

プログラミング言語 第一回

担当: 篠沢 佳久
櫻井 彰人

平成20年度: 春学期

1

クラス分け

2

クラス分け①

- プログラミング言語は二つの教室で同時に行います
 - 703教室(50人収容)
 - 704教室(100人収容)
- どちらの教室も同じ内容の講義をします

3

クラス分け②

- 703
 - 60707310までの学籍番号の学生
- 704
 - 60707755以降の学籍番号の学生
 - 管理工学科2年生以外の学生

4

講義のガイダンス

講義の目的, 進め方

5

この講義の目指すもの Part1

- プログラミングの基礎を理解
 - プログラミングの基礎知識を中心に
 - プログラムとは
 - プログラムの実行とは
 - 命令とデータ
 - 判断と分岐
 - プログラミングの構造と実行制御

6

この講義の目指すもの Part2

- プログラミングという行為
 - 書く、テストする、使う
- プログラミングがひとりのできるように
 - アルゴリズム
 - データ構造
- プログラミング言語とは
- プログラミングの基本をプログラム言語Rubyを通して学ぶ
- 後半はJava言語も行なう

7

この講義の目指すもの Part3

- Ruby言語でプログラムが作れるように
 - 基本的な演算
 - 選択
 - 繰り返し
 - 配列
 - 関数
 - ファイル入出力・文字列操作

8

Rubyとは何か?



- Ruby: まつもとゆきひろ氏による、便利さと容易さを兼ね備えたオブジェクト指向スクリプト言語
 - まつもとゆきひろ: <http://www.rubyist.net/~matz/>
 - スクリプト言語: 動作内容を、台本 (Script) のように記述するための、簡易的なプログラミング言語の総称
 - かなり簡単に (周辺環境が) インストールできる
 - 皆さんのコンピュータで実習ができる
 - かなり簡単にプログラムできる
 - 初心者にも容易に学習できる
 - 結構まともに動くプログラムも書ける
 - Ruby on Rails により、Webアプリが容易に書ける
 - そして、Ruby が有名になった

9

この講義では

- 演習をできる限り行います。
 - そのためには、実は、Ruby プログラムを実行するシステムとして irb をよく用います。
- irb (interactive Ruby) は対話的に Ruby プログラムを実行するもので、ちょっと実習をするには、適しているのです。

10

この講義で目指せたら

- もう少し先に行くと
 - ファイル処理
 - DB処理
 - Web アプリケーション
 - 日本語処理

11

内容に関する注意

- 基本的 (初歩的) なことに注力する
- ただし、ところどころ細かい話もする
 - 少し深いことを知りたい方への追加
 - 疑問に対する答えとして
 - 初級者は無視をしてよい

12



Javaとは何者か？

- ジャワコーヒーのようですが、あるときから、あるコンピュータ言語の名前になりました
 - その経緯には、技術開発、想定するビジネスモデルの変転がみてとれる
 - それはともかくとして、
- Sun Microsystems が開発した言語
- プログラム開発環境(コンパイラ等)を無料で公開している

13

進め方:

- (繰り返しになりますが) Ruby を使う
 - 特に irb を用いて実習を多く
- ある事例(課題)を考える
 - ある動作をする「プログラム」
 - もちろん、簡単版
- 最後に Java 言語について

14

Java の長所・短所

- 長所
 - フリーの開発環境
 - プログラムがしっかり書ける
 - かなり大規模な開発が可能
- 短所
 - 初心者にはしきいが高い
 - 必ずしも高速ではない(大量の実験に適さず)

15

管理工学科におけるプログラミングの講義

- 本講義(2年春)
 - Rubyを対象として、プログラミングの基礎を中心に
- Java言語, オブジェクト指向
 - ソフトウェア工学(2年秋, 飯島先生)
 - ソフトウェア工学実習(3年春, 飯島先生)
- プログラミングの応用
 - 管理工学実験演習Ⅱ 計算機(COM)実験(3年通年)

16

プログラミングの事例

- 「データ分析支援システム」を作ろう
 - 何をすれば支援になるか？
 - 例えば、
 - データの読み込み
 - 指定したデータの表示
 - 平均値や分散の計算
 - データの書き換え
などなど
- 「対話システム」を作ろう
 - 勿論、極めて基本的な部分のみ

17

データ分析って何？

- データ分析？
 - データから情報を取り出すこと、すなわち
 - 数字の羅列を意味のある言葉にすること
- 「データマイニング」もその一つ
- 様々なツールがある
- ここでは、その基礎の動作をさせる

18

データを読み込んで

例えば、このようなデータ

学籍番号	性別	成績	勉強時間	支出	喫煙の有無	飲酒日数	支持政党
1	0	55	2	6	0	3	自民党
2	1	70	7	3	1	1	公明党
3	0	60	1	6	1	2	公明党
4	1	80	10	2	0	0	自民党
5	0	85	6	5	0	1	公明党
6	1	80	2	4	0	2	自民党
7	0	75	5	4	1	4	自民党
8	0	60	3	2	1	1	その他
9	0	60	3	10	1	2	公明党
10	1	85	3	3	0	1	自民党
11	0	80	7	3	0	0	自民党
12	0	80	7	3	1	0	自民党
13	0	65	4	6	1	2	公明党
14	1	65	5	5	0	3	自民党
15	1	60	5	2	0	1	自民党
16	1	95	7	3	0	0	その他
17	0	55	3	7	1	5	自民党
18	0	60	2	5	1	4	自民党
19	0	75	9	5	0	1	その他
20	1	100	9	2	0	0	自民党

青山学院大学国際政経学部 高森寛

19

例えば、こんな応答をさせたい

該当する列番号と行番号を入力して下さい。

列(1:学籍番号 2:性別 3:成績 4:勉強時間 5:支出 6喫煙の有無 7:飲酒日数 8:支持政党)

>> 3 ←該当する列番号を数字で入力

行(1~40) >> 3 ←該当する行番号を数字で入力

値は 60 です。修正する場合にはその値を入力してください >>

演算の種類を指定して下さい。

1:最大 2:最小 3:平均 4:分散 5:相関 >> 3 ←させたい演算を数字で指定

演算の対象である列番号を入力してください(1~8) >> 3 ←該当する列番号を数字で入力

20

実習について

- この講義では理解を深めるために実習を交えて行ないます。
- 教室...日吉ITC 地下一階
 - 703(50人収容)
 - 704(100人収容)
 - どちらも同じ講義内容

21

成績について

- 成績のつけかた
 - 講義以外の時間にレポートを作成
 - 5回程度を予定
 - 講義中の演習問題(平常点)
 - 平常点+レポートの成績から判定
 - Rubyでプログラムが書ける(自信のある)人は、授業に出席しなくてもレポートさえ出せば単位がとれる
 - 予め申告することが条件

22

講義に関する情報

- 講義資料のURL
 - <http://www.sakurai.comp.ae.keio.ac.jp/class.html>
- 教員, TAへの質問
 - 電子メール
 - 直接質問(アポイント必要)

23

プログラムとは

プログラミングの必要性
プログラムとプログラム言語

24

なぜプログラミング？

- 他の講義・実験・演習、卒論に必要
- 必要な技術
- 知っておくべき技術
- 論理的思考力の訓練

25

プログラムとは①

- 日常使う「プログラム」はどのような意味か？
- すなわち、手順・動作を記した書類
 - 書類といっても、紙に書かれているわけではない

26

プログラムとは②

- コンピュータにおいて用いる「プログラム」とは？
- コンピュータが行う動作を
 - 事細かに
 - 逐一記述したもの

27

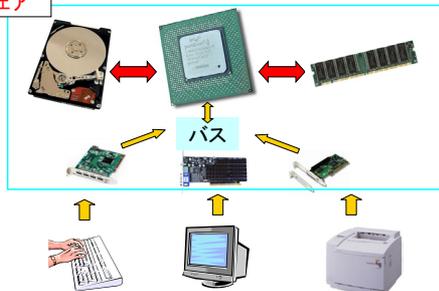
プログラムとは③

- コンピュータの「記憶装置」に蓄えられている
 - メモリ: 普通はコンピュータの中に隠されている。
 - 内容を持ち運びたいときに、USBメモリとかCD-RとかDVD-RAMとかいったものにコピーする
- すなわち、プログラムは「ソフトウェア(軟件)」
 - ハードウェア(硬件)ではない
 - つまり、触って感じる物ではない

28

コンピュータとは

ハードウェア



29

プログラム プログラム プログラム

- OS: プログラムの実行の制御や、ハードウェアの制御と管理など、コンピュータを安全にそして効率良く働かせるための基本ソフトウェア。
 - 例: Windows 2000/XP/Vista, UNIX, Linux, FreeBSD, Solaris, Tron,
- アプリケーション
 - 例: 表計算、文書作成、Java
- ユーザ作成プログラム
 - 例: 「こんにちは」プログラム

30

プログラムは何語で書くか

- 「書類」だから、記述する言語が必要
 - 言語: 意味のある文字列
- 日本語や英語がだめなことは、勿論
 - なぜか?
- コンピュータが分かる言語?
 - 比喩が過ぎる。コンピュータは意味は分からないから。
 - コンピュータが、文字列から自分がすべき動作に変換できればよい
- コンピュータ用の言語を作ればよい

31

それをプログラム言語という

- コンピュータは、メモリのどこかに書いてある「命令」を自分の動作に変換すればよい。
 - この「命令」の構成規則が言語
 - この変換規則は言語ではない
 - コンピュータ(機械)にとっては言語(かな?)なので、機械語といったりする
 - この変換規則の例:
 - 01100 → 出力電圧を5Vに
 - 某神経細胞on → 右手親指曲る (人間の脳)

32

プログラミング言語とは

- 人間の思いをコンピュータに伝える言葉
 - といったって相手はコンピュータですから
 - 人間の言葉より、機械の言葉にずっと近い。ということは
 - 硬い。すなわち、規則にやかましい
 - 手書き文字ではない。すなわち、キーボード入力

33

どんなものがあるか?

- 高級 (high-level) 言語
 - 実行方法による分類
 - コンパイラ言語
 - Ex. Java
 - インタプリタ言語
 - Ex. Ruby
 - 概念による分類
 - 命令型言語
 - Ex. Ruby, Java
 - 関数型言語
 - 分類にならない分類
 - オブジェクト指向言語
 - Ex. Ruby, Java
- アセンブリ言語・機械語

34

Ruby の長所・短所

- 長所
 - 始めやすい
 - インストールが簡単
 - プログラムもその実行も簡単
 - 一行から始められる
 - (実は隠れた長所がたくさんあります。急成長中)
- 短所
 - 「作法」「行儀」が学びにくい
 - 個性が非常に強い
 - 高速な実行に向かない
 - 大きなプログラムが作りにくい

35

プログラミング実習

Rubyプログラムの作成手順

36

“こんにちは”のRubyプログラム

これより右側の文字は、Ruby は無視します。

```
# ファイル名 sample1.rb とします  
puts "こんにちは"
```

#から行末までをコメントと見なします。

日本語以外は半角文字で書いて下さい
全角の空白は使わないで下さい
" " は半角文字で書いて下さい

37

まずは、やってみよう

- 皆さんの「マイドキュメント」は、日吉のPCでは、HDドライブになっています。
- そこに、Ruby という名のフォルダを作ってください。



38

ディレクトリ/フォルダとは

- ハードディスクやCD-ROMなどの記憶装置において、ファイルを分類・整理するための保管場所。
- UNIXやMS-DOSではディレクトリといい、MacintoshやWindowsではフォルダいう。
- Windows の GUI では  のように見えるもの

39

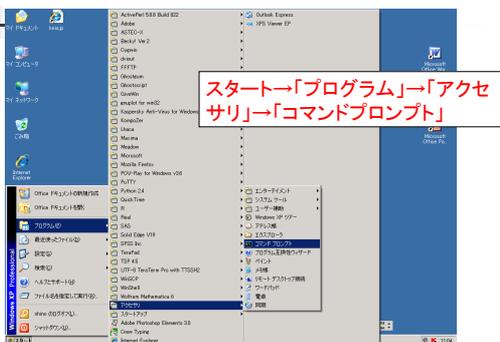
どうすれば、プログラムを書いたことになるの？

- Ruby 言語の場合
 - ①「**コマンドプロンプト**」というプログラムを起動して行う
 - ② メモ帳(でなくてもいいが)で、プログラムを書く(キーボードから入力する)
 - ③ ファイルにセーブ(ハードディスクに入れること)
 - 仮に sample1.rb (全て小文字)という名前だとして以下の話は、
 - ④ 「**コマンドプロンプト**」上で `ruby sample1.rb` と入力。
 - ⑤ エラーがなければ結果が得られる

Enterキー

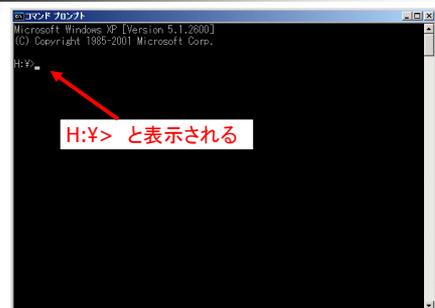
40

① コマンドプロンプトの起動



41

コマンドプロンプトの画面



42

プログラムの書き方

- 二つの作成手順を紹介します
- 初心者はファイルの「拡張子」で混乱します
- どちらの方法でもよいので慣れて下さい

43

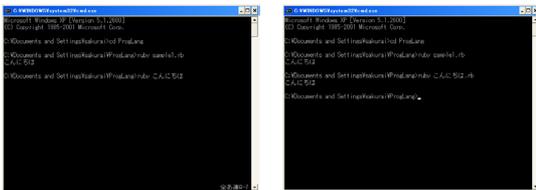
ファイル

- ハードディスクやCD-ROMなどの記憶装置に記録されたデータのまとまり。
- OS(Windows OSなど)は記憶装置上のデータをファイル単位で管理する
- プログラムはファイルに記述する

44

ファイル名

- 識別のために、ファイルにつけられた名前。一つのディレクトリでは、一名一ファイル
- Windows は、大文字・小文字を区別しない。
- 日本語Windowsでは、かな・カナ・漢字も使える。
 - 入力は、Alt + 半角/全角。



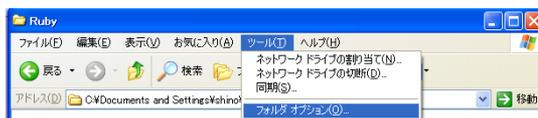
拡張子

- ファイル名の末尾にファイルの種類をあらわす「拡張子」と呼ばれる数数字のアルファベットを付けるのが普通
- ただし、Windows が拡張子を(真剣に!)見るのは、ファイル・アイコンがダブルクリックされたとき
 - ダブルクリックしたときに、メモ帳を起動したいなら ff.txt と、MsWord を起動したいなら ff.doc とする
- Windows ではインストール直後は、拡張子があっても拡張子を表示しなくなっている。便利なようで、極めて不便。
- 皆さんは、当然、表示する方を選ぶ

46

拡張子を常に表示する方法①

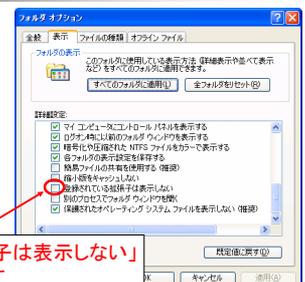
フォルダのメニューバー→「ツール」
→「フォルダーオプション」を選択



47

拡張子を常に表示する方法②

「表示」を選択



「登録されている拡張子は表示しない」
チェックマークをはずす

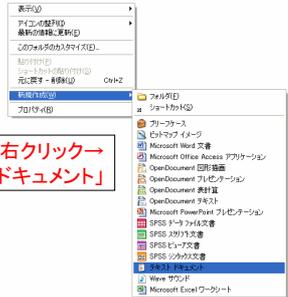
48

プログラムの書き方その①

49

プログラムの記述方法①

「Ruby」のフォルダー内で右クリック→
「新規作成」→「テキストドキュメント」



50

プログラムの記述方法②

ファイル名の変更
「新規テキスト ドキュメント.txt」
から
「sample1.rb」
に変更する

51

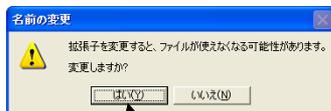
ファイル名の変更方法①

- ファイルアイコンの下にあるファイル名を**ゆっくり二回左クリック**する
- 右クリック → 「名前の変更(M)」
- ファイルの名前を **sample1.rb** としてください。
 - 半角文字
 - 今回の講義では、拡張子(この例でいえば(.rb)は .rb でなくても(.txt)でも)問題はおこらない(はず)。

52

ファイル名の変更方法②

ファイル名を変更すると

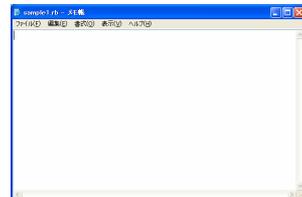


「はい(Y)」をクリック→ファイル名が変更される

53

エディターの起動①

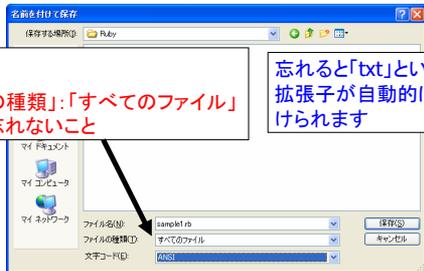
- sample1.rbが  というアイコンであれば、ダブルクリックしてください。
- メモ帳が立ち上がるでしょう、きっと。



54

プログラムの保存②

- メニューバーの「ファイル」→「名前を付けて保存」



重要!

「ファイルの種類」:「すべてのファイル」
の選択を忘れないこと

忘れると「txt」という
拡張子が自動的に付
けられます

61

Rubyプログラムの実行①



コマンドプロンプト上で

cd Ruby

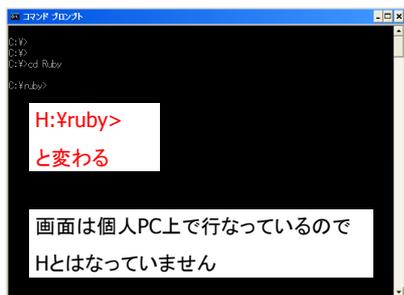
cd

change directory

Enterキー

62

Rubyプログラムの実行②



H:¥ruby>

と変わる

画面は個人PC上で行なっているので
Hとはなっていません

63

Rubyプログラムの実行②

- ruby とは Ruby プログラムを実行するコマンド
 - 指定されたファイルの中身を見て、それに従った動作をする

```
G:¥Ruby> ruby sample1.rb
こんにちは
G:¥Ruby>
```

ruby sample1.rb

Ruby プログラムの実行

ruby Rubyプログラム

64

GUI (graphical user interface)

- 表示として、グラフィックスを用いたユーザインタフェース。入力は、マウスやそれと類似した装置を用いる。
- パソコンでは、Macintosh が使い始めた。
- 今では、これが常識

65

CLI or CUI (command line user interface)

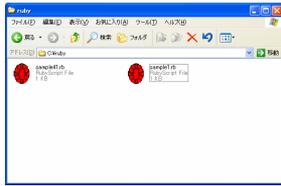
- 表示として、文字列を用いたユーザインタフェース
- 入力はキーボードを用いる。
- 入力するものは、コンピュータに対するコマンドであり、行(ライン)単位に入力する。入力する場所をコマンドラインという。コマンドの実行結果はコマンド入力直後に表示する。画面を使い切ると、スクロールする。
- Windows Vista/XP/2000 では、コマンドプロンプトという言葉が用いられる
 - コマンドプロンプトは、本来は、コマンドラインの先頭にコンピュータが書く文字である

66

コマンドプロンプト①



フォルダーをダブルクリック



コマンドプロンプト上で

H:> cd Ruby

67

コマンドプロンプト②

dir と入力

```
C:\Ruby>dir
ドライブ C のボリューム ラベルがありません。
ボリューム シリアル番号は 74B7-7992 です

C:\Ruby のディレクトリ

2008/04/03 17:47 <DIR>          .
2008/04/03 17:47 <DIR>          ..
2008/04/03 18:15                51 sample1.rb
2008/03/03 16:14                103 sample41.rb
                2 個のファイル          154 バイト
                2 個のディレクトリ 229,682,393,088 バイトの空き領域
```

68

文字について

- 日本語 Windows が取り扱う文字には、1バイトコード(所謂半角文字)と2バイトコード(所謂全角文字)とがある。
- 昔は、本当に、半角と全角で表示されていたので分かりやすかったが、今では、プロポーショナルフォントなどを用いるので、分かりにくい。
 - 例: A A と並べれば分かるが KEIO keio
- コンピュータはちゃんと区別するからやっかいだ

69

Keio.rb
というファイルがあるはず...

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\sakurai>G:
G:>cd Ruby
G:\Ruby>ruby Keio.rb
ruby: No such file or directory -- Keio.rb (LoadError)
```

エラーメッセージ
「Keio.rb」というファイルがないというエラーが表示

70

```
G:\Ruby のディレクトリ
2007/03/25 21:59 <DIR>          .
2007/03/25 21:59 <DIR>          ..
2007/03/25 21:47                19 Keio..rb
2007/03/25 21:47                19 Keio.rb
                2 個のファイル          38 バイト
                2 個のディレクトリ 1,877,010,688 バイトの空き領域
```

「i」が半角文字のファイル
「i」が全角文字のファイル

71

空白について

- Ruby にとって、空白は区切り文字。連続する空白は一つの空白と同じ。しかし、
- Ruby が空白とみなす空白は1バイトコードの空白だけ。2バイトコードの空白は Ruby にとっては空白ではない。
 - よく読んでください。決して、禅問答ではありません。
- どちらの空白かは、人間がみて区別しにくいので、ちょっと目には訳の分からないこと、しかし、よく考えれば分かることが起こる

72

```
# Keio1.rb
puts "こんにちは"
```

```
G:~$Ruby>ruby Keio1.rb
こんにちは

G:~$Ruby>ruby Keio2.rb
Keio2.rb:1: Invalid char `¥201' in expression
Keio2.rb:1: syntax error, unexpected $undefined, expecting $end
puts "こんにちは"
^
G:~$Ruby>
```

```
# Keio2.rb
puts "こんにちは"
```

全角の空白が含まれている

73

どこが間違っているでしょうか

```
# Keio4.rb
puts "こんにちは"
```

```
G:~$Ruby>ruby Keio4.rb
Keio4.rb:2: unterminated string meets end of file
G:~$Ruby>
```

74

どこが間違っているでしょうか

```
G:~$Ruby>ruby Keio4.rb
Keio4.rb:2: unterminated string meets end of file
G:~$Ruby>
```

```
# Keio4.rb
puts "こんにちは"
```

全角文字

75

```
# Keio5.rb
puts "こんにちは"
```

```
G:~$Ruby>ruby Keio5.rb
Keio5.rb:2: Invalid char `¥201' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥261' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥361' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥311' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥277' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥315' in expression
Keio5.rb:2: unterminated string meets end of file
Keio5.rb:2: warning: parenthesize argument(s) for future version
G:~$Ruby>
```

76

```
G:~$Ruby>ruby Keio5.rb
Keio5.rb:2: Invalid char `¥201' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥261' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥361' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥311' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥277' in expression
Keio5.rb:2: Invalid char `¥202' in expression
Keio5.rb:2: Invalid char `¥315' in expression
Keio5.rb:2: unterminated string meets end of file
Keio5.rb:2: warning: parenthesize argument(s) for future version
G:~$Ruby>
```

```
# Keio5.rb
puts "こんにちは"
```

全角文字

77

コンピュータは忠実である

- 言われたとおりに、実行する
- 規則通りに書かれていない場合は、実行せずに、エラーメッセージを出力する
- 書かれたように読む
 - 決して、「きっとこう書きたかったのだろうなあ」と考えて読むことはない
 - 勿論、「『きっとこう書きたかったのだろうなあ』と読んで読む」ようにプログラムを書けば、そう書いた範囲で「考えて読む」ようにはなる

78

プログラム構文上の大原則

- 括弧(広い意味での括弧です)は、開いたら、必ず閉じる。
 - Ruby での例外:「#」で始まるコメント(プログラムと関係のない書き込み)は、改行(そして改行のみ)が閉じる記号
- 複数種の括弧が混じるときには、互いに交錯してはならない
 - 例: { ([]) }
 - 誤例: {] { ([]) }

79

練習問題

80

他の例題(同じようにプログラミングしてみてください)

```
# sample12.rb ← ファイル名
print "春の"
print "うららの"
puts "隅田川"
```

```
G:%Ruby>ruby sample12.rb
春のうららの隅田川
G:%Ruby>
```

```
# sample13.rb
print "春の" + "うららの"
puts "隅田川"
```

```
G:%Ruby>ruby sample13.rb
春のうららの隅田川
G:%Ruby>
```

```
# sample14.rb
puts "春の"
puts "うららの"
puts "隅田川"
```

```
G:%Ruby>ruby sample14.rb
春の
うららの
隅田川
G:%Ruby>
```

他の例題

```
# sample15.rb ← ファイル名
a=5
b=4
```

```
print "a+b=", a+b, "\n"
print "a-b=", a-b, "\n"
print "a*b=", a*b, "\n"
print "a/b=", a/b, "\n"
```

```
C:%Ruby>ruby sample15.rb
a+b=9
a-b=1
a*b=20
a/b=1
```

82

Rubyに関する情報

83

Ruby 関連サイト

- Official site:
<http://www.ruby-lang.org/ja/>
 - 「ドキュメント」→本文中の「チュートリアル」とすると「プログラミング入門」
 - 「ドキュメント」→本文中の「リファレンスマニュアル」
 - 「ダウンロード」→本文中の「ActiveScriptRuby」または「Ruby-mswin32」
- <http://www.namaraii.com/rubytips/> も便利です

84

参考書

- 各種出ています。ご自分の気に入ったものでよいと思います。on-line文書もあります。
 - プログラミング入門 - Rubyを使って -
 - <http://www1.tf.chiba-u.jp/~shin/tutorial/>

85

Rubyのインストール

個人PCへのRubyのインストール

86

Ruby のインストール①

- ActiveScriptRuby
- もっとも簡単。ただし、administrator 権限が必要(自分のPCならOK)
- ActiveScriptRuby をインストールする
 - <http://arton.hp.infoseek.co.jp/indexj.html> を参照のこと

87

ActiveScriptRubyのインストール①



ActiveRuby.msi
Windows インストーラ パッケージ
5,590 KB

ダウンロードしてきたファイルをダブルクリック



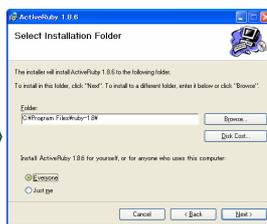
「実行」をクリック

88

ActiveScriptRubyのインストール②



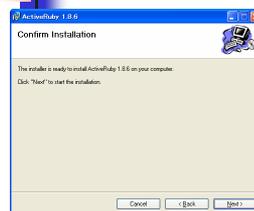
「Next」をクリック



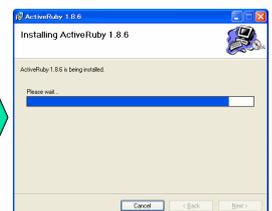
「Next」をクリック

89

ActiveScriptRubyのインストール③



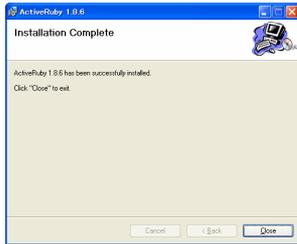
「Next」をクリック



インストール中...

90

ActiveScriptRubyのインストール④



インストール終了

91

ActiveScriptRubyのインストール⑤



ダブルクリック



コマンドプロンプトの起動

日吉ITCとはコマンドプロンプトを起動した時のドライブ、フォルダーが異なるの注意して下さい

92

Ruby のインストール②

- mswin32 を用いる方法 (2007年4月現在)
 - ruby-1.8.6-p114-i386-mswin32.zip を <http://www.garbagecollect.jp/ruby/mswin32/ja/> よりダウンロード
 - readline.dll を <http://jarp.does.network.org/win32/readline-4.3-2-mswin32.zip> よりダウンロード

関連情報

<http://d.hatena.ne.jp/benikujyaku/20070311>
<http://asaasa.tk/wiki/index.php?Ruby%2F%E5%89%8D%E6%BA%96%E5%82%99%2FWindows>