

一年生向け講義

資料は

<http://cohalz.github.io/slide/14rdlecture.pdf>

自己紹介

- ID: @cohalz (Twitter ・ Skype)

自己紹介

- ID: @cohalz (Twitter ・ Skype)
- 主コンピュータサイエンス、副数理情報学コースのRD2年生

自己紹介

- ID: @cohalz (Twitter ・ Skype)
- 主コンピュータサイエンス、副数理情報学コースのRD2年生
- ゲームが作りたくてこの大学に入学。

自己紹介

- ID: @cohalz (Twitter ・ Skype)
- 主コンピュータサイエンス、副数理情報学コースのRD2年生
- ゲームが作りたくてこの大学に入学。
- しかし、入学してまもなくプログラミングと数学、さらに大学という空間が好きになり大学教員を目指しはじめた。

自己紹介

- 高1でプログラミングを始めたが何をすればいいのかわからなかったので、大してやっていたわけではない

自己紹介

- 高1でプログラミングを始めたが何をすればいいのかわからなかったので、大してやっていたわけではない
- 「Scalaで始めるプログラミング入門」というものを書いてます

自己紹介

- 高1でプログラミングを始めたが何をすればいいのかわからなかったなので、大してやっていたわけではない
- 「Scalaで始めるプログラミング入門」というものを書いています
- 去年の鳩山祭にiOS用ライツアウトゲームを展示しました

自己紹介

- 高1でプログラミングを始めたが何をすればいいのかわからなかったので、大してやっていたわけではない
- 「Scalaで始めるプログラミング入門」というものを書いてます
- 去年の鳩山祭にiOS用ライツアウトゲームを展示しました
- いろいろなプログラミング言語を見ていくのが好き
ScalaやHaskell、OCamlなどが気に入っている

自己紹介

- 高1でプログラミングを始めたが何をすればいいのかわからなかったので、大してやっていたわけではない
- 「Scalaで始めるプログラミング入門」というものを書いてます
- 去年の鳩山祭にiOS用ライツアウトゲームを展示しました
- いろいろなプログラミング言語を見ていくのが好き
ScalaやHaskell、OCamlなどが気に入っている

プログラミングの話

の前に

エディタの話

何を使ってプログラムを書いていますか？

えーマジ

メモ帳

!?

メモ帳が許されるのは
小学生までだよなー

キモイ

キヤハハハハ



メモ帳使っているプログラマなんて
存在しない

「弘法は筆を選ばず」は
絶対に信じてはいけない

偉大な人が作った

プログラミング環境を利用しよう

- テキストエディタ
- 統合開発環境(IDE)

テキストエディタ

- Vim
- Emacs
- Sublime Text

統合開発環境(IDE)

- Visual Studio
- Eclipse
- IntelliJ IDEA

VimかEmacsの操作に慣れる

プログラミングの話

どんなプログラミング言語がある？

どんな区分がある？

C ・ Scala ・ Ruby ・ Haskell ・ JavaScript

はどこに入る？

型付け

- 静的型付け

C ・ Scala ・ Haskell

- 動的型付け

Ruby ・ JavaScript

型の強さ

- 強い型付け

Scala ・ Ruby ・ Haskell

- 弱い型付け

C ・ JavaScript

ソースコード中の型

- 書く

C

- 書かないがコンパイル時に型が推論される(型推論)

Scala ・ Haskell

- 書かないが実行時に決まる

Ruby ・ JavaScript

正格性

- 正格言語

C · Scala · Ruby · JavaScript

- 非正格言語

Haskell

基本パラダイム

- オブジェクト指向

Scala ・ Ruby ・ JavaScript

- 関数型プログラミング

Haskell (Scala ・ Ruby ・ JavaScript)

現在使われている言語とその書籍の 紹介

C

- UNIXを書くために作られた言語
- 現在でもOSやマイコン、プログラミング言語などに使われる
- 記述性と効率のバランスに優れているが、古臭い
- 『Head First C』や『C言語総合講座15講 ~理論編~(無料)』など

C++

- Cを改良しようとして生まれた言語
- 現在でも広く使われ、仕様も新しくなっている
- プログラミング言語界の生きた怪物
- 『ロベールのC++入門講座』や
『C++ ポケットリファレンス』 など

Java

- どこでも動くことを利点に作られた言語
- オブジェクト指向を広めた言語
- 現在では日本のSEなどでよく使われている
- 『スッキリわかるJava入門』や
『スッキリわかるJava入門 実践編』 など

Scala

- Java仮想マシン上で動く、Javaを改良したような言語
- Javaの資産、オブジェクト指向、関数型言語の書き方を利用することが可能
- 基本的にJavaの上位互換だが、コンパイル時間が長いのが欠点
- 『Scalaスケーラブルプログラミング第2版』や
『Guide to Scala—Scalaプログラミング入門』など

Haskell

- 唯一の有名な純粋関数型プログラミング言語
- 数学的構造や強い型付けを使いバグを出させない
- そのためコアなファンが他の言語に比べても多い
- 『すごいHaskellたのしく学ぼう！』や
『Real World Haskell』 など

OCaml

- Haskellについて人気のある関数型プログラミング言語(非純粋)
- 非常に強い型付けと型付けにより品の良いプログラムが書ける
- オブジェクト指向も採用している
- 『プログラミング in OCaml』 や 『プログラミングの基礎』 など

Ruby

- 日本人が作った言語であり、「enjoy programming」がモットー
- Ruby on Railsやmikutterなどが有名
- Lisp由来の柔軟性と、Smalltalk由来のオブジェクト指向が強力
- 『たのしいRuby』や『プログラミング言語 Ruby』など

Python

- Rubyと同じ強い動的型付け言語
- オブジェクト指向言語
- Rubyとは違い、規則正しいプログラムを強制している
- 『みんなのPython』や『初めてのPython』など

JavaScript

- Webで動く事実上の標準言語
- ただし、奇妙な言語仕様と弱い安全でない動的型付け
- そのためaltJSというJavaScriptを置き換える動きが活発
- 『JavaScript(オライリー)』や『JavaScript本格入門』など

Lisp

- COBOL、FORTRANとほぼ同時期に生まれた言語
- if、関数型、再帰、動的型付け、ガベージコレクション、継続などがLispから生まれている
- 色々なLispがあり、有名なのはCommon Lisp、Scheme、Clojure
- Clojureは2007年に生まれたJava動くLispのため期待されている
- 『素数夜曲』や『Land of Lisp』など

情報系大学生としてやるべきこと

情報系大学生としてやるべきこと

- 自分から動く、自分でなんでも調べる
- 授業で使っていない言語を何か一つやる
- UNIX系のOSを使う
- GitHubを使う
- 勉強会に行く
- いろんな本をたくさん読む

自分から動く、自分でなんでも調べる

- 大学は教育機関ではない、まして就職のための課程でもない
- やりたいこと、気になることを「自分で」見つけて研究しよう
- ただ金を搾取されることのないよう、先輩や教員も
どんどん使って自分のやりたいことを進めよう
- それができる人間は研究も就職も強い

『多くの、いやひょっとしてほとんどの
ハッカーたちは我流で勉強してきたのです。

役に立つのは、コードを読むこと、

そしてコードを書くことです。』

ハッカーになろう(<http://cruel.org/freeware/hacker.html>) より

集中して学べるのは今だけ

- 歳を取ると、やる時間も集中力もなくなっていく
- 働けば大学図書館も使えなくなる
- 遊ぶことは後になってもできる

授業で使っていない言語を何か一つやる

- 英語を知って日本語の特徴を知るようなもの
- 好きな言語を見つけるためにいろいろ見ていくべき
- 授業でやるプログラミングは、就職に役立つわけではない
- 最低でも一つだが、二つ以上やるとなおよい

GitHubを使う

- ソフトウェア開発のためのサイト
- オープンソースソフトウェアの標準となっている
- バージョン管理
(過去の状態に戻せたり、beta版と安定版などで分けたりなど)
保存と公開のためだけに使うもよし
- GitHub Educationという学生のためのプランがある
- GitHubを活用しているかどうかが選考基準になる企業もある

他人のソースコードを読むということ

- 『プログラミングで成功するには他のプログラマーと話をし、他人のプログラムを読むこと』 - Peter Norvig
- 期限の終わった課題など、GitHubに公開したり、見せ合う時間を設けたりして他人とソースコードの交流を行う
- 他人のコードを多く読む(サンプルコードの多い書籍を含む)
ことでしか成長することはできない

UNIX系のOSを使う

- Windowsはプログラミングのための環境も揃ってない
- それに加えゲームをやってしまうため良くない
- たいていのソフトウェアはLinux対応しているので、
困ることは殆ど無い

勉強会に行く

- 堅苦しく聞こえるけど実は交流の場
Twitter開きながら話を聴いて、面白いものはTweetする
- 有名な人の面白い話を有名なオフィスで聴いたあと、
一緒にどこか食べに行く事が多い
- プレゼン発表の練習にもなる

本を読む

- 小説ではないのでどんな読み方でもいい
- 「講義は本を読む時の手助けでしかない」という言葉の通り、本を読まなければ大学行っている意味はほとんど無い
- 「その本、図書館にあります」というChrome拡張が便利
- Amazon Studentに登録すれば本が10%ポイント還元

<http://amzn.to/1wZ5M4z>

おすすめ書籍

/ THEORY / IN / PRACTICE

リーダブルコード

より良いコードを書くための
シンプルで実践的なテクニック

O'REILLY®
オライリー・ジャパン

Dustin Boswell 著
Trevor Foucher 角 征典 訳

『リーダブルコード』

- 理解しやすい読みやすいコードを書くための本
- ジュンク堂池袋本店コンピュータ書籍売上ランキングを二年連続一位
- 変数名の改善からコード群の改善まで幅広い

プログラミング作法

↕ THE PRACTICE OF PROGRAMMING

Brian W. Kernighan
Rob Pike
福崎 俊博 訳

PROGRAMMING

ASCII

『プログラミング作法』

- K&Rの二人が書いたプログラムの書き方に関する本
- 本当の意味でのK&Rスタイルを学ぶことができる
- リーダブルコードと同じく非常に人気のある本

D is for Digital
What a well-informed person should know about
computers and communications

ディジタル作法

カーニハン先生の「情報」教室

Brian W. Kernighan 著
久野 靖 訳

『ディジタル作法』

- K&Rのカーニハンが専門でない一般の人向けに書いた本
- すべての人が知っておくべきIT教養を解説した本
- ハードウェア・ソフトウェア・ネットワークの概要を知ることができる

Hackers and Painters: Big Ideas from the Computer Age

ハッカーと画家

コンピュータ時代の創造者たち

Paul Graham 著 / 川合史朗 監訳

『ハッカーと画家』

- インターネット黎明期にWebアプリを開発した人によるハッカーエッセイ
- 数式やプログラムはほとんど載っていないので気軽に読める
- ハッカーとは結局何なのか？
実はハッカーと画家は似ている？
学生時代蔑まれたオタクがなぜ世界を動かした？
その答えが書いてある本

普通のやつらの上を行け

天才LISPハッカーにして、ソフトウェアベンチャーを成功させた筆者が、ものづくりのセンス、成功するビジネスの秘密、発想を刺激するアイデアを語る

OHM
Ohrsha

入門 コンピュータ科学

ITを支える技術と理論の基礎知識

J. Glenn Brookshear 著
神林 靖、長尾高弘 訳

UCバークレイ、
ハーバード大学を
はじめ全米126校で
採用され続ける
定番教科書！



『入門コンピュータ科学』

- コンピュータ科学の様々な分野を一冊にまとめた本
- まとまっているので専門性は低いが、分野が広いので持っていて困らない
- 練習問題とその解答も豊富で、一人で読むにも複数人で一緒にやるのにも向いている

数学ガール

Mathematical Girls



結城 浩

Hiroshi Yuki

SB Creative

『数学ガール』

- 数学解説と青春の物語
- 数学が楽しく思えるようになる本なので、専門書に行く前に目を通すべき本
- 数学が理解できなくても話が面白いので読み通せてしまう本



『数学文章作法 基礎編』

- 数学ガールの著者が書いた、わかりやすい文章を書くための本
- この内容はレポートなどにも活用することができる
- 近いうちに推敲編も出る

『素数夜曲』

- 整数論とLisp(Scheme)に関する本
- 大著であるが、数式を詳しく書いてあるので難しいわけではない
- 数学ガールの著者が書き方を参考にした人の本なので、数学ガールが好きな人に特におすすめ

素数夜曲

女王陛下のLISP

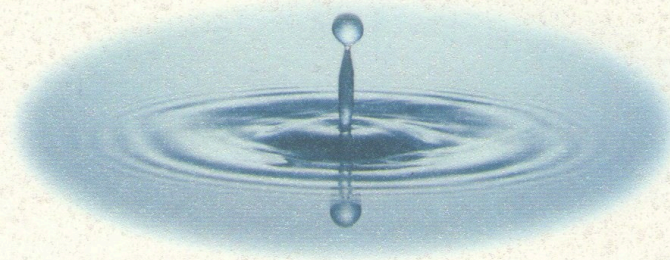
吉田武 著

『虚数の情緒』

- 素数夜曲と同じ人が書いた本だが、主に数学と物理の本
- 20世紀最後の大著であり、これほどまでに野望のある本は見たことがない
- 中学生以上のすべての人が知らなければいけない一冊

虚数の情緒

中学生からの全方位独学法



吉田 武 著

おすすめサイト

『AtCoder』

<http://atcoder.jp/>

- 与えられた問題をプログラムを書いて解いていく
競技プログラミングをすることができるサイト
- 非常に簡単な問題から始まっているのでプログラミングの練習
に最適

『ドットインストール』

<http://dotinstall.com/>

- プログラミング言語やライブラリフレームワークなどの基本を動画で学べる
- 一回一回が短いので気楽に見ることができる

『プログラマの為の数学勉強会』

[https://www.youtube.com/watch?
v=2rd6d41dHc&list=PLzJWjr7AvxH0YYpi2uAH
QHLaSJQ5fZrR](https://www.youtube.com/watch?v=2rd6d41dHc&list=PLzJWjr7AvxH0YYpi2uAHQHLaSJQ5fZrR)

- 数学がどのようにプログラムに応用されているのか眺めることができる
- グラフィカルな説明も多いので内容を理解するのが非常に楽

『東北大教授による大学新入生への助言』

<http://togetter.com/li/285298>

- 大学に入ってどうすればいいのか、
教員の意見を述べたTweetをまとめたもの
- 一ヶ月おきくらいで見ると目的を見失わずに済むかも

『IT勉強会カレンダー』

<http://bit.ly/1nU7azL>

- 開催されている勉強会の日程を知ることができる
- キーワードで検索したい場合は以下から

<http://utf-8.jp/tool/calsearch.html>

その他文献

『ハッカーになろう』

<http://cruel.org/freeware/hacker.html>

『プログラミングを独習するには10年かかる』

<http://www.yamdas.org/column/technique/21-daysj.html>

『情報系の道に悩める君へ』

http://d.hatena.ne.jp/syonbori_tech/20130827/1377605684

ソフトウェア

- Dropbox
ファイルのバックアップ、複数PCでの共有
登録は <http://bit.ly/1qZCQZJ> から
- Google日本語入力
標準の日本語入力よりも賢い
- Microsoft OneNote
デジタルノートアプリケーション
ノートとしてはWordなどをはるかに凌ぐ使いやすさ

Dropboxでも過去のバージョンに戻すことは できる

- 「以前のバージョンを表示」から過去のコードが見れる
- 一ヶ月間のみ、更にどこが変更されたのか見ることはできない
- プログラムのバージョン管理はおとなしくGitHubを使うべき

Macアプリケーション

- BetterTouchTool
トラックパッドでショートカットキーのような動作を設定する
- Go2Shell
ボタンひとつでFinderのディレクトリからターミナルを開く
- KeyRemap4MacBook & Seil
キーの動作を変更する(caps lockをctrlに変更など)
- PopClip
選択したものをコピーや検索などを簡単に行える