



# The Unix Shell

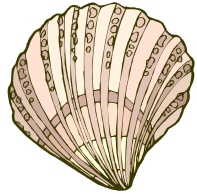
## Variables



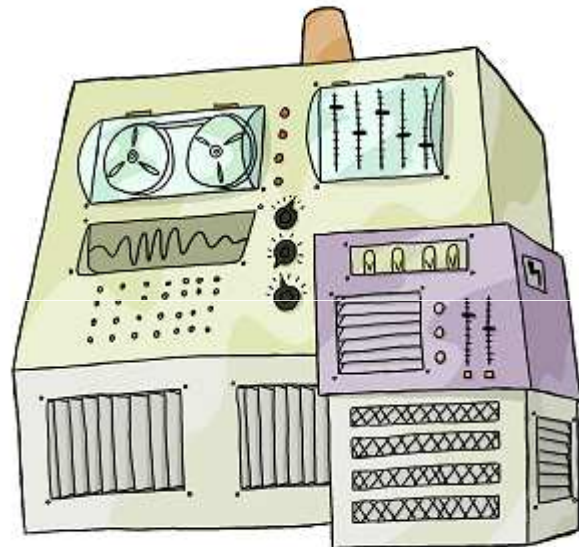
Copyright © Software Carpentry 2010

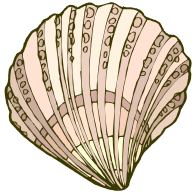
This work is licensed under the Creative Commons Attribution License

See <http://software-carpentry.org/license.html> for more information.



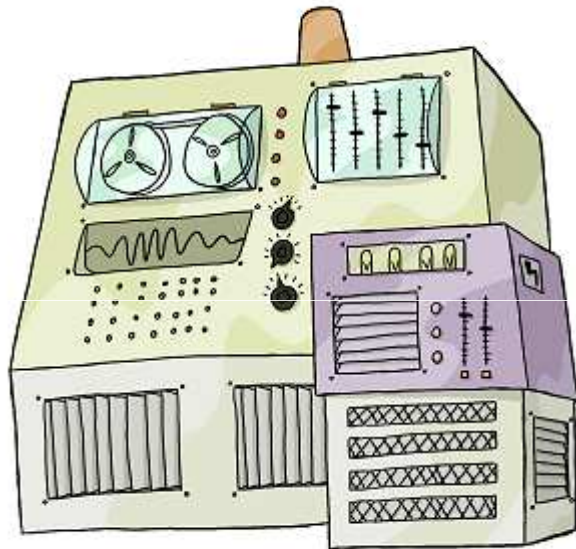
shell

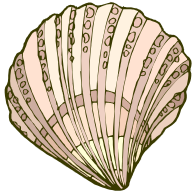




shell

The shell is a program

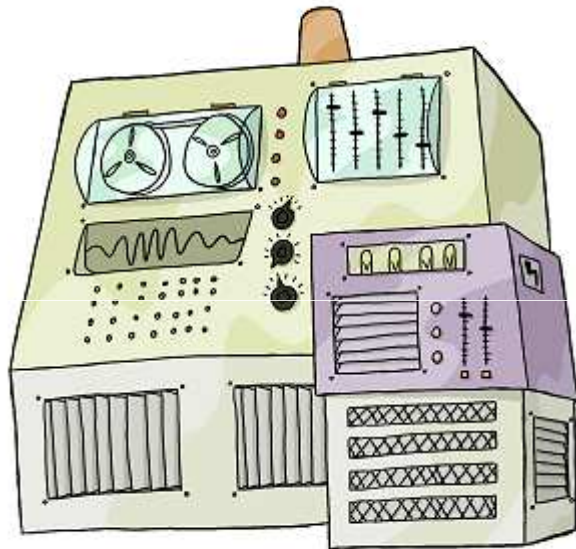


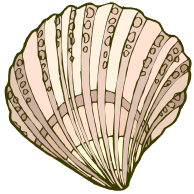


shell

The shell is a program

It has variables





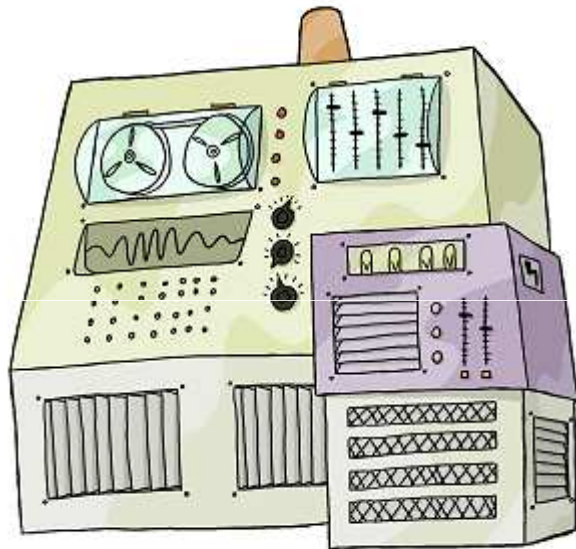
shell

The shell is a program

It has variables

Changing their values

changes its behavior



```
$ set
```

```
COMPUTERNAME=TURING
```

```
HOME=/home/vlad
```

```
HOMEDRIVE=C:
```

```
HOSTNAME=TURING
```

```
HOSTTYPE=i686
```

```
MANPATH=/usr/local/man:/usr/share/man:/usr/man
```

```
NUMBER_OF_PROCESSORS=4
```

```
OS=Windows_NT
```

```
PATH=/usr/local/bin:/usr/bin:/bin:/cygdrive/c/Windows/system32:
```

```
/cygdrive/c/Windows:/cygdrive/c/bin:/cygdrive/c/Python27
```

```
PWD=/home/vlad
```

```
UID=1000
```

```
USERNAME=vlad
```

\$ **set**



With no arguments, shows all variables and their values

*COMPUTERNAME=TURING*

*HOME=/home/vlad*

*HOMEDRIVE=C:*

*HOSTNAME=TURING*

*HOSTTYPE=i686*

*MANPATH=/usr/local/man:/usr/share/man:/usr/man*

*NUMBER\_OF\_PROCESSORS=4*

*OS=Windows\_NT*

*PATH=/usr/local/bin:/usr/bin:/bin:/cygdrive/c/Windows/system32:*

*/cygdrive/c/Windows:/cygdrive/c/bin:/cygdrive/c/Python27*

*PWD=/home/vlad*

*UID=1000*

*USERNAME=vlad*

\$ set

**COMPUTERNAME**=TURING

HOME=/home/vlad

HOMEDRIVE=C:

HOSTNAME=TURING

HOSTTYPE=i686

MANPATH=/usr/local/man:/usr/share/man:/usr/man

NUMBER\_OF\_PROCESSORS=4

OS=Windows\_NT

PATH=/usr/local/bin:/usr/bin:/bin:/cygdrive/c/Windows/system32:

/cygdrive/c/Windows:/cygdrive/c/bin:/cygdrive/c/Python27

PWD=/home/vlad

UID=1000

USERNAME=vlad

Standard to use upper-case names



\$ set

COMPUTERNAME=TURING

HOME=/home/vlad

HOMEDRIVE=C:

HOSTNAME=TURING

HOSTTYPE=i686

MANPATH=/usr/local/man:/usr/share/man:/usr/man

NUMBER\_OF\_PROCESSORS=4

OS=Windows\_NT

PATH=/usr/local/bin:/usr/bin:/bin:/cygdrive/c/Windows/system32:

/cygdrive/c/Windows:/cygdrive/c/bin:/cygdrive/c/Python27

PWD=/home/vlad

UID=1000

USERNAME=vlad

All values are strings

\$ set

COMPUTERNAME=TURING

HOME=/home/vlad

HOMEDRIVE=C:

HOSTNAME=TURING

HOSTTYPE=i686

MANPATH=/usr/local/man:/usr/share/man:/usr/man

NUMBER\_OF\_PROCESSORS=4

OS=Windows\_NT

PATH=/usr/local/bin:/usr/bin:/bin:/cygdrive/c/Windows/system32:

/cygdrive/c/Windows:/cygdrive/c/bin:/cygdrive/c/Python27

PWD=/home/vlad

UID=1000

USERNAME=vlad

All values are strings

Programs must convert to other types when/as necessary

\$ set

COMPUTERNAME=TURING

HOME=/home/vlad

HOMEDRIVE=C:

HOSTNAME=TURING

HOSTTYPE=i686

MANPATH=/usr/local/man:/usr/share/man:/usr/man

NUMBER\_OF\_PROCESSORS=4

OS=Windows\_NT

PATH=/usr/local/bin:/usr/bin:/bin:/cygdrive/c/Windows/system32:

/cygdrive/c/Windows:/cygdrive/c/bin:/cygdrive/c/Python27

PWD=/home/vlad

UID=1000

USERNAME=vlad

int(string) for numbers

```
$ set
```

```
COMPUTERNAME=TURING
```

```
HOME=/home/vlad
```

```
HOMEDRIVE=C:
```

```
HOSTNAME=TURING
```

```
HOSTTYPE=i686
```

```
MANPATH=/usr/local/man:/usr/share/man:/usr/man
```

```
NUMBER_OF_PROCESSORS=4
```

```
OS=Windows_NT
```

```
PATH=/usr/local/bin:/usr/bin:/bin:/cygdrive/c/Windows/system32:  
/cygdrive/c/Windows:/cygdrive/c/bin:/cygdrive/c/Python27
```

```
PWD=/home/vlad
```

```
UID=1000
```

```
USERNAME=vlad
```

split(':') for lists



# PATH controls where the shell looks for programs

PATH controls where the shell looks for programs

\$ ./analyze ← Run the analyze program  
in the current directory

PATH controls where the shell looks for programs

```
$ ./analyze
```

```
$ /bin/analyze
```

← Run the analyze program  
in the /bin directory

## PATH controls where the shell looks for programs

```
$ ./analyze
```

```
$ /bin/analyze
```

```
$ analyze
```



PATH controls where the shell looks for programs

```
$ ./analyze
```

```
$ /bin/analyze
```

```
$ analyze
```

← directories = split(PATH, ':')

for each directory:

if directory/analyze exists, run it

## PATH controls where the shell looks for programs

```
$ ./analyze
```

```
$ /bin/analyze
```

```
$ analyze
```

```
/usr/local/bin
```

```
/usr/bin
```

```
/bin
```

```
/cygdrive/c/Windows/system32
```

```
/cygdrive/c/Windows
```

```
/cygdrive/c/bin
```

```
/cygdrive/c/Python27
```

## PATH controls where the shell looks for programs

```
$ ./analyze
```

```
$ /bin/analyze
```

```
$ analyze
```

```
/usr/local/bin
```

```
/usr/bin
```

```
/bin
```

```
/cygdrive/c/Windows/system32
```

```
/cygdrive/c/Windows
```

```
/cygdrive/c/bin
```

```
/cygdrive/c/Python27
```

```
/bin/analyze
```

```
/cygdrive/c/bin/analyze
```

```
/users/vlad/analyze
```

## PATH controls where the shell looks for programs

```
$ ./analyze
```

```
$ /bin/analyze
```

```
$ analyze
```

```
/usr/local/bin
```

```
/usr/bin
```

```
/bin
```

```
/cygdrive/c/Windows/system32
```

```
/cygdrive/c/Windows
```

```
/cygdrive/c/bin
```

```
/cygdrive/c/Python27
```

```
/bin/analyze
```

```
/cygdrive/c/bin/analyze
```

```
/users/vlad/analyze
```

## PATH controls where the shell looks for programs

```
$ ./analyze
```

```
$ /bin/analyze
```

```
$ analyze
```

```
/usr/local/bin
```

```
/usr/bin
```

```
/bin
```

```
/cygdrive/c/Windows/system32
```

```
/cygdrive/c/Windows
```

```
/cygdrive/c/bin
```

```
/cygdrive/c/Python27
```

```
/bin/analyze
```

```
/cygdrive/c/bin/analyze
```

```
/users/vlad/analyze
```

echo prints its arguments

echo prints its arguments

Use it to show variables' values

echo prints its arguments

Use it to show variables' values

```
$ echo hello transylvania!
```

```
hello transylvania!
```

```
$
```



echo prints its arguments

Use it to show variables' values

```
$ echo hello transylvania!
```

```
hello transylvania!
```

```
$ echo HOME
```

echo prints its arguments

Use it to show variables' values

```
$ echo hello transylvania!
```

```
hello transylvania!
```

```
$ echo HOME
```

```
HOME
```

```
$
```

echo prints its arguments

Use it to show variables' values

```
$ echo hello transylvania!
```

```
hello transylvania!
```

```
$ echo HOME
```

```
HOME
```

```
$ echo $HOME
```

```
/home/vlad
```

```
$
```

echo prints its arguments

Use it to show variables' values

```
$ echo hello transylvania!
```

```
hello transylvania!
```

```
$ echo HOME
```

```
HOME
```

```
$ echo $HOME
```

```
/home/vlad
```

```
$
```

Ask shell to replace variable name  
with value before program runs



echo prints its arguments

Use it to show variables' values

```
$ echo hello transylvania!
```

```
hello transylvania!
```

```
$ echo HOME
```

```
HOME
```

```
$ echo $HOME
```

```
/home/vlad
```

```
$
```

Ask shell to replace variable name  
with value before program runs  
Just like \* and ? are expanded  
before the program runs

echo prints its arguments

Use it to show variables' values

```
$ echo hello transylvania!
```

```
hello transylvania!
```

```
$ echo HOME
```

```
HOME
```

```
$ echo $HOME → echo /home/vlad
```

```
/home/vlad
```

```
$
```

# Create variable by assigning to it

Create variable by assigning to it

Change values by reassigning to existing variables



Create variable by assigning to it

Change values by reassigning to existing variables

```
$ SECRET_IDENTITY=Dracula
```

```
$ echo $SECRET_IDENTITY
```

*Dracula*

```
$ SECRET_IDENTITY=Camilla
```

```
$ echo $SECRET_IDENTITY
```

*Camilla*

```
$
```

Assignment only changes variable's value  
in *this* shell

Assignment only changes variable's value  
in *this* shell

```
$ SECRET_IDENTITY=Dracula
```

```
$ echo $SECRET_IDENTITY
```

```
Dracula
```

```
$
```

Assignment only changes variable's value  
in *this* shell

```
$ SECRET_IDENTITY=Dracula
```

```
$ echo $SECRET_IDENTITY
```

```
Dracula
```

```
$ bash
```

```
$
```

Assignment only changes variable's value  
in *this* shell

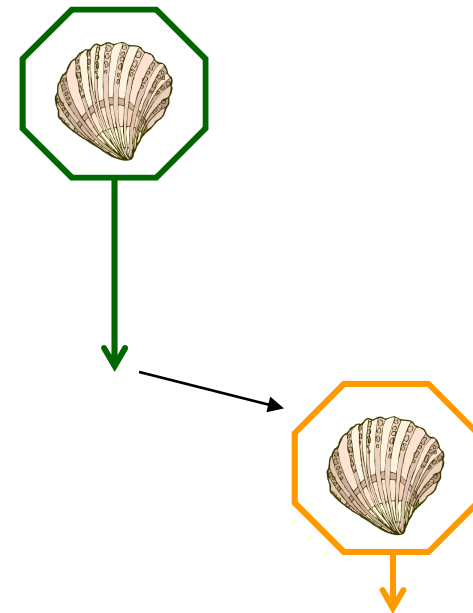
```
$ SECRET_IDENTITY=Dracula
```

```
$ echo $SECRET_IDENTITY
```

```
Dracula
```

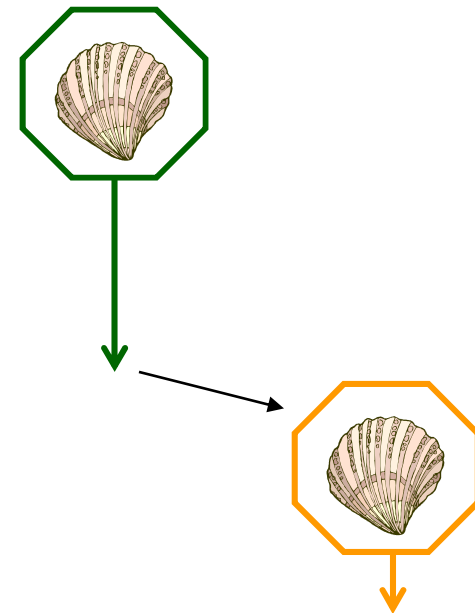
```
$ bash
```

```
$
```



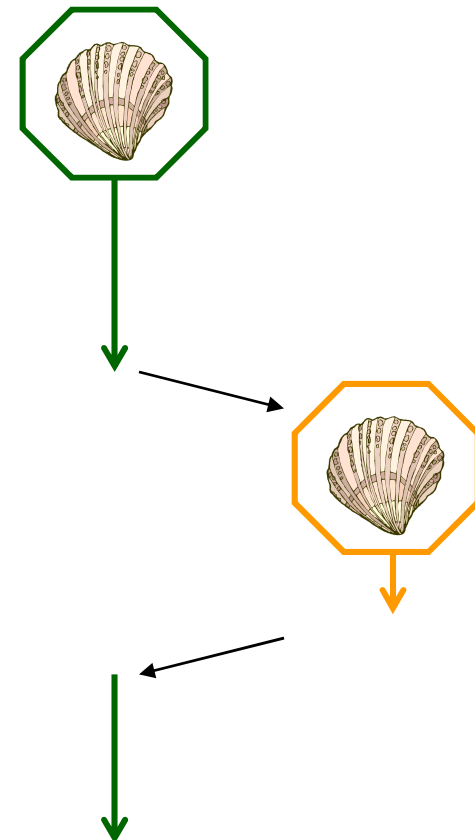
Assignment only changes variable's value  
in *this* shell

```
$ SECRET_IDENTITY=Dracula
$ echo $SECRET_IDENTITY
Dracula
$ bash
$ echo $SECRET_IDENTITY
$
```



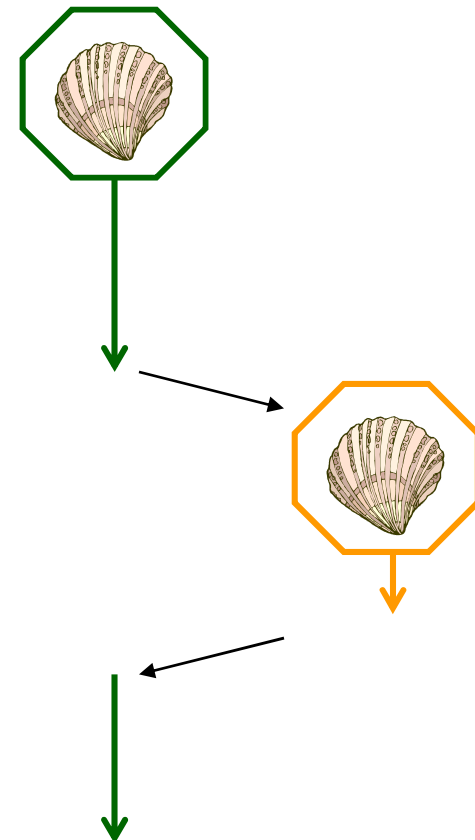
Assignment only changes variable's value  
in *this* shell

```
$ SECRET_IDENTITY=Dracula
$ echo $SECRET_IDENTITY
Dracula
$ bash
$ echo $SECRET_IDENTITY
$ exit
$
```



Assignment only changes variable's value  
in *this* shell

```
$ SECRET_IDENTITY=Dracula
$ echo $SECRET_IDENTITY
Dracula
$ bash
$ echo $SECRET_IDENTITY
$ exit
$ echo $SECRET_IDENTITY
Dracula
$
```

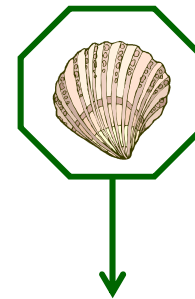




Use export to signal that the variable should be visible to subprocesses

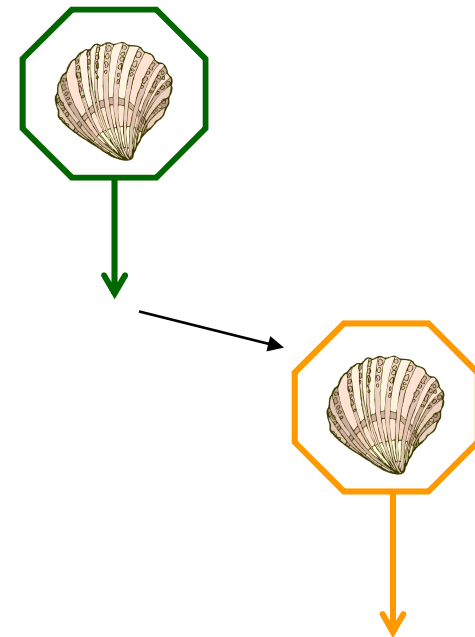
Use export to signal that the variable should be visible to subprocesses

```
$ SECRET_IDENTITY=Dracula  
$ export SECRET_IDENTITY  
$
```



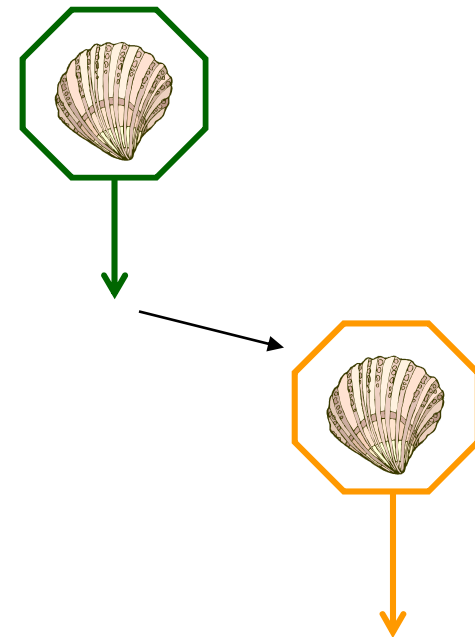
Use export to signal that the variable should be visible to subprocesses

```
$ SECRET_IDENTITY=Dracula  
$ export SECRET_IDENTITY  
$ bash  
$
```



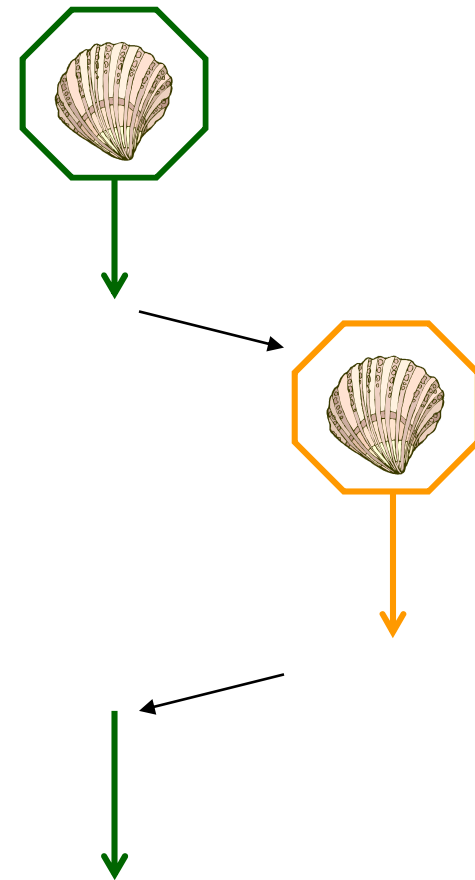
Use export to signal that the variable should be visible to subprocesses

```
$ SECRET_IDENTITY=Dracula
$ export SECRET_IDENTITY
$ bash
$ echo $SECRET_IDENTITY
Dracula
$
```



Use export to signal that the variable should be visible to subprocesses

```
$ SECRET_IDENTITY=Dracula
$ export SECRET_IDENTITY
$ bash
$ echo $SECRET_IDENTITY
Dracula
$ exit
$
```



Commands in `$HOME/.bashrc` are executed  
when shell starts

Commands in `$HOME/.bashrc` are executed  
when shell starts

```
export SECRET_IDENTITY=Dracula  
export BACKUP_DIR=$HOME/backup
```

`/home/vlad/.bashrc`

Commands in `$HOME/.bashrc` are executed  
when shell starts

```
export SECRET_IDENTITY=Dracula  
export BACKUP_DIR=$HOME/backup
```

Also common to use `alias` to create shortcuts



Commands in `$HOME/.bashrc` are executed  
when shell starts

```
export SECRET_IDENTITY=Dracula  
export BACKUP_DIR=$HOME/backup
```

Also common to use `alias` to create shortcuts

```
alias backup=/bin/zarble -v --nostir -R 20000 $HOME $BACKUP_DIR
```

Commands in `$HOME/.bashrc` are executed  
when shell starts

```
export SECRET_IDENTITY=Dracula  
export BACKUP_DIR=$HOME/backup
```

Also common to use `alias` to create shortcuts

```
alias backup=/bin/zarble -v --nostir -R 20000 $HOME $BACKUP_DIR
```

Not something you want to type over and over



created by

Greg Wilson

August 2010



Copyright © Software Carpentry 2010

This work is licensed under the Creative Commons Attribution License

See <http://software-carpentry.org/license.html> for more information.