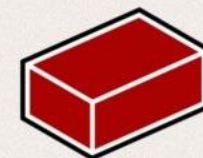


Setting up a Dev Space

Welcome!

So you wanna write code?

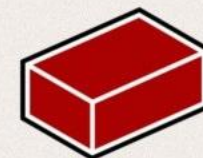


Redbrick
DCU's Networking Society

This is what you need right?

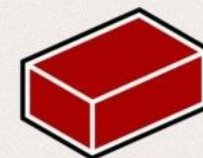


moar rgb == better



Redbrick
DCU's Networking Society

But this is impractical..

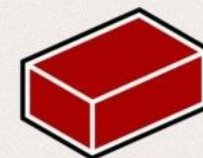
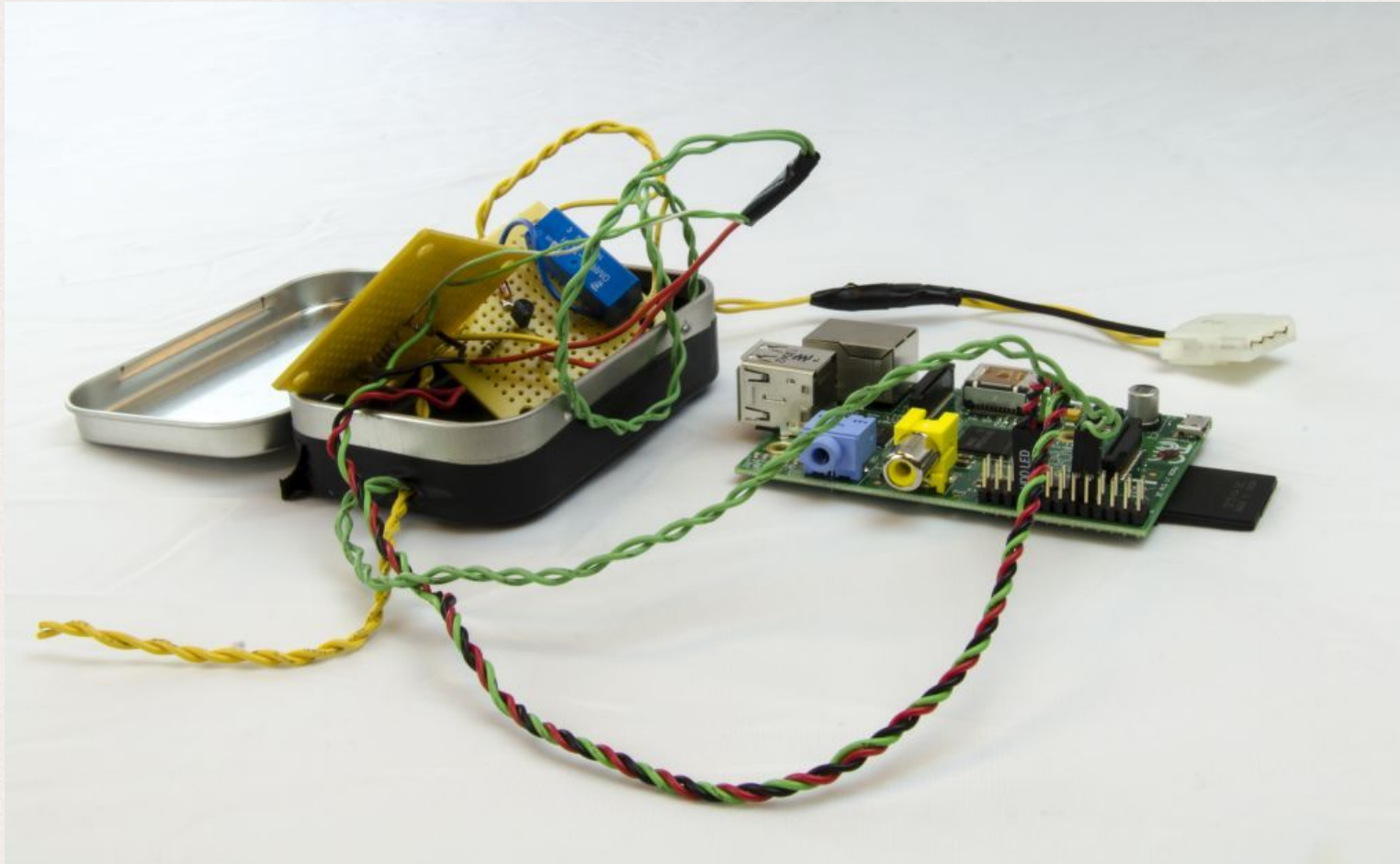


Redbrick
DCU's Networking Society

So we figured out size is a factor...

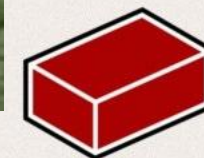


I'll just carry a raspberry pi!



Redbrick
DCU's Networking Society

We might need something more powerful than this!

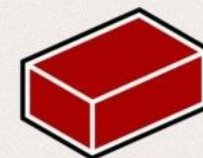


Redbrick
DCU's Networking Society

But not much!

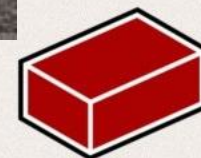
More than likely any mid range laptop will do!

Small and Powerful



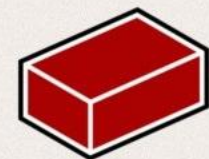
Redbrick
DCU's Networking Society

You definitely need a Razer Blade



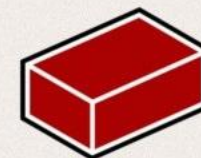
Redbrick
DCU's Networking Society

Or a macbook pro



Redbrick
DCU's Networking Society

And especially a dell xps



Redbrick
DCU's Networking Society

Now if you have a sugar daddy...

... by all means go and get one of these fantastic beasts

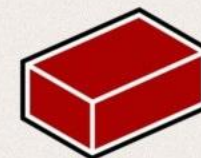
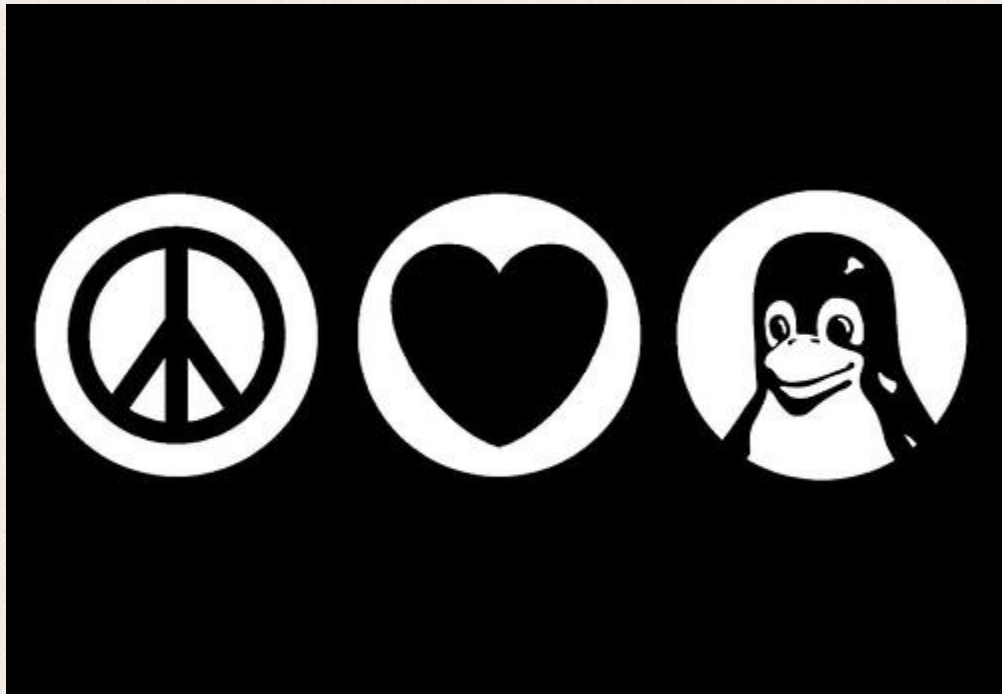


But you don't need these laptops

especially if you are running linux...



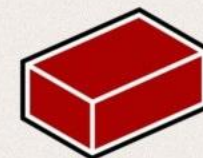
and we love linux



Redbrick
DCU's Networking Society

Dells usually take linux well..

.. some even you can buy with ubuntu

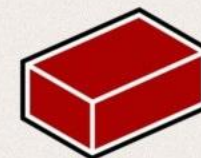


Redbrick
DCU's Networking Society

Thinkpads are great too



I really like these machines



Redbrick
DCU's Networking Society

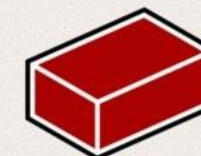
Adverts, Done Deal and Ebay

- Make sure that you know what to look for
- Don't be afraid to say when something is not up to standard

But they can be really good!



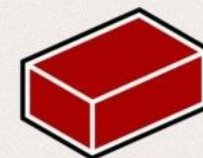
But there are so many versions of linux!



Redbrick

DCU's Networking Society

But there are so many versions of linux!



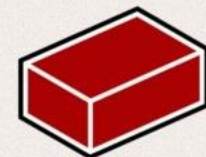
Redbrick
DCU's Networking Society



Ubuntu, Fedora or Linux Mint

All these distros are good starting points

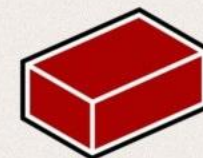
They are both easy to install and easy to use



Redbrick
DCU's Networking Society

Now that you are all setup...

You want to write some code!



Redbrick
DCU's Networking Society

But what text editor do you use?

But what text editor do you use?

There are at least 2..

But what text editor do you use?

The two I speak of is emacs and vim

But what text editor do you use?

I have no idea how to use emacs

But what text editor do you use?

So I use vim

But what text editor do you use?

The whole point of it is you don't have to use the mouse

You can navigate and edit all your text with the keyboard

But what text editor do you use?

But.... It does have a steep learning curve

gg
up 1 line

h sp -> dash-motions

	ts	sw	sts	et	
use spaces only	n	n	n	on	tabstop sw Columns per tabstop
use tabs only	n	n	0	off	shiftwidth sw Columns per <code><<</code>
Set n to desired tab width (default 8)					softtabstop sts Spaces per tab
					expandtab et <code><Tab></code> inserts spaces

MIXING TABS AND SPACES IS RIGHT OUT.
(that means don't do it.)


u
up 1/2 page

:retab Replace all tabs with spaces according to current tabstop setting

k
up 1 line

fileformat ff	Try changing this if your line-endings are messed up
list	Display whitespace visibly according to listchars

h left-right-motions



vim

h: tags-and-searches

h: keycodes

[Jump to tag under cursor, including [tags] in help files	<CR>	^m	\r	Enter
^t	Jump back up the tag-list	<Tab>	^i	\t	Tab
g^]	Jump to tag if it's the only match; else list matching tags	<C- <i>n</i> >	^n		Ctrl- <i>n</i>
		<M- <i>n</i> >			Alt- <i>n</i>
		<Esc>	^[Escape
		<BS>	^h	\b	Backspace
					Delete

	next character l	end of word e	beginning of next word w	end of word E	beginning of next word w	end of line \$
j down 1 line	p paste after	P paste before	^ [return to Normal mode			
u undo	^r redo	. repeat				
gd down ½ page	gf find file under cursor in path and jump to it	dd delete current line	yy yank current line			
x delete character after cursor	% jump to matching paren	r replace char under cursor				
nf down 1 page	nG jump to line n	oG jump back	^i jump forward			
zz center screen on cursor	zt align top of screen with cursor	zb align bottom of screen with cursor				
G last line	== auto-indent current line	<< shift current line left by shiftwidth	>> shift current line right by shiftwidth			

hidden	hid	Lets you switch buffers without saving
laststatus	ls	Show status line never (0), always (2) or with 2+ windows (1)
hlsearch	hls	Highlight search matches. Also see 'highlight'
number	nu	Show line numbers
showcmd	sc	Show commands as you type them
ruler	ru	Show line and column number of the cursor
backspace	bs	Set to '2' to make backspace work like sane editors
wrap		Control line wrapping
background	bg	Set to 'dark' if you have a dark color scheme

Using **⌘** to return to Normal mode lets you keep your fingers on the home row. It's even easier if you map Caps Lock to Control!

`:h insert.txt`

COOL INSERT MODE STUFF

⌘w	delete word before cursor	⌘u	delete line before cursor
⌘r	insert the contents of register z	⌘r=	use the expression register (try <code>⌘r=10</code>)
⌘t	increase line indent by shiftwidth	⌘d	decrease line indent by shiftwidth
⌘x	line completion	⌘n	find next completion suggestion according to complete

`:h cmdline.txt`

COMMAND-LINE MODE ONLY

REGISTERS are CLIPBOARDS

All commands that delete, copy, or paste text use registers. To change which register is used by a command, type the register before the command. The default register is called "the unnamed register" and it is invoked with a pair of double-quotes (""). Typing **dd** or **yy** is the same as typing **"dd** or **"yy**. Think of the first **"** as a short way of saying "register", so **"dd** is pronounced "register d", and **"a**, "register a".

```
:registers
```

View all current registers

```
:echo @r
```

Access register **r** as a variable

```
" /
```

Last search pattern register

Contains the last pattern you searched for

```
" _
```

The black hole register

Use this to delete without clobbering any register (**"dd**)

Use **:map** to view all current custom key mappings. Read **:h map-which-keys** for a guide on which keys are best for your own custom mappings. Get used to Vim's help system - it's a fantastic resource!

edit using
Normal mode
undo

f

insert word
under cursor

A

a

completion
suggestions

d

cnd lno - complete line

Put `<normal> %s (%-C-R)-expand(%h,h) / ^ ^ ^ ^ ^` in your `vimrc` so you can type `%s` in Command-line mode to refer to the directory of the current file, regardless of `%:`.

Supply `%s` as a range to the `substitute` command to run it on every line in the file.

```
%s:/Scriptbl/Design/ "Scribled" -> "Designed"
```

Specify the `g` flag to apply the substitution to every match on a line.

```
%s:[dia]//g "badly" -> "by" %s:flags, /h /l
```

Vim supports many regular expression features.

```
%s:./k/ax/ "Mook" -> "Max" %h user_27, /h /z
```

Use `__` instead of `__` if you want to search across multiple lines.

```
%s:/heat_\_.+Bungle/anto/ "CheatSheet/nBungle" -> "Cantor" %h \_\_
```

Special escapes can be used to change the case of substitutions.

```
%s:(\f.\_)\_U\VE_ "foobar" -> "FOOBAR" %h sub-replace-special
```

Use `g` global to perform a command on matching lines.

```
%g/foobar/d\etete Delete all lines containing "foobar"
```

If your pattern contains slashes, just use a different character as your delimiter.

```
%s:_Data/Lore_Brent_Spiner_
```

Use `=` to evaluate expressions with replacement groups.

```
%s:_d_\=submatch(0) + 1_g "10 25" -> "21 36" %h sub-replace-lu
```

"0"	Last yank register	Contains the last text you yanked	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
"1"	Last big delete register	Contains the last line(s) you deleted	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
"2"-9"	Big delete register stack	Every time "0" is written to, its content is pushed to "2" , then "2" to "3" , and so on	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
"_"	Small delete register	Contains the last text you deleted within a single line	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
"+"	System clipboard	If the OS integration gods smile upon you, this register reads and writes to your system clipboard.	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
"a"-z"	Named registers	26 registers for you to play with	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
"A"-Z"	Append registers	Using upper-case to refer to a register will append to it rather than overwrite it	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
qr	Record	Record into register r . Stop recording by hitting q again	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
@r	Playback	Execute the contents of register r	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis
@@	Repeat last playback	Repeat the last @r , this is particularly useful with a count	Ed: Erik Nieuwenhuis Ed: Erik Nieuwenhuis

vim one-liner used to sort the list of names by length:

```
vim -c ':sort $@' vimrc
```


But what text editor do you use?

If you want an easier text editor to get you started you can look into:

- atom
- VS code
- sublime
- gedit
- nano

What are IDEs?

These are integrated development environments...

What are IDEs?

These are integrated development environments...

- They do a lot of work for you

What are IDEs?

These are integrated development environments...

- They do a lot of work for you
- They autocomplete words

What are IDEs?

These are integrated development environments...

- They do a lot of work for you
- They autocomplete words
- Auto add brackets
- And sooo much more.....

What are IDEs?

BUT...

What are IDEs?

You don't wanna use it....

What are IDEs?

Now that may be controversial

What are IDEs?

Now that may be controversial

But in my opinion they hinder learning

What are IDEs?

Now that may be controversial

But in my opinion they hinder learning

You won't have one in an exam, coding interview, etc

What are IDEs?

By all means when you get out into industry and are writing
professional

Get an IDE it will make you life so much better!

What are IDEs?

But when you are learning as a student

Do as much as you can manually so you learn how things work

A better solution?

I am going to introduce you very slightly to my setup...

.. not that it is good just a taste of the vast customizability you can get with linux

My setup most days

```
def main():
    answers = []
    N = int(input())
    while N > 0:
        noAd, ad, cost = input().split(" ")
        profit = int(ad) - int(cost)
        noAd = int(noAd)
        if profit > noAd:
            answers.append("advertise")
        elif profit == noAd:
            answers.append("does not matter")
        else:
            answers.append("do not advertise")
        N -= 1

    for answer in answers:
        print(answer)
```

```
if __name__ == '__main__':  
    main()
```

[illegible][illegible]

```
greenday@saturn:~/Documents/kattis$ python3
modulo.py          nastyhacks.py          QuadrantSelect.py  r2.py             tarifa.py
greenday@saturn:~/Documents/kattis$ python3 nastyhacks.py
1 2 3
Traceback (most recent call last):
  File "nastyhacks.py", line 20, in <module>
    main()
  File "nastyhacks.py", line 3, in main
    N = int(input())
ValueError: invalid literal for int() with base 10: '1 2 3'
greenday@saturn:~/Documents/kattis$
```

"saturn" 22:29 03-Oct-17

My setup most days

As you can see I am using tmux

```
def main():
    answers = []
    N = int(input())
    while N > 0:
        noAd, ad, cost = input().split(" ")
        profit = int(ad) - int(cost)
        noAd = int(noAd)
        if profit > noAd:
            answers.append("advertise")
        elif profit == noAd:
            answers.append("does not matter")
        else:
            answers.append("do not advertise")
        N -= 1

    for answer in answers:
        print(answer)

if __name__ == '__main__':
    main()
```

```
greenday@saturn:~/Documents/kattis$ python3
modulo.py          nastyhacks.py      QuadrantSelect.py  r2.py              tarifa.py
greenday@saturn:~/Documents/kattis$ python3 nastyhacks.py
1 2 3
Traceback (most recent call last):
  File "nastyhacks.py", line 20, in <module>
    main()
  File "nastyhacks.py", line 3, in main
    N = int(input())
ValueError: invalid literal for int() with base 10: '1 2 3'
greenday@saturn:~/Documents/kattis$
```

"saturn" 22:29 03-Oct-17

My setup most days

```
def main():
    answers = []
    N = int(input())
    while N > 0:
        noAd, ad, cost = input().split(" ")
        profit = int(ad) - int(cost)
        noAd = int(noAd)
        if profit > noAd:
            answers.append("advertise")
        elif profit == noAd:
            answers.append("does not matter")
        else:
            answers.append("do not advertise")
        N -= 1

    for answer in answers:
        print(answer)

if __name__ == '__main__':
    main()
```

1

2

This allows me to
split screen in 3

3

```
greenday@saturn:~/Documents/kattis$ python3
modulo.py          nastyhacks.py      QuadrantSelect.py  r2.py              tarifa.py
greenday@saturn:~/Documents/kattis$ python3 nastyhacks.py
1 2 3
Traceback (most recent call last):
  File "nastyhacks.py", line 20, in <module>
    main()
  File "nastyhacks.py", line 3, in main
    N = int(input())
ValueError: invalid literal for int() with base 10: '1 2 3'
greenday@saturn:~/Documents/kattis$
```

"saturn" 22:29 03-Oct-17

My setup most days

```
def main():
    answers = []
    N = int(input())
    while N > 0:
        noAd, ad, cost = input().split(" ")
        profit = int(ad) - int(cost)
        noAd = int(noAd)
        if profit > noAd:
            answers.append("advertise")
        elif profit == noAd:
            answers.append("does not matter")
        else:
            answers.append("do not advertise")
        N -= 1

    for answer in answers:
        print(answer)
```

I have my
code

My input
data

Where I run
the program

```
greenday@saturn:~/Documents/kattis$ python3
modulo.py          nastyhacks.py      QuadrantSelect.py  r2.py              tarifa.py
greenday@saturn:~/Documents/kattis$ python3 nastyhacks.py
1 2 3
Traceback (most recent call last):
  File "nastyhacks.py", line 20, in <module>
    main()
  File "nastyhacks.py", line 3, in main
    N = int(input())
ValueError: invalid literal for int() with base 10: '1 2 3'
greenday@saturn:~/Documents/kattis$
```

"saturn" 22:29 03-Oct-17

My setup most days

```
def main():
    answers = []
    N = int(input())
    while N > 0:
        noAd, ad, cost = input().split(" ")
        profit = int(ad) - int(cost)
        noAd = int(noAd)
        if profit > noAd:
            answers.append("advertise")
        elif profit == noAd:
            answers.append("does not matter")
        else:
            answers.append("do not advertise")
        N -= 1

    for answer in answers:
        print(answer)

if __name__ == '__main__':
    main()
```

And tmux has
key bindings

That once I know
them

I don't need the
mouse to move
spaces

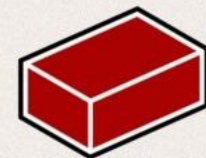
1,1

All

```
greenday@saturn:~/Documents/kattis$ python3
modulo.py          nastyhacks.py      QuadrantSelect.py  r2.py              tarifa.py
greenday@saturn:~/Documents/kattis$ python3 nastyhacks.py
1 2 3
Traceback (most recent call last):
  File "nastyhacks.py", line 20, in <module>
    main()
  File "nastyhacks.py", line 3, in main
    N = int(input())
ValueError: invalid literal for int() with base 10: '1 2 3'
greenday@saturn:~/Documents/kattis$
```

"saturn" 22:29 03-Oct-17

Now we need to run our programs



Redbrick
DCU's Networking Society

Now we need to run our programs

First python:

Now we need to run our programs

First python2

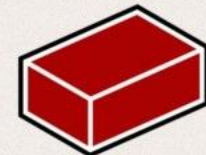
If you are on ubuntu linux it is already installed:

```
python2
```

This will bring up a python shell

To run your written python type

```
python2 your_file.py
```



Redbrick
DCU's Networking Society

Now we need to run our programs

For python3

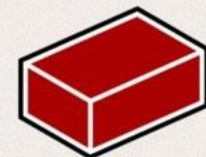
If you are on ubuntu linux it is already installed:

```
python3
```

This will bring up a python shell

To run your written python type

```
python3 your_file.py
```



Redbrick
DCU's Networking Society

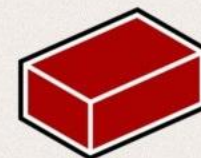
Now we need to run our programs

We can't tell you how to install every language on every operating system

Now we need to run our programs

But we have a friend who can!

Google



Redbrick
DCU's Networking Society

What you need to find

Search:

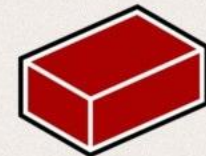
How to install {x} on {y}

where {x} = language

{y} = operating system

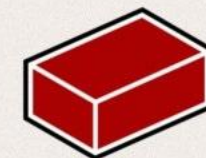
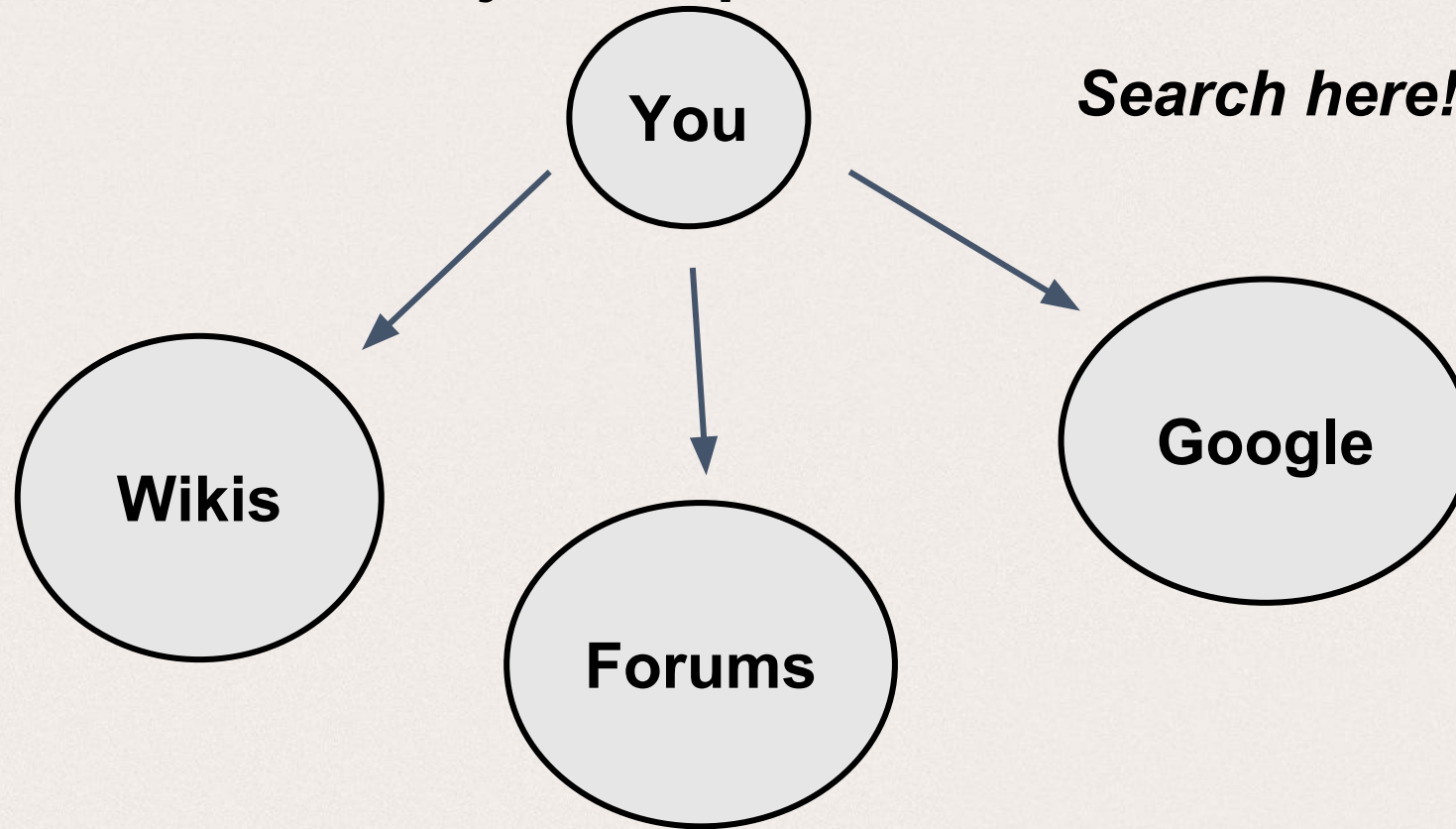
e.g. for fedora and ruby

```
sudo dnf install ruby
```



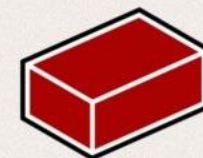
Redbrick
DCU's Networking Society

If you need any help..



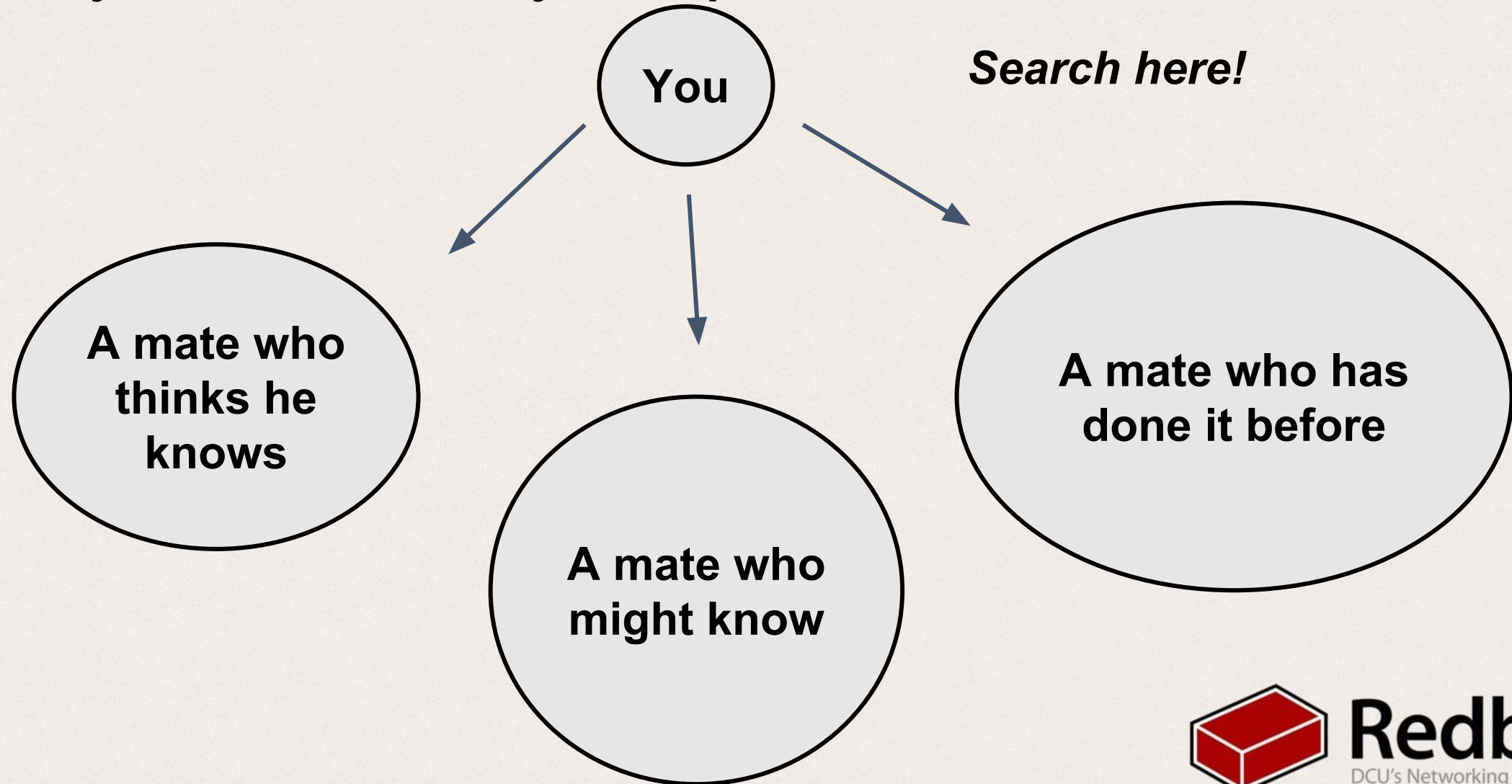
Redbrick
DCU's Networking Society

Can't find what you're looking for?

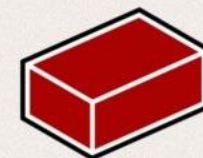


Redbrick
DCU's Networking Society

If you need any help..

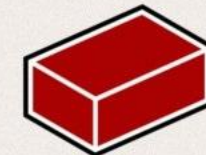
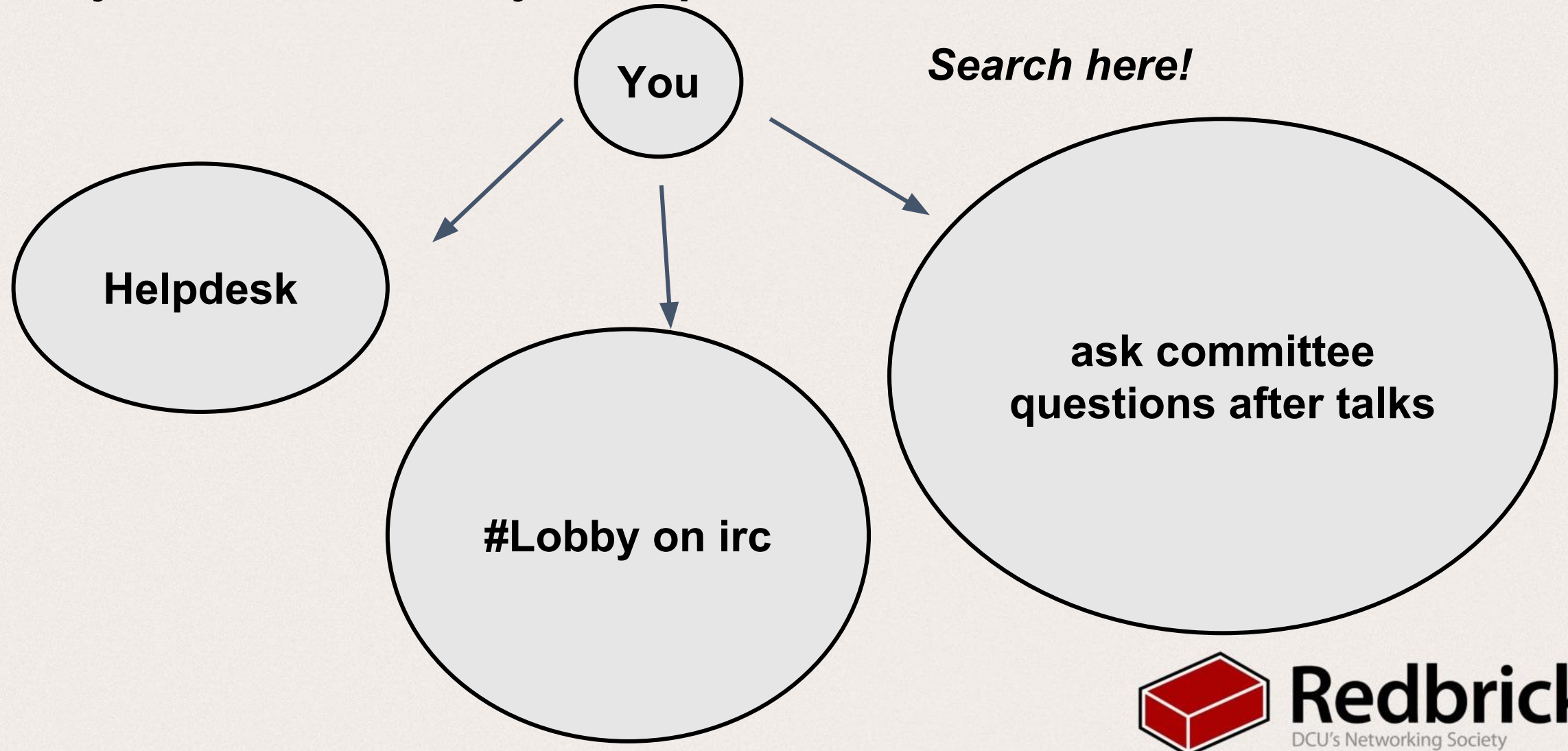


Can't find what you're looking for?



Redbrick
DCU's Networking Society

If you need any help..



Redbrick
DCU's Networking Society

Anyone interested in an install fest?

Everyone come together and tries to install linux at the same time together and help each other.

RIGHT NOW THOUGH

We have an egm right after this talk and you all need to come

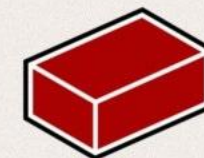
We need a first year rep and

RIGHT NOW THOUGH

We have an egm right after this talk and you all need to come
We need a first year rep and



It might be you



Redbrick
DCU's Networking Society

And tomorrow

We have the fabulous Ben give an intro to using your webspace

