

# A Whirlwind Introduction to Alien Technologies

Alpheus Madsen

Pyrofex Corporation

OpenWest Conference  
Friday, April 12th, 2019

# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

6 Electronics

7 Mathematics

8 Questions

9 References

# Keyboards

VT 100 (and the birth of Vi)



# Keyboards

## Space Cadet (and the birth of Emacs)



# Keyboards



# Keyboards

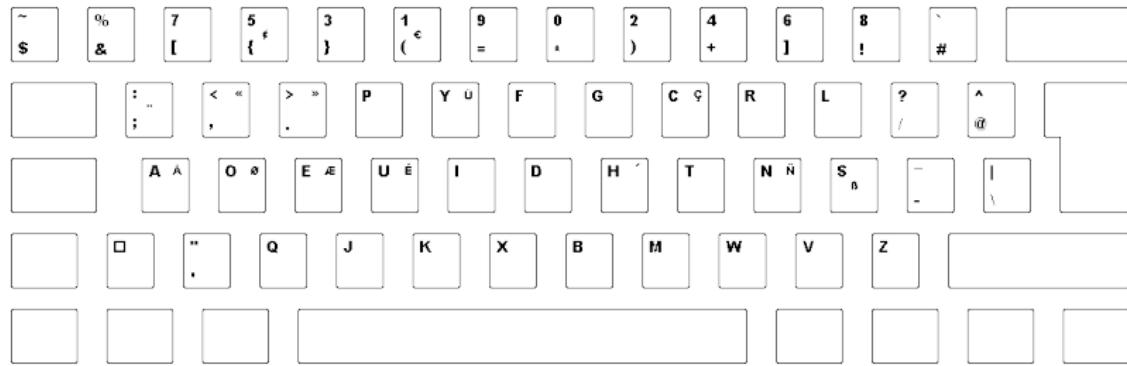
## Dvorak Keyboard Layout



[steve-lovelace.com](http://steve-lovelace.com)

# Keyboards

## Dvorak Programmer's Keyboard Layout



# Keyboards

## MessagEase



# Keyboards

## Janko Piano Keyboard



# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

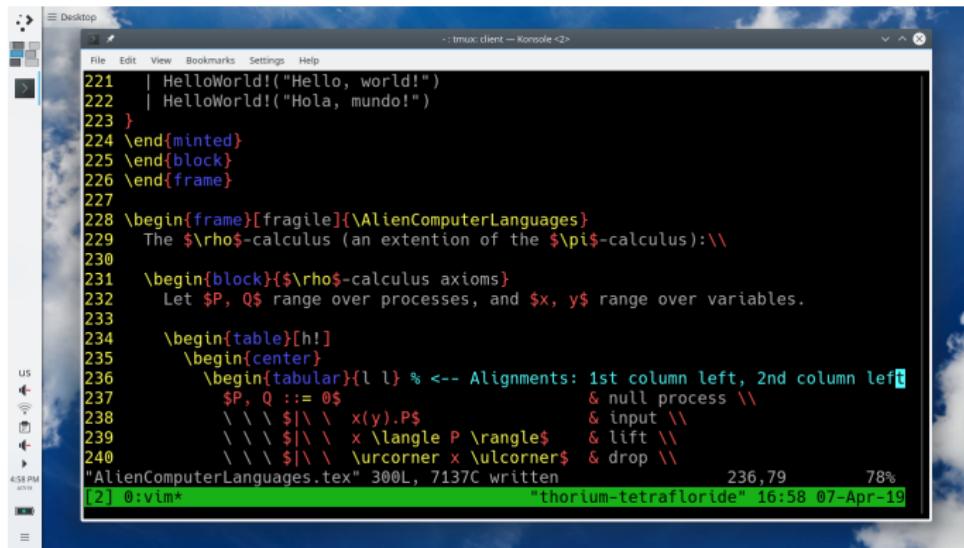
6 Electronics

7 Mathematics

8 Questions

9 References

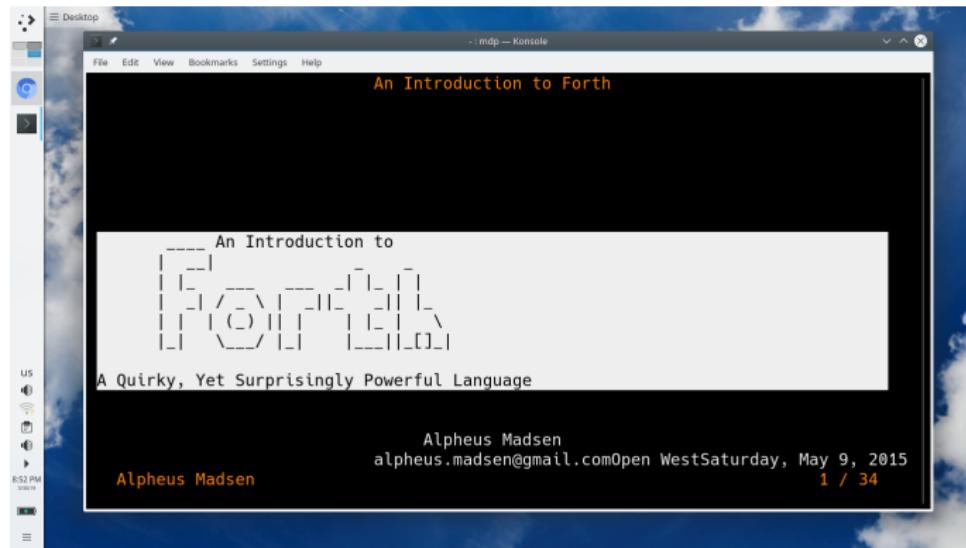
## LATEXand Beamer (as an alternative to Word and PowerPoint)



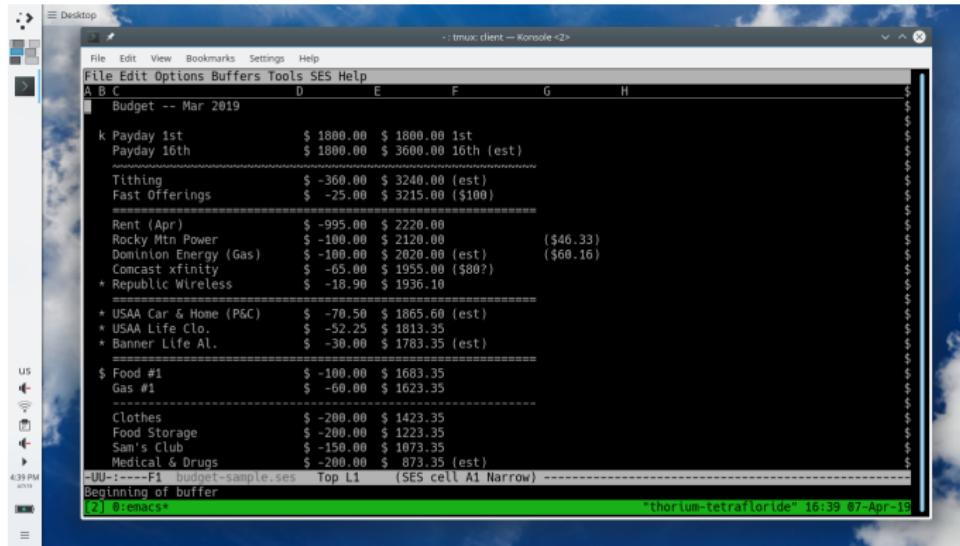
The screenshot shows a tmux session in a terminal window titled "tmux: client — Konsole <2>". The terminal displays LaTeX code for a presentation slide. The code includes a title block, a section for the \$\rho\$-calculus, and a table with two columns. The terminal window is part of a desktop environment with icons for volume, network, and battery status visible on the left.

```
221 | HelloWorld!("Hello, world!")
222 | HelloWorld!("Hola, mundo!")
223 }
224 \end{minted}
225 \end{block}
226 \end{frame}
227
228 \begin{frame}[fragile]{\AlienComputerLanguages}
229   The $\rho$-calculus (an extention of the $\pi$-calculus):\\
230
231   \begin{block}{$\rho$-calculus axioms}
232     Let $P$, $Q$ range over processes, and $x$, $y$ range over variables.
233
234   \begin{table}[h!]
235     \begin{center}
236       \begin{tabular}{l l} % << Alignments: 1st column left, 2nd column left
237         $P$, $Q := 0$ & null process \\
238         $x(y).P$ & input \\
239         $x \langle P \rangle$ & lift \\
240         $x \langle y \rangle$ & drop
241       \end{tabular}
242     \end{center}
243   \end{table}
244
245 \end{block}
246
247 \end{frame}
```

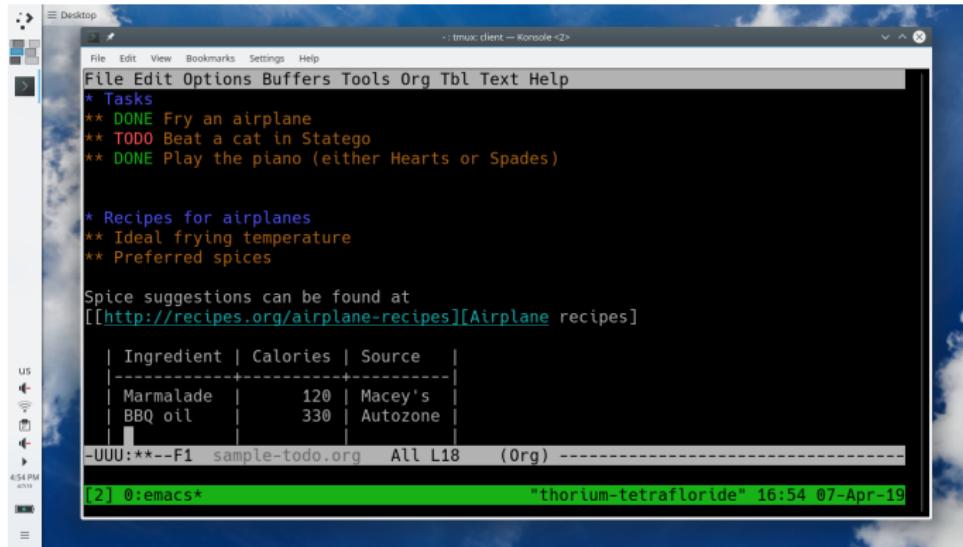
## mdp (as a command line alternative to PowerPoint)



## Emacs SES (an alternative to Excel, for small things)



## Emacs Org Mode (for outlines and text-based notes)



## Lily Pond (for music typesetting)

```
<<
\chords {
    c1:m7 f2:7 c2
}
\relative c' {
    g2 es8( c4) es8
    f8 es d c~ c2
}
\addlyrics {
    You are
    the sky and my rain,
}
>>
```

Enter chord names

Enter melody

Enter lyrics

Combine melody and lyrics



You are the sky and my rain,

# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

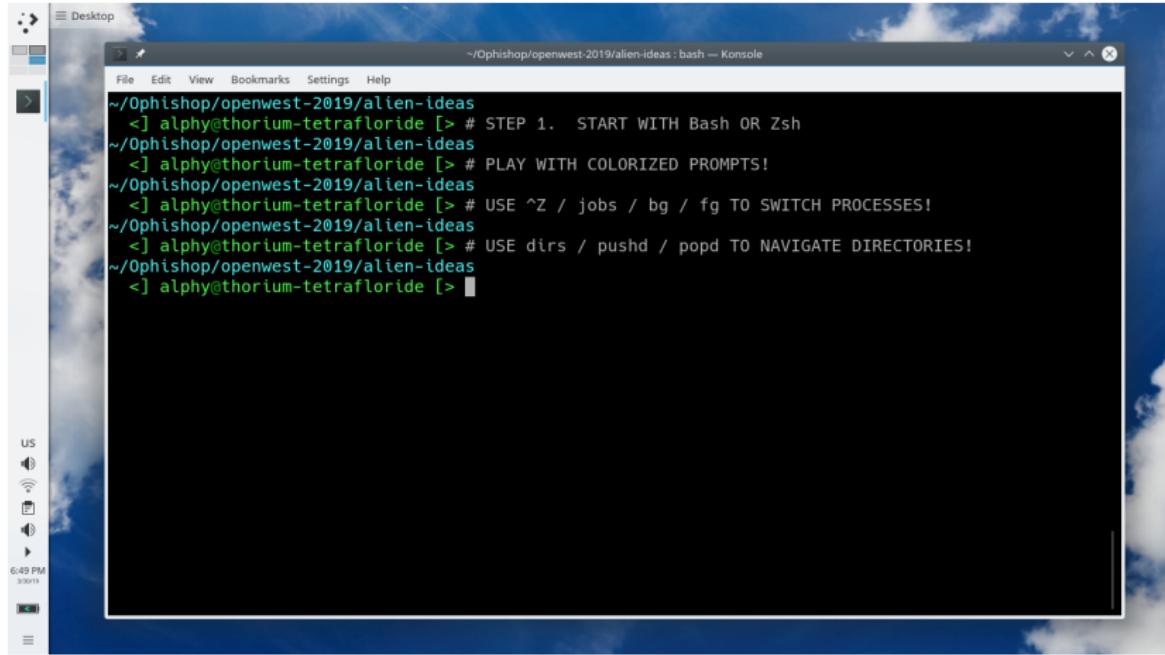
6 Electronics

7 Mathematics

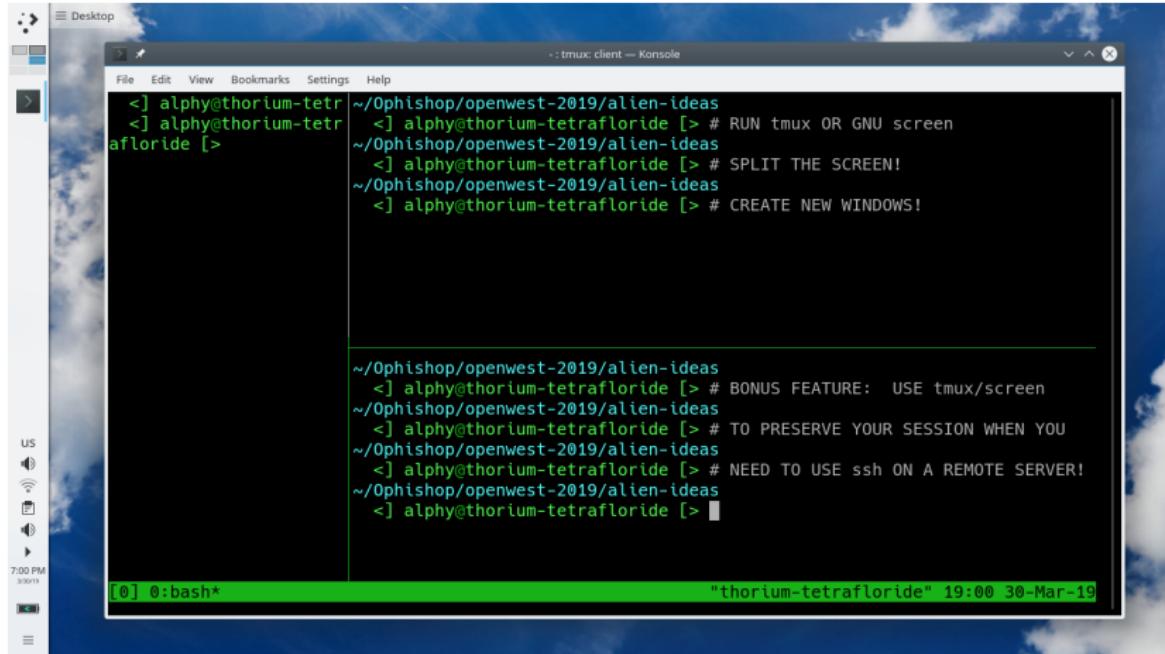
8 Questions

9 References

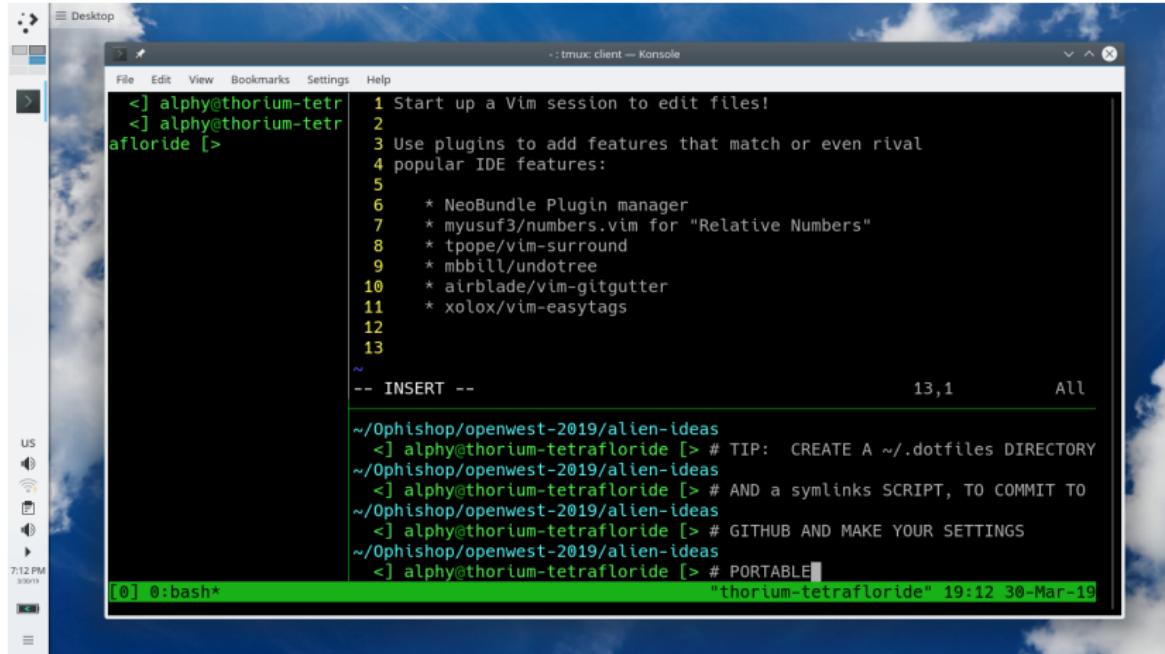
# Build Your Own IDE



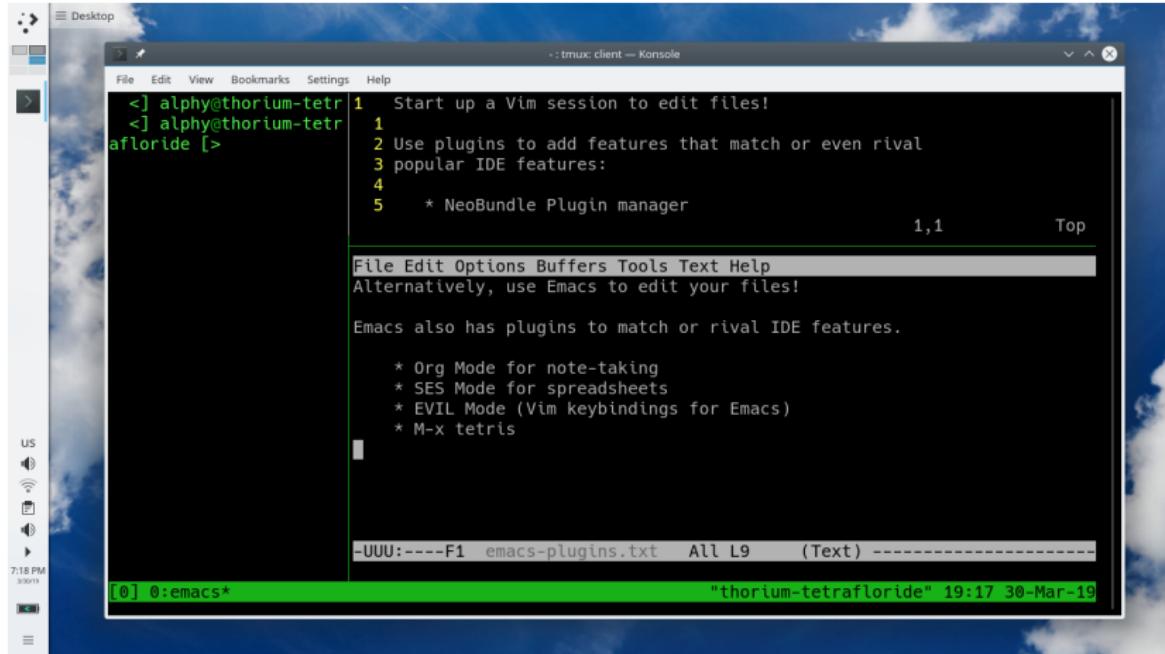
# Build Your Own IDE



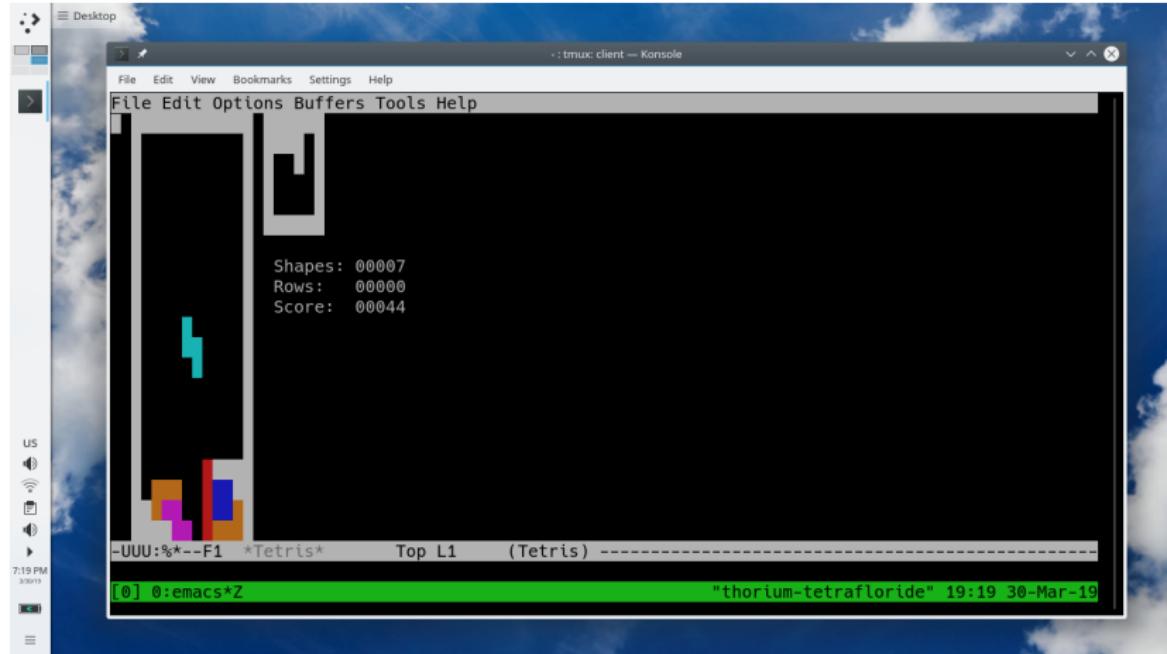
# Build Your Own IDE



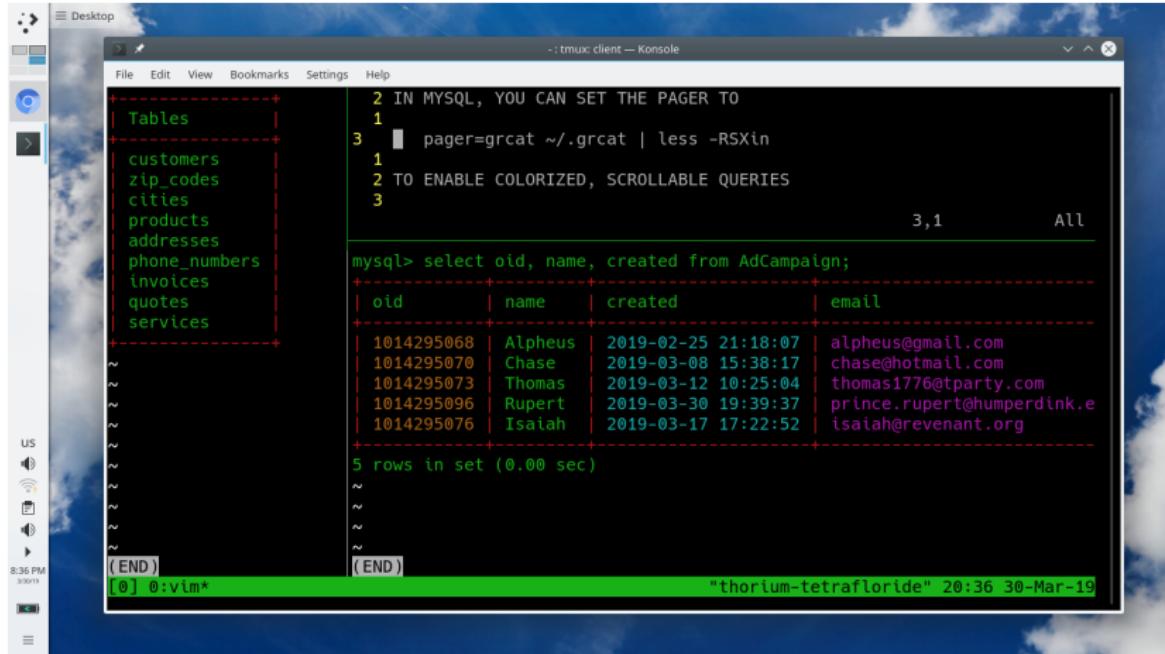
# Build Your Own IDE



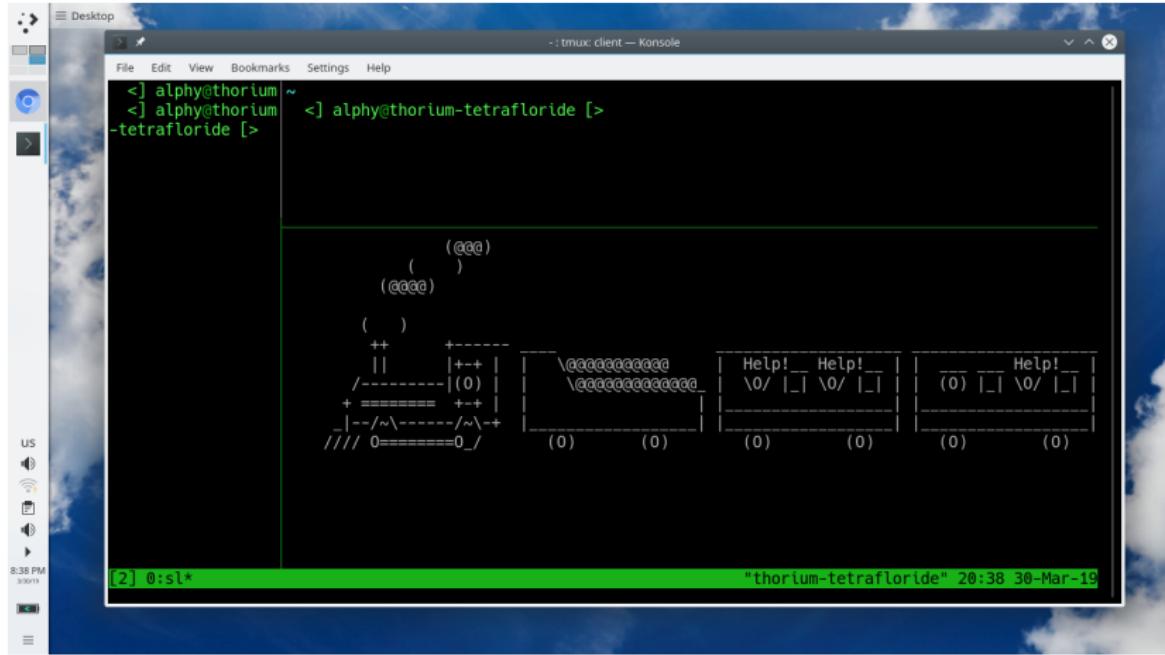
# Build Your Own IDE



# Build Your Own IDE



# Build Your Own IDE



# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

6 Electronics

7 Mathematics

8 Questions

9 References

## Alien Computing – Miscellaneous things

## Alien Computing – Miscellaneous things

- Cryptography for Identification and Privacy

## Alien Computing – Miscellaneous things

- Cryptography for Identification and Privacy
  - nonces
  - cryptographic pseudorandom numbers
  - cryptographic hashes (SHA256) and merkle trees
  - password hashes (scrypt) and salts
  - public key cryptography (RSA)
  - symmetric key cryptography (AES)

## Alien Computing – Miscellaneous things

- RISC-V Architecture (an open source instruction set)

## Alien Computing – Miscellaneous things

- RISC-V Architecture (an open source instruction set)
- NixOS Package Management

## Alien Computing – Computer Languages

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.
- Should make abstractions easy.

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.
- Should make abstractions easy.
- Should be able to do very concrete things.

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.
- Should make abstractions easy.
- Should be able to do very concrete things.

I like to think of such languages as “*transcendental*”.

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.
- Should make abstractions easy.
- Should be able to do very concrete things.

I like to think of such languages as “*transcendental*”.

***Take note!*** Precedence (and syntax, to a certain extent) is the enemy of programming. *Precedence is evil.*

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.
- Should make abstractions easy.
- Should be able to do very concrete things.

I like to think of such languages as “*transcendental*”.

***Take note!*** Precedence (and syntax, to a certain extent) is the enemy of programming. *Precedence is evil.*

- It should be significantly different from C or Python.

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.
- Should make abstractions easy.
- Should be able to do very concrete things.

I like to think of such languages as “*transcendental*”.

***Take note!*** Precedence (and syntax, to a certain extent) is the enemy of programming. *Precedence is evil.*

- It should be significantly different from C or Python.
- (Most importantly) Needs to offer me something new to learn.

## Alien Computing – Computer Languages

What does it take for a computer language to interest me?

- Should be simple.
- Should make abstractions easy.
- Should be able to do very concrete things.

I like to think of such languages as “*transcendental*”.

***Take note!*** Precedence (and syntax, to a certain extent) is the enemy of programming. *Precedence is evil.*

- It should be significantly different from C or Python.
- (Most importantly) Needs to offer me something new to learn.

## A taste of Forth (for microcontroller programming)

### Forth

```
: clockwise 0 ;           : counter-clockwise 1 ;  
  
: step ( motor -- )  
@ dup 1 !      ( run )  
10 delay      ( delay for a step )  
0 ! ;         ( end step )  
  
: set-direction ( direction motor -- )  
2+ @ ! ;  
  
clockwise motor-1 set-direction  
motor-1 step
```

# Alien Computing

## A taste of Prolog (for declarative programming)

### Prolog

```
% A solution to the Towers of Hanoi
move(1,X,Y,_) :-  
    write('Move top disk from '),  
    write(X),  
    write(' to '),  
    write(Y),  
    nl.  
  
move(N,X,Y,Z) :-  
    N>1,  
    M is N-1,  
    move(M,X,Z,Y),  
    move(1,X,Y,_),  
    move(M,Z,Y,X).  
  
?- move(3, left, right, center).
```

## A taste of J (for tinkering with mathematics)

J

```
frac      =: % +/
percent   =: (100 & *) @: frac
round     =: <. @: (+ & 0.5)
comp      =: round @: percent
br        =: ,. ; (,. @: comp)
tr        =: ('Data';'Percentages') & ,:
display   =: tr @: br
```

# Alien Computing

## A taste of Smalltalk (for sending messages to objects)

### Smalltalk

```
" looping over a collection "
#('a' 'b' 'c' ) do[:each |
    Transcript show: each.
    Transcript cr.
].
```

```
" the same, using a local variable "
|myArray|
myArray := #('a' 'b' 'c' ).
myArray do[:each |
    Transcript show: each.
    Transcript cr.
].
```

# Alien Computing

A taste of E and EROS (for object capabilities)

## JavaScript

```
emethod forkReturn (Philosopher thePhilosopher) {  
    if (nowServing == thePhilosopher) {  
        theFork <- revoke ();  
        if (otherIsWaiting) {  
            timerSerialNr += 1;  
            serveOther();  
        } else {  
            nowServing = null;  
            theFork = null;  
        }  
    }  
}
```

# Alien Computing

## A taste of Erlang (for programming with actors)

### Erlang

```
-module(dolphins).  
-compile(export_all).  
  
dolphin() ->  
    receive  
        do_a_flip ->  
            io:format("How about no?~n");  
        fish ->  
            io:format("Thanks for all the fish!~n");  
        _ ->  
            io:format("Heh, we're smarter than humans.~n")  
    end.
```

## A taste of Haskell (for purely functional programming)

### Haskell

```
eggSize :: Float -> Float -> String
eggSize height radius
| volume < hummingbird = "Just humming a bird along."
| volume < ostrich     = "What are you...chicken?"
| otherwise              = "Talk about being ostrich-sized"
where
    volume = 2 * pi * height * radius ^ 2
    hummingbird = 1000
    ostrich     = 5000
```

## A taste of Rholang (for the pi and rho calculi)

### Rholang

```
new HelloWorld, stdout(`rho:io:stdout`) in {
    for (@text <= HelloWorld) {
        stdout!(text)
    }
    | HelloWorld!("Hello, world!")
    | HelloWorld!("Hola, mundo!")
}
```

The  $\rho$ -calculus (an extention of the  $\pi$ -calculus):

## $\rho$ -calculus axioms

Let  $P, Q$  range over processes, and  $x, y$  range over variables.

$P, Q ::= 0$	null process
$x(y).P$	input
$x\langle P \rangle$	lift
$\neg x^\lceil$	drop
$P \mid Q$	parallel
$x, y ::= \lceil P \rceil$	quote

# Alien Computing

A taste of Common Lisp (for the borg-like “all your base are belong to us” nature of the lambda calculus)

## Common Lisp

```
(defview employee-form
  (:type form :default-method :post)
  first-name
  (last-name :requiredp t
             :label "Family Name")
  (contract :present-as (radio :choices
                                '(:full-time :part-time
                                  :consultant :intern))
             :parse-as keyword)
  (age :present-as (input :max-length 3)
       :parse-as integer))
```

The  $\lambda$ -calculus:

## $\lambda$ -calculus axioms

Let  $M, N$  be lambda expressions and  $x, y$  be variables.

 $(\lambda x.M)$ 

Abstraction (Function Definition)

 $(MN)$ 

Function Application

 $(\lambda x.M[x]) \rightarrow (\lambda y.M[y])$ 

$\alpha$ -conversion

 $((\lambda x.M)E) \rightarrow (M[x := E])$ 

$\beta$ -reduction

# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

6 Electronics

7 Mathematics

8 Questions

9 References

# Nuclear Power!

## Home-brew Fusion



# Nuclear Power!

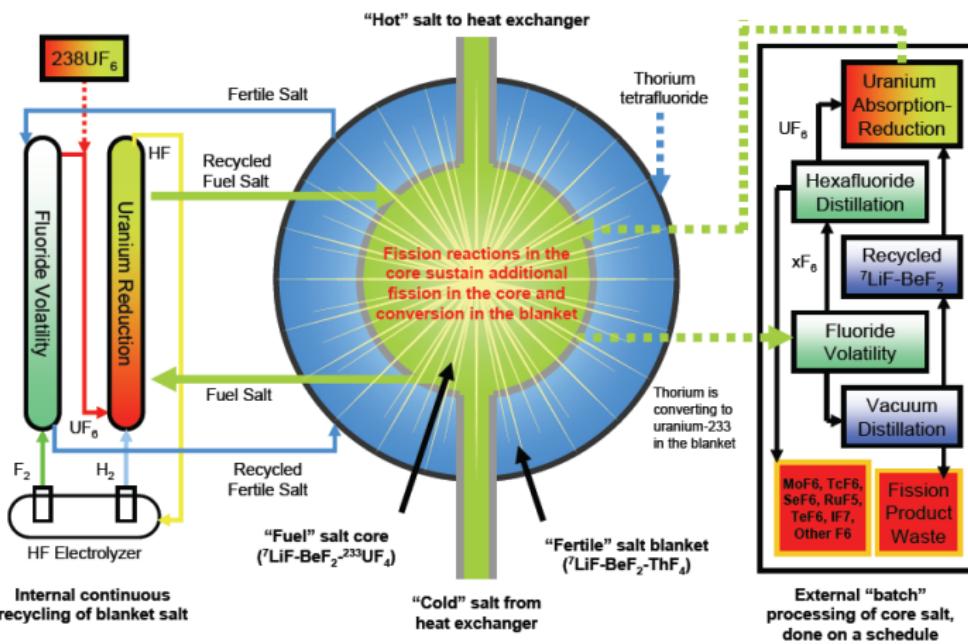
## Polywell Fusion



# Nuclear Power!

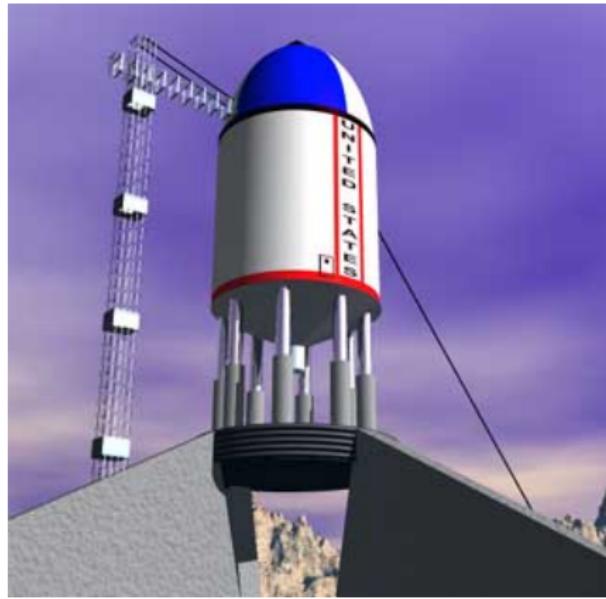
## Molten Salt Fission

### How does a fluoride reactor use thorium?



# Nuclear Power!

The Orion Project: To Saturn by 1970!



# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

6 Electronics

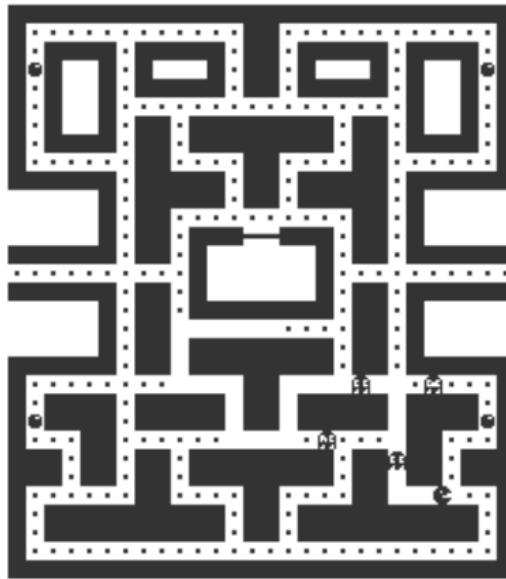
7 Mathematics

8 Questions

9 References

## From NAND to Tetris

Score: 3500  
Lives: 3  
Level: 0



## Ternary Computing



## Ternary Computing: Balanced Ternary

Base-10 Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

## Ternary Computing: Balanced Ternary

Base-10 Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ternary Digits: 0, 1, 2

## Ternary Computing: Balanced Ternary

Base-10 Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ternary Digits: 0, 1, 2

Balanced Ternary Digits:  $\hat{1}$ , 0, 1

## Ternary Computing: Balanced Ternary

Base-10 Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ternary Digits: 0, 1, 2

Balanced Ternary Digits:  $\hat{1}$ , 0, 1

Base 10

---

6

5

4

3

2

1

0

-1

-2

-3

-4

-5

-6

## Ternary Computing: Balanced Ternary

Base-10 Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ternary Digits: 0, 1, 2

Balanced Ternary Digits:  $\hat{1}$ , 0, 1

Base 10	10's Comp
6	6
5	5
4	4
3	3
2	2
1	1
0	0
-1	999
-2	998
-3	997
-4	996
-5	995
-6	994

## Ternary Computing: Balanced Ternary

Base-10 Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ternary Digits: 0, 1, 2

Balanced Ternary Digits:  $\hat{1}$ , 0, 1

Base 10	10's Comp	Ternary
6	6	20
5	5	12
4	4	11
3	3	10
2	2	2
1	1	1
0	0	0
-1	999	222
-2	998	221
-3	997	220
-4	996	212
-5	995	211
-6	994	210

## Ternary Computing: Balanced Ternary

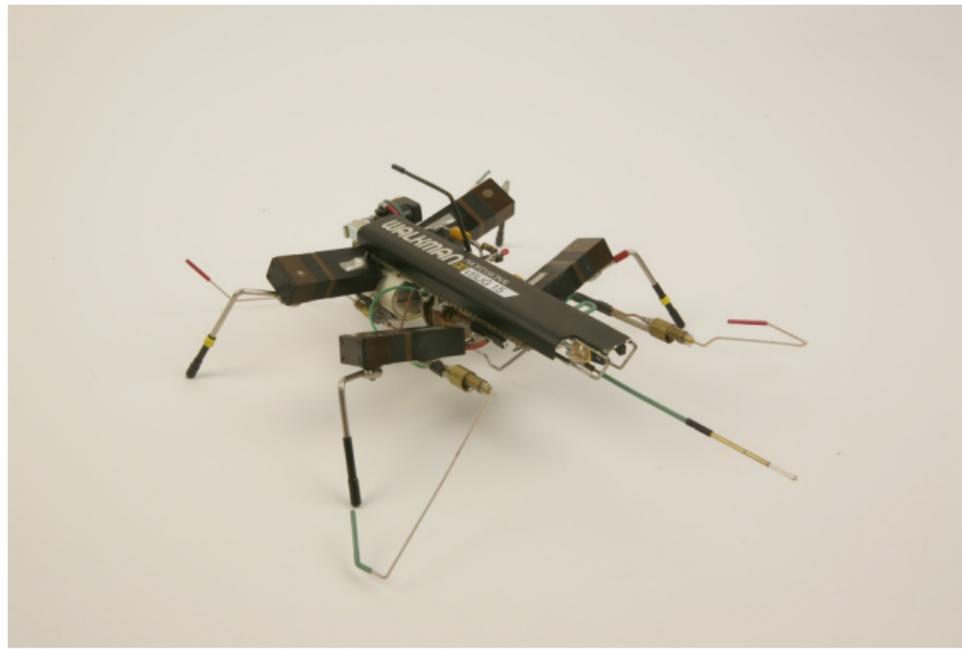
Base-10 Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ternary Digits: 0, 1, 2

Balanced Ternary Digits:  $\hat{1}$ , 0, 1

Base 10	10's Comp	Ternary	Balanced
6	6	20	$1\hat{1}0$
5	5	12	$1\hat{1}\hat{1}$
4	4	11	11
3	3	10	10
2	2	2	$1\hat{1}$
1	1	1	1
0	0	0	0
-1	999	222	$\hat{1}$
-2	998	221	$\hat{1}1$
-3	997	220	$\hat{1}0$
-4	996	212	$\hat{1}\hat{1}$
-5	995	211	$\hat{1}11$
-6	994	210	$\hat{1}10$

## BEAM (Analog) Robotics

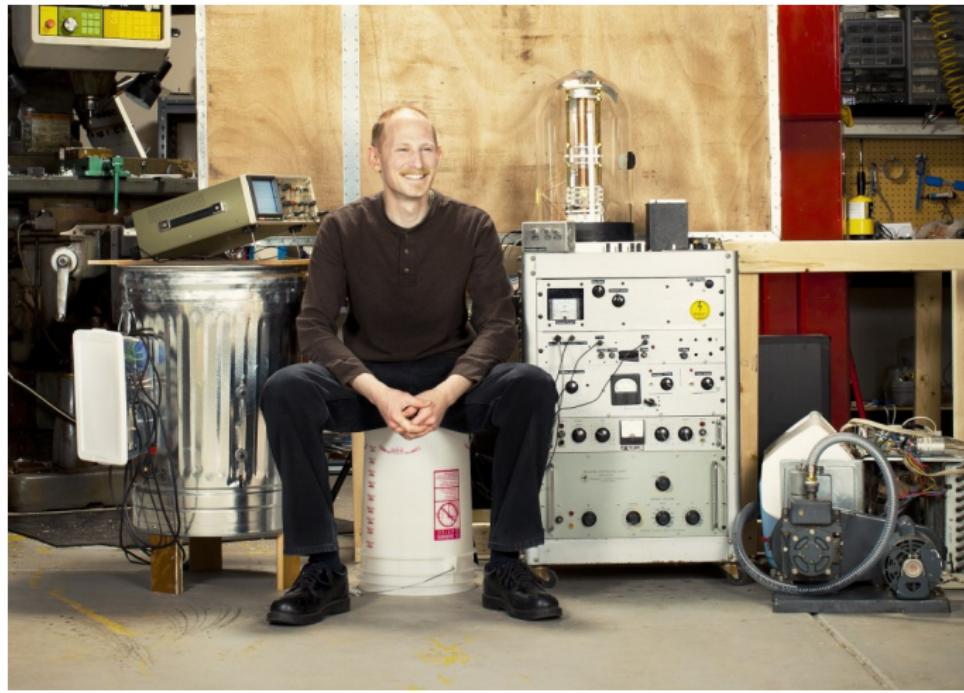


## Amateur Radio



# Electronics

## DYI Electron Microscopy



# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

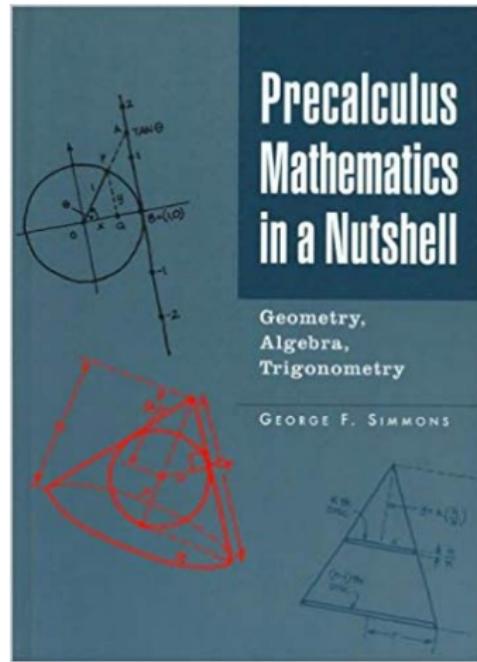
5 Nuclear Power!

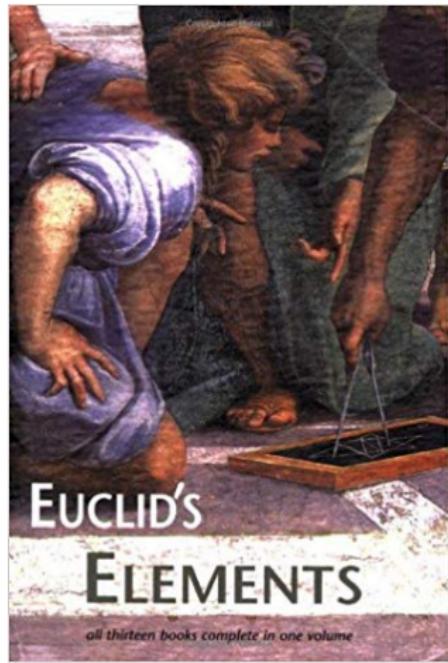
6 Electronics

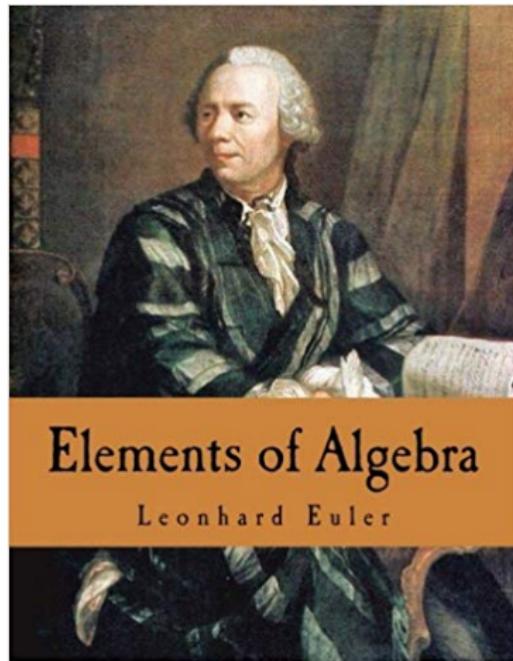
7 Mathematics

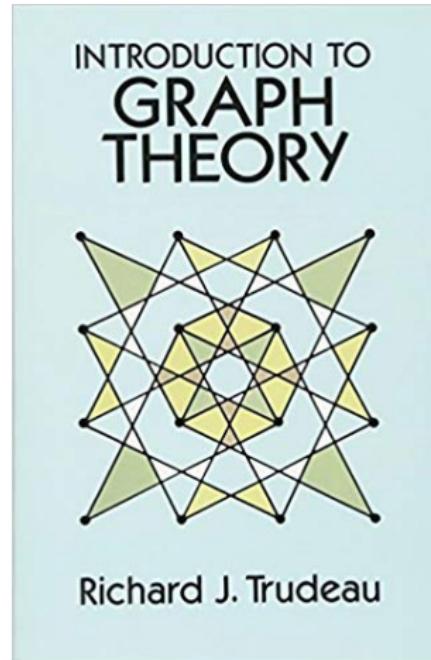
8 Questions

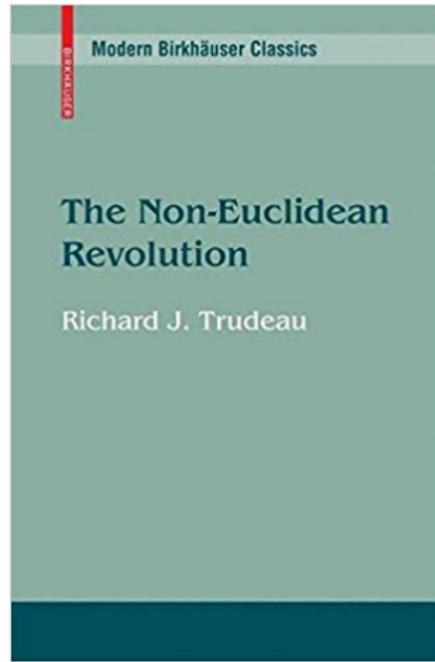
9 References









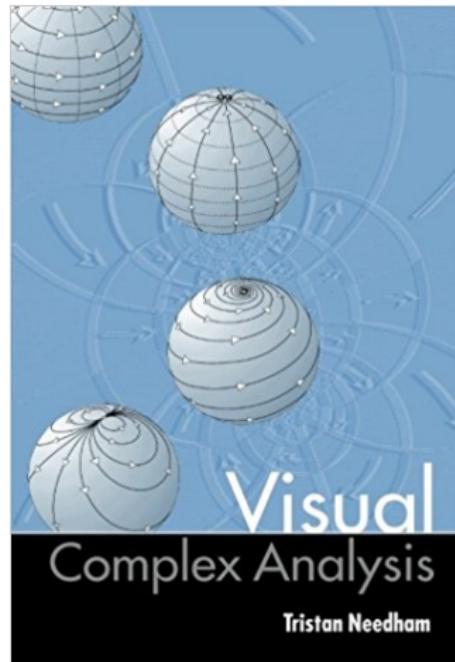


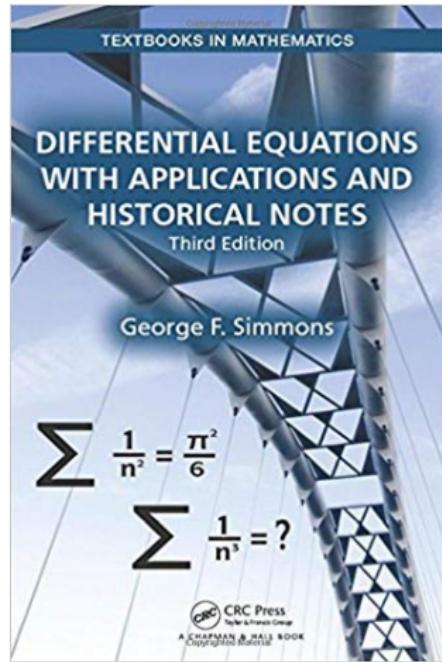
## Mathematics

W. W. SAWYER

# A CONCRETE APPROACH TO ABSTRACT ALGEBRA







# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

6 Electronics

7 Mathematics

8 Questions

9 References

# Questions

Slides:

<https://github.com/snowfarthing/slides/blob/master/2019-openwest-whirlwind-intro-to-alien-tech.pdf>

Any Questions?

# Outline

1 Keyboards

2 Alien Office

3 Build Your Own IDE

4 Alien Computing

5 Nuclear Power!

6 Electronics

7 Mathematics

8 Questions

9 References

# References

- Keyboards

*VT 100 (vi)*

<https://dave.cheney.net/2017/08/21/the-here-is-key>

*Space Cadet (emacs)*

[http://xahlee.info/kbd/space-cadet\\_keyboard.html](http://xahlee.info/kbd/space-cadet_keyboard.html)

*Dvorak*

<http://gigliwood.com/abcd>

<http://steve-lovelace.com/the-dvorak-keyboard>

*Programmer's Dvorak*

<https://www.kaufmann.no/roland/dvorak>

*MessagEase*

<https://www.exideas.com/ME/index.php>

*Janko Piano Keyboard*

<http://collections.nmmusd.org/Keyboards/JankoPiano/JankoPiano.html>

# References

- Alien Office

*LaTeX (and Beamer)*

<https://www.latex-project.org>

<https://www.amazon.com/LaTeX-Document-Preparation-System-2nd/dp/0201529831>

<https://people.orie.cornell.edu/jmd388/design/guides/Beamer%20Tutorial.pdf>

<http://www.maths.dundee.ac.uk/~dfg/icms09/beamer-art.pdf>

<https://www.lyx.org> (An WYSIWYG editor that internally uses LATEX)

*mdp*

<https://github.com/visit1985/mdp>

*Emacs SES*

[https://www.gnu.org/software/emacs/manual/html\\_node/ses/index.html](https://www.gnu.org/software/emacs/manual/html_node/ses/index.html)

*Emacs Org*

[https://orgmode.org/worg/org-tutorials/orgtutorial\\_dto.html](https://orgmode.org/worg/org-tutorials/orgtutorial_dto.html)

<https://orgmode.org/manual/index.html>

*Lily Pond*

<http://lilypond.org>

# References

- Build Your Own IDE

## *Bash/Zsh*

<https://github.com/ohmybash/oh-my-bash>

<https://ohmyz.sh>

<https://www.howtogeek.com/307701/how-to-customize-and-colorize-your-bash-prompt>

## *Dotfile Management*

<https://www.atlassian.com/git/tutorials/dotfiles>

<https://thoughtbot.com/upcase/videos/intro-to-dotfiles>

<https://dotfiles.github.io>

<https://github.com/webpro/awesome-dotfiles>

<https://medium.freecodecamp.org/dive-into-dotfiles-part-1-e4eb1003cff6>

<https://medium.freecodecamp.org/dive-into-dotfiles-part-2-6321b4a73608>

<http://blog.smalleycreative.com/tutorials/using-git-and-github-to-manage-your-dotfiles>

## *Tmux/GNU Screen*

<https://www.hamvoeke.com/blog/a-quick-and-easy-guide-to-tmux>

<https://hackernoon.com/a-gentle-introduction-to-tmux-8d784c404340>

<https://www.nixtutor.com/linux/introduction-to-gnu-screen>

# References

- Build Your Own IDE (continued)

*Emacs* (Spreadsheet, Org, Tetris, EVIL)

<https://www.gnu.org/software/emacs>

<https://www.gnu.org/software/emacs/tour>

<https://github.com/emacs-tw/awesome-emacs>

<http://xmodulo.com/best-plugins-to-increase-productivity-on-emacs.html>

<https://spin.atomicobject.com/2014/07/06/best-emacs-addons>

*Vim* (and plugins)

<https://www.vim.org>

<https://danielmiessler.com/study/vim>

<https://github.com/akruchy/awesome-vim>

<https://medium.com/@huntee/10-essential-vim-plugins-for-2018-39957190b7a9>

<https://spf13.com/post/the-15-best-vim-plugins>

*MySQL pager (colorized!)*

<http://manpages.ubuntu.com/manpages/xenial/man1/grc1.1.html>

<https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-using-pager.html>

*SL*

<https://github.com/mtoyoda/sl>

# References

- Nuclear Power

*Molten Salt Fission*

[https://en.wikipedia.org/wiki/Liquid\\_fluoride\\_thorium\\_reactor](https://en.wikipedia.org/wiki/Liquid_fluoride_thorium_reactor)

<https://energyfromthorium.com>

<https://liquidfluoridethoriumreactor.glerner.com>

*Home-made Fusion via the Fusor*

<https://en.wikipedia.org/wiki/Fusor>

<http://www.fusor.net>

*PolyWell Fusion*

<https://en.wikipedia.org/wiki/Polywell>

<https://www.ialtenergy.com/polywell-fusion.html>

*Orion Project*

[https://en.wikipedia.org/wiki/Project\\_Orion\\_\(nuclear\\_propulsion\)](https://en.wikipedia.org/wiki/Project_Orion_(nuclear_propulsion))

<http://www.unmuseum.org/orionproject.htm>

# References

- Alien Computing

*Cryptography*

<https://en.wikipedia.org/wiki/Cryptography>

<https://learn cryptography.com>

\*\*\* CONFOUNDED IT: WHERE'S A GOOD TUTORIAL FOR WHAT YOU CAN DO  
WITH BASIC ELEMENTS?!?

<https://en.wikipedia.org/wiki/Cryptography>

<https://www.cs.auckland.ac.nz/~pgut001/tutorial>

<https://www.garykessler.net/library/crypto.html>

<https://www.cs.unibo.it/babaoglu/courses/security/resources/documents/intro-to-crypto.pdf>

[https://crypto.stanford.edu/~dabo/cryptobook/draft\\_0\\_2.pdf](https://crypto.stanford.edu/~dabo/cryptobook/draft_0_2.pdf)

<https://www.cryptool.org/images/ctp/documents/CT-Book-en.pdf>

<https://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf>

*RISC-V Architecture*

<https://riscv.org>

*Nix OS*

<https://nixos.org>

# References

- Alien Computing (continued)

## *Forth*

<https://www.amazon.com/Starting-Forth-Introduction-Operating-Professionals/dp/0138429227>  
<https://www.amazon.com/Thinking-Forth-Leo-Brodie/dp/0976458705>  
<https://playground.arduino.cc/CommonTopics/ForthOnArduino>

## *Prolog*

<https://www.cis.upenn.edu/~matuszek/Concise%20Guides/Concise%20Prolog.html>  
<http://www.cs.toronto.edu/~sheila/384/w11/simple-prolog-examples.html>  
[https://www.cpp.edu/~jrfisher/www/prolog\\_tutorial/contents.html](https://www.cpp.edu/~jrfisher/www/prolog_tutorial/contents.html)  
<http://www.learnprolognow.org>  
[https://www.doc.gold.ac.uk/~mas02gw/prolog\\_tutorial/prologpages](https://www.doc.gold.ac.uk/~mas02gw/prolog_tutorial/prologpages)  
<https://learnxinyminutes.com/docs/prolog>

## *J*

<http://www.jsoftware.com>  
<http://www.jsoftware.com/help/learning/contents.htm>  
<http://www.jsoftware.com/books/pdf/expmath.pdf>  
<http://www.jsoftware.com/books/pdf/arithmetic.pdf>  
<http://www.jsoftware.com/books/pdf/calculus.pdf>

## *Smalltalk/Pharo*

<http://wiki.c2.com/?SmalltalkExamples>  
<http://www.jera.com/techinfo/readingSmalltalk.pdf>  
<http://sdmeta.gforge.inria.fr/FreeBooks/Art/artAdded174186187Final.pdf>  
<https://medium.com/learn-how-to-program/latest>  
<https://amber-lang.net/learn.html>  
<http://books.pharo.org/updated-pharo-by-example/pdf/2018-09-29-UpdatedPharoByExample.pdf>  
<http://sdmeta.gforge.inria.fr/FreeBooks/ByExample>

# References

- Alien Computing (continued)

## *Haskell*

<http://learnyouahaskell.com> (Learn You a Haskell for Great Good)  
<https://www.haskell.org/tutorial/index.html> (A Gentle Introduction)  
<http://book.realworldhaskell.org> (Real World Haskell)  
<https://hackage.haskell.org>  
<https://hoogle.haskell.org>  
<https://vaibhavsagar.com/blog/2016/10/12/monad-anti-tutorial>  
<https://unknownparallel.wordpress.com/zero-analogy-monad-tutorial>  
<https://byorgey.wordpress.com/2009/01/12/abstraction-intuition-and-the-monad-tutorial-fallacy>  
<https://ncatlab.org/nlab/files/WadlerMonads.pdf>  
<http://web.cecs.pdx.edu/~mpj/pubs/RR-1004.pdf> (Composing Monads)  
<http://math.mit.edu/~dspivak/teaching/sp18/7Sketches.pdf>  
    (Seven Sketches of Composability)  
[http://adit.io/posts/2013-04-17-functors,\\_applicatives,\\_and\\_monads\\_in\\_pictures.html](http://adit.io/posts/2013-04-17-functors,_applicatives,_and_monads_in_pictures.html)

## *Erlang*

<https://learnyousomeerlang.com/introduction>  
<https://www.brianstorti.com/the-actor-model>

# References

- Alien Computing (continued)

## *E and EROS*

<http://www.skyhunter.com/marcs/ewalnut.html> (E in a Walnut)  
<https://www.crockford.com/ec/dining.html> (Satan Comes to Dinner)  
[ftp://ftp.cis.upenn.edu/pub/eros/public\\_html/eros.html](ftp://ftp.cis.upenn.edu/pub/eros/public_html/eros.html)  
<http://habitatchronicles.com/2017/05/what-are-capabilities>

## *Rholang*

<https://www.sciencedirect.com/science/article/pii/S1571066105051893>  
<https://github.com/rchain/rchain/blob/master/docs/rholang/rholangtut.md>  
<https://github.com/JoshOrndorff/LearnRholangByExample>

## *Common Lisp/Scheme*

[https://en.wikipedia.org/wiki/Lambda\\_calculus](https://en.wikipedia.org/wiki/Lambda_calculus)  
<http://www.defmacro.org/ramblings/lisp.html>  
<http://www.gigamonkeys.com/book> (Practical Common Lisp)  
<https://mitpress.mit.edu/sites/default/files/sicp/index.html>  
    (Structure and Interpretation of Computer Programs)  
<http://www.paulgraham.com/onlisp.html>  
<https://letoverlambda.com>  
<https://mitpress.mit.edu/books/art-metaobject-protocol>  
<http://landoflisp.com>

# References

- Electronics

*From NAND to Tetris*

<https://www.nand2tetris.org>  
<http://dwscope.freeshell.org/projects>

*Ternary Computing/Balanced Ternary*

<https://earlcampbell.com/2014/12/29/the-setun-computer>  
[https://en.wikipedia.org/wiki/Balanced\\_ternary](https://en.wikipedia.org/wiki/Balanced_ternary)  
<https://www.youtube.com/watch?v=a2nrCrNDAHk>  
<https://hackaday.com/2016/12/16/building-the-first-ternary-microprocessor>  
<https://hackaday.io/project/6284-tern-ternary-logic-circuits>  
<http://www.cis.upenn.edu/~matuszek/cit590-2015/Pages/balanced-ternary.html>

*BEAM Robotics*

<https://solarbotics.com/beam>  
<https://robohub.org/robots-beam-robotics>  
<http://solarbotics.net/bftgu/beam.html>  
<https://www.element14.com/community/blogs/SalsCorner/2015/10/26/beam-robotics-a-blast-from-the-past>  
[https://en.wikipedia.org/wiki/BEAM\\_robots](https://en.wikipedia.org/wiki/BEAM_robots)  
<http://grant.solarbotics.net/walkman.htm>

# References

- Electronics (continued)

## *Amateur Radio*

<http://www.arrl.org/what-is-ham-radio>

[https://tntacticalsupply.com/free-instructions-for-a-cheap-antenna  
-to-boost-the-range-of-your-baofeng-or-other-brand-of-2-meter-ham-radio](https://tntacticalsupply.com/free-instructions-for-a-cheap-antenna-to-boost-the-range-of-your-baofeng-or-other-brand-of-2-meter-ham-radio)

## *Electron Microscopes*

<https://makezine.com/2012/07/24/my-scanning-electron-microscope>

<https://hackaday.io/project/21831-3d-printed-scanning-electron-microscope>

<https://carlwillis.wordpress.com/2008/02/20/uranium-chemistry>

# References

- Mathematics

*Precalculus mathematics in a nutshell: Geometry, algebra, trigonometry*

<https://www.amazon.com/Precalculus-mathematics-nutshell-Geometry-trigonometry/dp/0760706603>

*Euclid's Elements/The Bones*

<https://www.amazon.com/Euclids-Elements-Euclid/dp/1888009195>

<https://www.amazon.com/Bones-Kivar-waterproof-soft/dp/1888009217>

<https://www.amazon.com/Works-Archimedes-Dover-Books-Mathematics/dp/0486420841>

<https://www.amazon.com/Conics-Books-I-IV-Apollonius-Perga/dp/1888009411>

*Euler's Elements of Algebra*

<https://www.amazon.com/Elements-Algebra-Leonhard-Euler/dp/150890118X>

*Introduction to Graph Theory (aka Dots and Lines)*

<https://www.amazon.com/gp/product/0486678709>

*The Non-Euclidian Revolution*

<https://www.amazon.com/Non-Euclidean-Revolution-Richard-J-Trudeau/dp/0817633111>

*A Concrete Approach to Abstract Algebra*

<https://www.amazon.com/Concrete-Approach-Abstract-Algebra/dp/B0037EYL7Y>

[\(Prelude to Mathematics\)](https://www.amazon.com/gp/product/B00A73AGF0)

[\(Mathematician's Delight\)](https://www.amazon.com/gp/product/0486462404)

# References

- Mathematics (continued)

*Visual Complex Analysis*

<https://www.amazon.com/Visual-Complex-Analysis-Tristan-Needham/dp/0198534469>

*Differential Equations with Applications and Historical Notes*

<https://www.amazon.com/gp/product/B01N9111RS>

<http://math.mit.edu/directory/profile.php?pid=175> (Arthur Mattuck – MIT Lectures)

<https://www.youtube.com/watch?v=XDhJ81VGbl8&list=PL59B01CDD80573FDO>

<https://ocw.mit.edu/courses/mathematics/18-03-differential-equations-spring-2010>