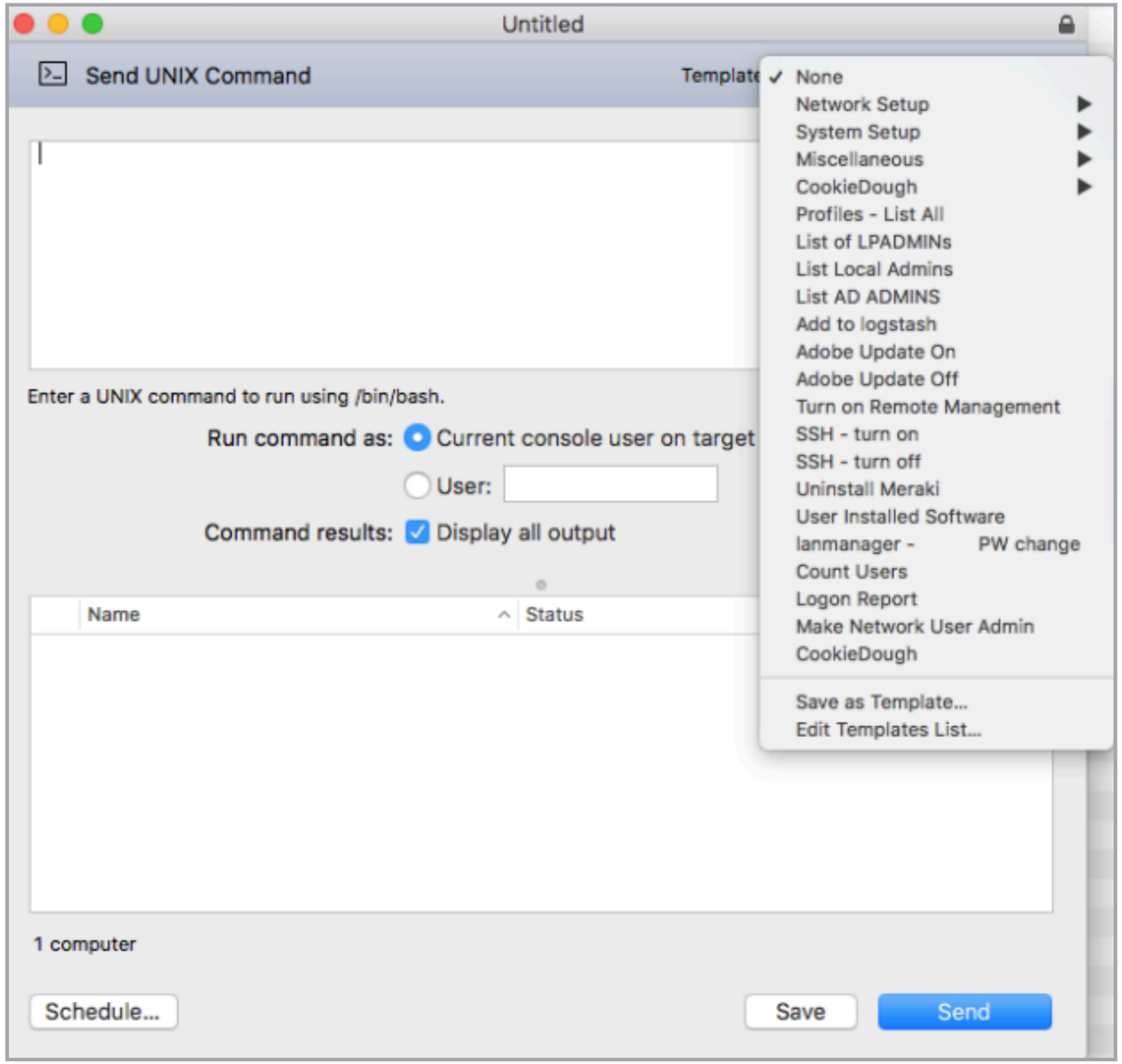
List apps in a users dock

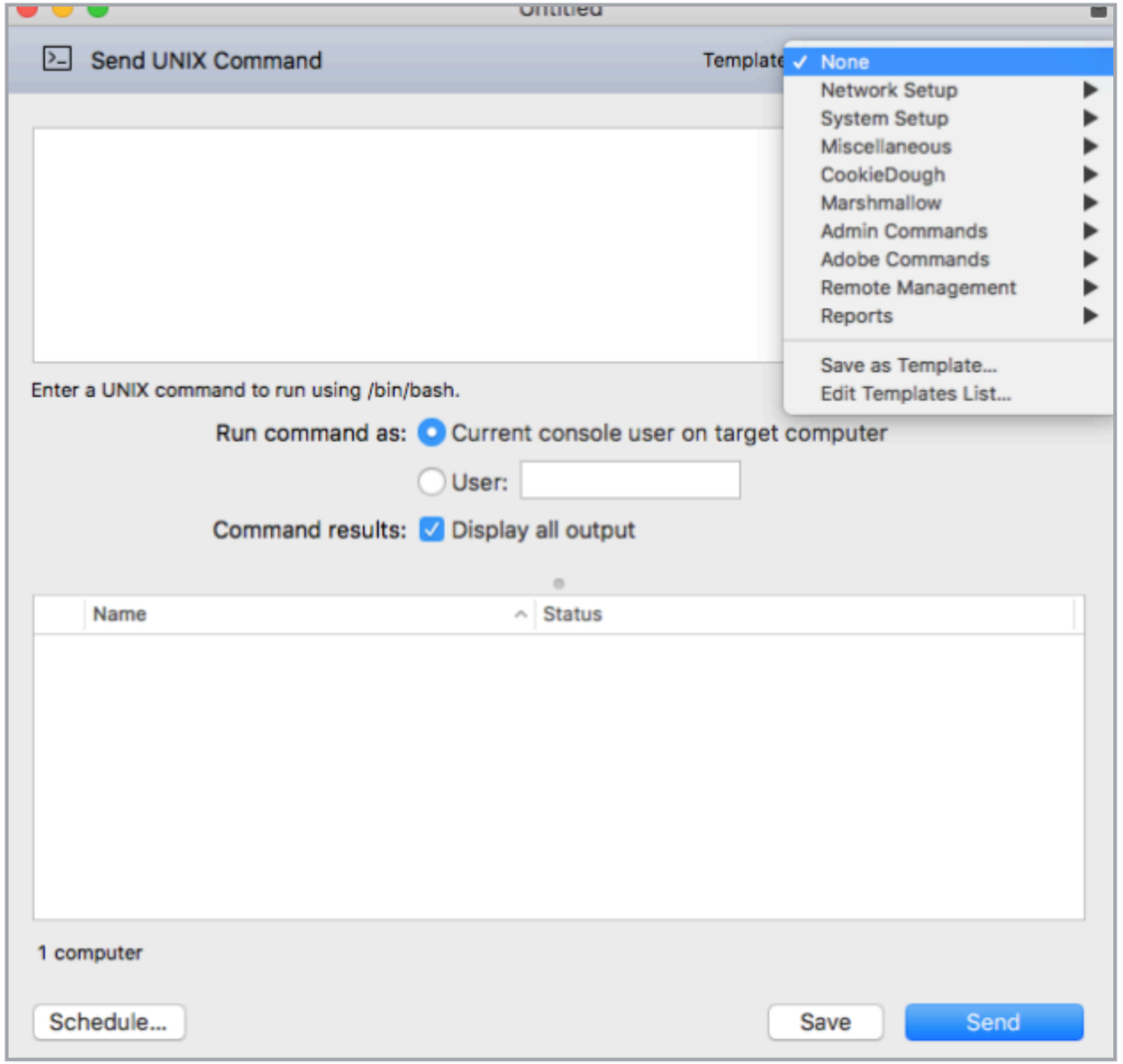
```ruby
#!/usr/bin/ruby

require 'CFPropertyList'
require 'etc'

CURRENTUSER=Etc.getlogin

#==========================================================================
#list the aspps in a users dock
def dock_apps(user = CURRENTUSER)

    plist = CFPropertyList::List.new(:file => "/Users/#{user}/Library/Preferences/com.apple.dock.plist")
    results=CFPropertyList.native_types(plist.value)
    apps=[]
    for key, value in results['persistent-apps']
        for key, value in key
            if value.class == Hash
                for x, y in value
                    if x == "file-label"
                        apps.push(y)
                    end
                end
            end
        end
    end
    return apps
end

#==========================================================================
#default is current users dock
puts dock_apps
```

```
Messages
Firefox
Google Chrome
Safari
Notes
Reminders
Sublime Text
Atom
Dash
BitBar
Slack
SourceTree
GitHub Desktop

[Finished in 0.5s]
```

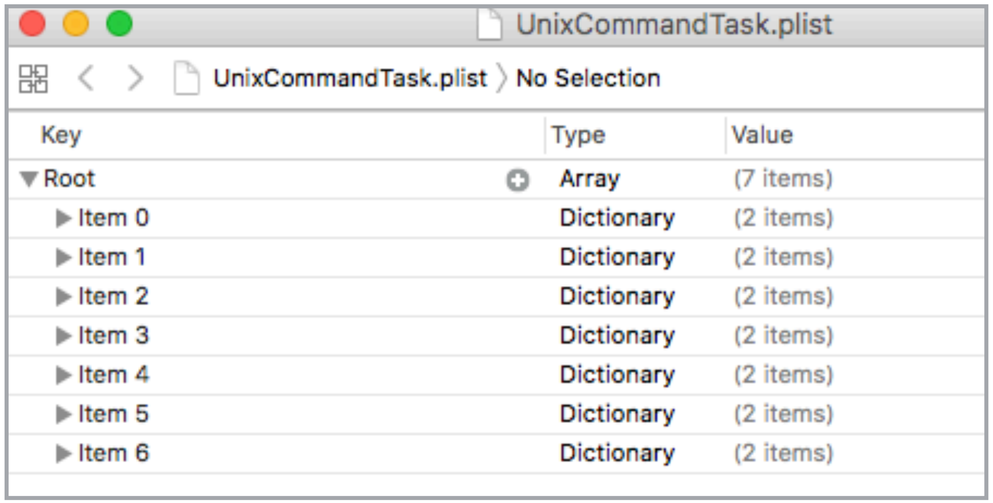
```ruby
#!/usr/bin/ruby

DOCKUTIL="/usr/local/bin/dockutil"
APPS="/Applications"

@add_app=["Launchpad.app","App Store.app", "Safari.app", "Google Chrome.app", "Google Drive.app", "Mail.app", "Contacts.app",
    "Calendar.app", "Notes.app", "Reminders.app", "Maps.app", "Photos.app", "Messages.app", "FaceTime.app",
    "iTunes.app", "Microsoft Excel.app", "Microsoft PowerPoint.app", "Microsoft Word.app", "blender.app",
    "SketchUp 2017/SketchUp.app", "Adobe Photoshop CC 2017/Adobe Photoshop CC 2017.app"]

@folder='~/Downloads'

def check_for_dockutil
    if !File.exist?("/usr/local/bin/dockutil")
        abort "Dockutil is not installed"
    end
end

def set_user_dock
    removeallapps = "#{DOCKUTIL} --remove all --allhomes --no-restart"
    system(removeallapps)

    @add_app.each do |appname|
        addapp="#{DOCKUTIL} --add '#{APPS}/#{appname}' --allhomes --no-restart"
        system(addapp)
    end

    if !@folder.empty?
        restart="#{DOCKUTIL} --add '#{@folder}' --view grid --display folder --allhomes --no-restart"
        system(restart)
    end
end

def set_user_template
    removeallapps = "#{DOCKUTIL} --remove all --no-restart '/System/Library/User Template/English.lproj'"
    system(removeallapps)

    @add_app.each do |appname|
        addapp="#{DOCKUTIL} --add '#{APPS}/#{appname}' --no-restart '/System/Library/User Template/English.lproj'"
        system(addapp)
    end
end

#run commands
check_for_dockutil
set_user_dock
set_user_template

```

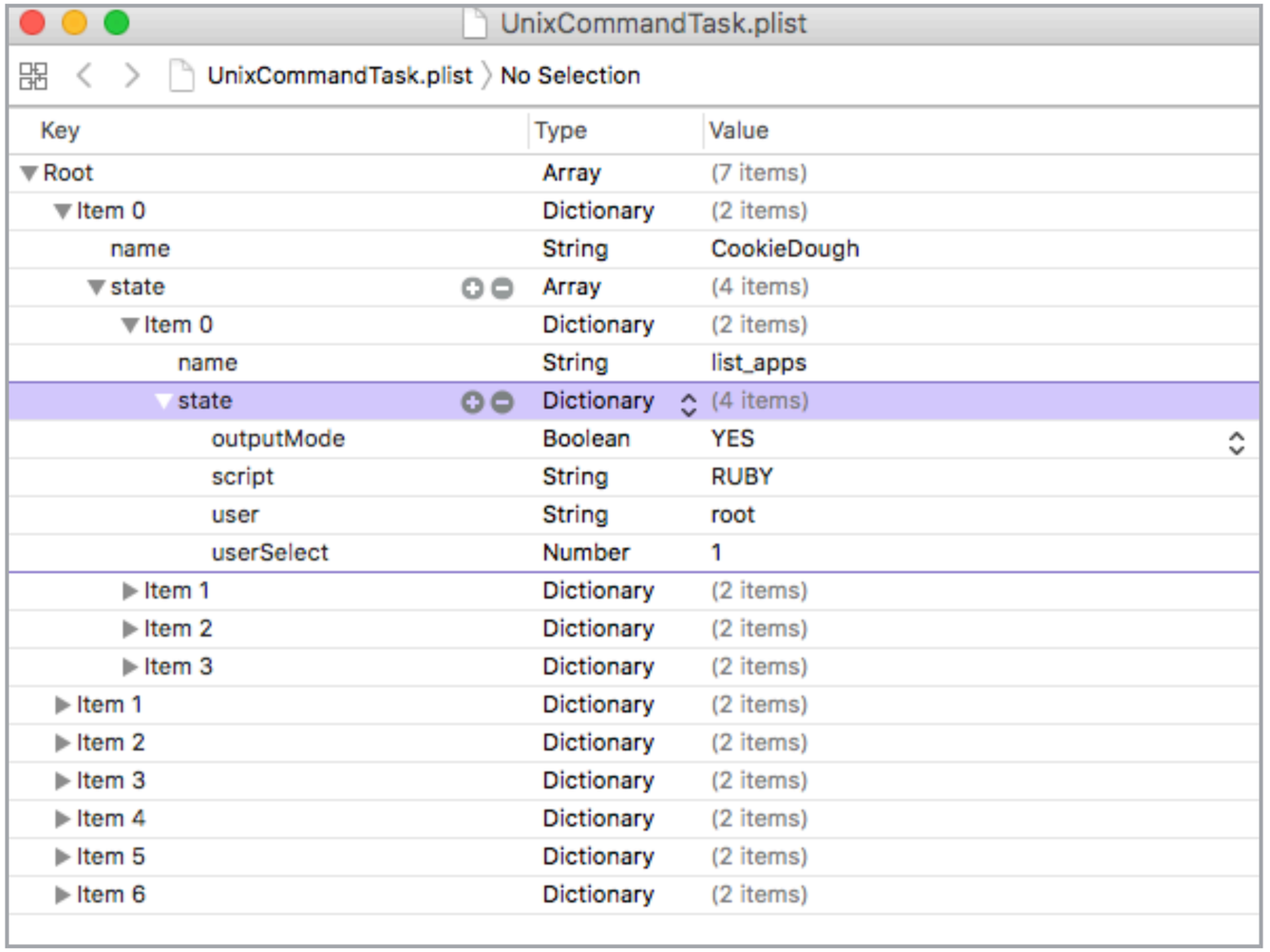
Organizing Unix Commands in ARD





Organizing Unix Commands in ARD

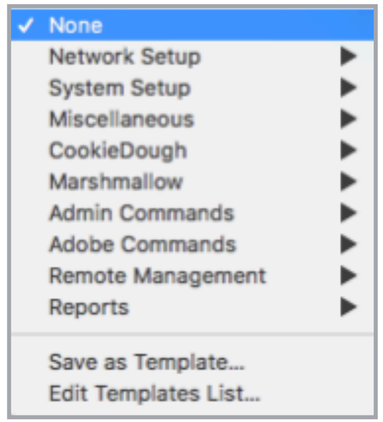




~/Library/Containers/com.apple.RemoteDesktop/Data/Library/Application\ Support/Remote\ Desktop/Presets/UnixCommandTask.plist

Organizing Unix Commands in ARD

```ruby
class Command
    attr_reader :foldername, :commandname, :outputmode, :script, :user, :userselect

    def initialize(foldername, commandname, outputmode, script, user, userselect)
        @foldername =   foldername
        @commandname =  commandname
        @outputmode = outputmode
        @script = script
        @user = user
        @userselect = userselect
    end
end

#=============================================================
# Methods

#print folder names
def self.list_folders
    folders=[]
    for x in @results
         folders.push(x['name'])
    end
    return folders
end

#find index of folder
def self.folder_index(folder)
    indexnumber=nil
    for x in @results
        if x['name'] == folder
            indexnumber = @results.index(x)
        end
    end
    if indexnumber.nil?
        return "Folder does not exist"
    else
        return indexnumber
    end
end
```



ARD.create_empty_folder("folder1")

ARD.move_command("Test Command", "Folder 1")

Ruby Gems

marshmallow 0.1.3

A collection of macOS facts and useful information for MacAdmins

cookiedough 0.1.1

A Ruby gem for macOS machines to report on Applications

panes 0.1.2

A gem to manage System Preference Panes. Enable or disable access to individual panes.

ard 0.1.0

A gem to create and organize commands in Apple Remote Desktop

```
1
2  require "Marshmallow"
3
4  puts Marshmallow.system_launchagents
```

```
1
2  com.adobe.AdobeCreativeCloud.plist
3  com.google.keystone.agent.plist
4  com.oracle.java.Java-Updater.plist
5  [Finished in 0.4s]
```

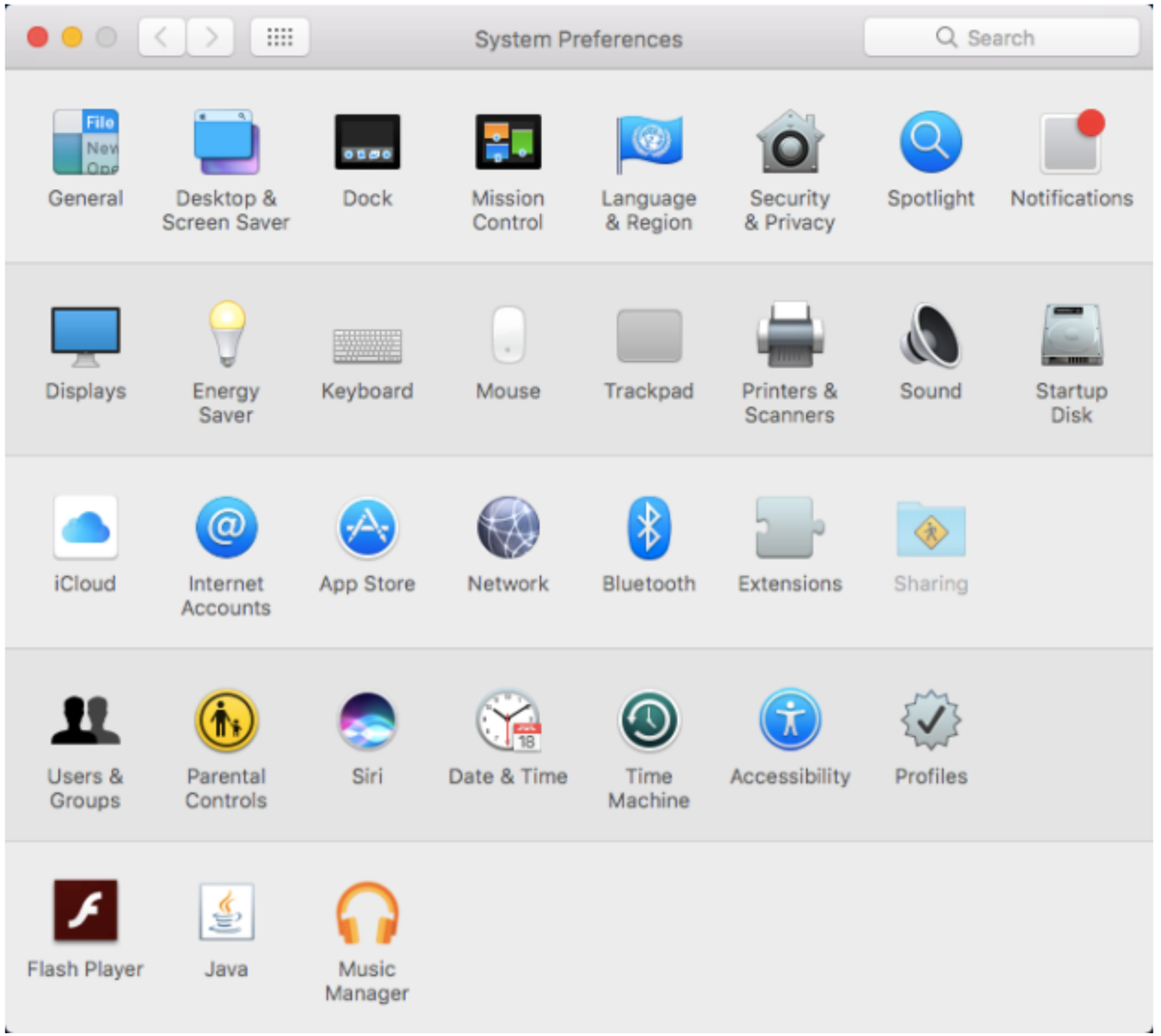
```
1
2  require "CookieDough"
3
4  puts CookieDough.dock_apps("nscott")
5
6
```

```
1
2  Messages
3  Firefox
4  Google Chrome
5  Safari
6  iTunes
7  Notes
8  Reminders
9  Sublime Text
10 Dash
11 BitBar
12 Slack
13 Atom
14 SourceTree
15 GitHub Desktop
16 Skype
17 [Finished in 0.2s]
```

```ruby

require "CookieDough"

puts CookieDough.all_app_versions

```

```
...
Microsoft OneNote.app = 15.33
Mission Control.app = 1.2
Monitor.app = 1.0.6
MusicManager.app = 1.0
NoMAD.app = 1.0.3
Notes.app = 4.3.1
Numbers.app = 4.1.1
OverSight.app = 1.0.0
Pages.app = 6.1.1
Photo Booth.app = 8.0
Photos.app = 2.0
Platypus.app = 5.1
Postgres.app = 2.0.1
Preview.app = 9.0
QuickTime Player.app = 10.4
Reminders.app = 4.0
...
```

Disable System Preference Panes



Panes.Disable("com.apple.preferences.sharing")

What Else?

Finder

Printers

DeployStudio Reporting

collecting or filtering data

References & Resources

Alleviate the Apprehension of Coding in Ruby: https://goo.gl/B4SQ4t

Ruby Docs: http://ruby-doc.org/

#ruby in Macadmins Slack

CodeCademy: https://www.codecademy.com/

LaunchSchool: https://launchschool.com/books/ruby/read

Ruby Gems: https://rubygems.org/

References & Resources

REDline: https://redlinetech.wordpress.com

GEMS: https://rubygems.org/profiles/nic_scott

GitHub: https://github.com/nlscott

Learn Ruby. Do Something Cool.