

概要

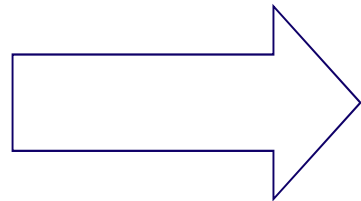
到達目標

内容

- ・ プログラムの**初心者**向け
- ・ **自宅等での独学では味あえないような体験/
対面学習を重視**
- ・ 講義の理解に必要な資料は、すべてウェブ
ページで公開

基本方針

知識よりも



体験を重視

- 数多くの練習を行い、
確実に理解しながら進む
- 「手本」を真似ることから
始める

到達目標 1

プログラミングの楽しさを実感

- プログラムを, 自分の意図通りに動かす.
 - 好奇心を満足する
- 講義と異なる「インタラクティブ性」
- 知的探求

到達目標 2

プログラムの実行イメージを持つ

- 「プログラムの指示に従って、コンピュータがどのように振る舞うか」というイメージを、頭の中で描けること
- 他の人が作ったプログラムを見て、意味が理解できること

この講義で身に付けて欲しいこと

- ① Scheme の式, 括弧の付け方, 関数の書き方に十分慣れているか
- ② 再帰について, よく理解しているか
 - プログラムを読み解く能力. プログラムを自分で(一から)作成できる能力.
- ③ 構造体 について
- ④ list を扱う再帰のプログラム
- ⑤ 再帰を使った実際的なプログラム (階乗, テーラー展開, 折れ線, 多角形・・・など) と高階関数, #i 表示
- ⑥ 「プログラムの作成手順」, 「エラー」を, 深く理解しているか

Scheme についての参考資料

- ホームページ

<http://www.cs.rice.edu/CS/PLT/Teaching/Lectures/Released/Book>

- 参考書

- M. Felleisen et al., How to Design Programs, MIT Press (<http://www.htdp.org/>)
- ケント ディヴィグ, プログラミング言語SCHEME, ピアソン・エデュケーション
- エイブルソンら, 計算機プログラムの構造と解釈, ピアソン・エデュケーション

授業内容と指導方針

授業の進め方

前半
(20分程度)

講義

基本事項
の理解

授業を聴き、
資料を読んで
理解する

後半
(70分程度)

実習

例題を実際に
実行して、
理解を深める

授業時間中
に、各自で行う

課題

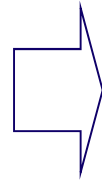
課題を解き
ながら、
プログラムに
慣れる

講師と
ティーチング
アシスタント
による巡回指導

講義では

コンピュータの
ソフトを知る

コンピュータでの
「計算」とは？
「情報」とは？



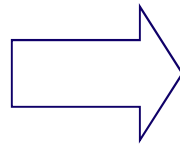
コンピュータの専門家
としての**素養**の取得

コンピュータの振る舞い
の理解

問題意識と問題解決能力
への最初のステップ

実習／課題では

プログラムの
実体験を積む



プログラムを
読む, 動かす, 書き換える

プログラムの振る舞い
を体験で理解する

プログラムに慣れる

「正しい」プログラム
を書けるようになる

課題について

- 課題は、さまざまな難易度の問題を数多く提示する
 - 自分の実力にあった問題を**選択**して解くこと
 - 全問を解くことは求めない(採点では、「数」よりも、「**質**」を重視して採点する)。
- 確実に理解しながら、自分の力で解くことが重要
- 分からないことがあれば、遠慮なく質問して欲しい

実習で心がけて欲しいこと

- 各自が、自分の力で練習に取り組む
 1. 「手本」となるプログラムを良く読み、理解する
 2. 「手本」をまねて、自分でプログラムを作ってみる.
 3. 動かしてみる
 4. 動いたら、自分の納得のいくまで手を加える
 5. 動かないようであれば、解決の糸口を自分で探してみる（失敗を重ねながら、理解を深める）

分からないことは、遠慮なく質問して欲しい