

# sp-1. 全体内容と方針

(Scheme プログラミング)

URL: <https://www.kkaneko.jp/pro/scheme/index.html>

金子邦彦



- プログラムの初心者向け
- 演習，体験を重視
- 必要な資料は，すべてウェブページで公開

## 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

## 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

# Scheme プログラミングの実践スキル



- ① Scheme の式, 括弧の付け方, 関数の書き方
- ② 再帰
- ③ リスト
- ④ 構造体
- ⑤ プログラムの読解, 作成能力  
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)  
高階関数, #i 表示
- ⑥ エラーの無いプログラムの作成能力  
プログラムの作成手順, エラー

# 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

# 資料の構成

# 資料の構成

前半

説明資料

基本事項  
の理解

パソコン  
演習

例題を実際に  
実行して、  
理解を深める

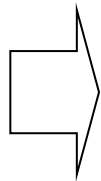
後半

課題

課題を解き  
ながら、  
プログラムに  
慣れる



コンピュータの  
ソフトを知る



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

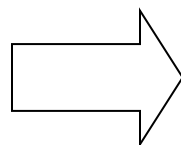
コンピュータでの  
「計算」とは？  
「情報」とは？

問題意識と問題解決能力  
への最初のステップ

# パソコン演習では



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

- 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)

- 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