

基礎プログラミングI

第1回 プログラミングの基礎

メディア情報コース
平居 悠（ひらい ゆたか）

自己紹介

平居 悠 (ひらい ゆたか)

メディア情報コース 講師

居室：H-2研究室

E-mail: yutaka.hirai@koeki-u.ac.jp

出身：埼玉県

専門：天文学、数値シミュレーション

学位：博士（理学）東京大学 2018年

職歴：理化学研究所、東北大学、

アメリカ・ノートルダム大学



アイスブレイク

同じチームの人に1人1分程度で自己紹介をしよう。その際、所属するコースとそのコースを選んだ理由を話そう。

この授業の位置付け

データサイエンス・AI教育プログラム：

データサイエンスやAIに関する基礎的な知識と技術、及びその知識や技術を他の科目や学修で応用する能力を育成

「データサイエンス・AI教育プログラム」の流れ（※8単位以上修得）

1年次秋学期

データリテラシー



2年次春学期

基礎プログラミングI



2年次秋学期

基礎プログラミングII



授業概要

やさしいプログラミング言語**Ruby**を通じて

- コンピュータによるものの考え方
 - 論理の組み立て
- を修得する。

到達目標

計算機上での情報の取り扱い方の基礎の理解

1. コンピュータに指示を送る基本を理解する。
2. 文字列や数値を基本とした値の概念とそれを格納する変数の仕組みを理解する。
3. 変数一つで集合を表す概念を理解する。
4. 2進数を基本としたコンピュータの内部表現について基本を理解する。
5. 現実社会の簡単な問題を扱うプログラムを作成できるようになる。

スキルの育成方法

反転学習（事前学習 + 学習者による解説）
を基本とする。

**事前学習がない場合、履修が進まないので
注意。**

反転学習の方法

- 解説動画
- 授業資料
- 事前課題

を次回授業までに学習する。それを踏まえて授業中は授業内課題等にグループで取り組む。

受講の流れ

1. 「本日の課題」のプログラムを全員で解く。
2. 自分ができた人は他の人の作成指導をする
3. 全員作れたらどの行がどんな意味か全員で確認する。
4. 確認できたらTAを呼んで試問してもらおう。
5. TAは1人ずつ「ここはどういう意味？」と聞いてくるので文法は働きについて説明する。
6. 点数をp. 11に示す基準によりつけてもらおう。

成績評価

平常点と試験を1:1の割合で評価し
タイピングの技能を加点

平常点と期末試験両方が特に優れた
者のみに「秀」を与える。

平常点

点数	基準
0	欠席
4	出席のみ（事前学習なしの場合授業内課題が解けてもこれが上限）
5	とにかく動くプログラムが作れた
6	プログラムも作れ内容の質問に的確に答えられる
8	事前学習をしっかりとした上で全面的に理解しており積極的に説明するなどして他のメンバーが「功労者」と認める活躍をした

これにtrr試験、確認テストの点数を加算する。

タイピング

trr 「日本国憲法」 2Hランキングスコアで評価

点数	スコア
9	300
8	250
7	200
6	150
<6	~99

期末試験

- 100点満点。
- 全クラス合同で試験期間中に実施。
- 平常点上位者には試験免除制度あり。
ただし「秀」を取得したい場合は要受験。
- 過去問：<http://roy.e.koeki-u.ac.jp/shiken/pf1/>

授業ページ

学生専用サーバ→平居→基礎プログラミングI

学内：<http://roy.e.koeki-u.ac.jp/>

学外：<https://www.koeki-prj.org/roy/>

参考書

「著作権保護コンテンツ」



Ruby プログラミング 基礎講座

Learning Computing with Ruby

広瀬雄二 著



技術評論社

「著作権保護コンテンツ」

たのしい

高橋征義 + 後藤裕蔵 著
Masayoshi Takahashi, Yuuzou Gotou
まつもとゆきひろ 監修
Yukihiro Matsumoto

Ruby

第6版

わかりやすくNo.1!
Ruby入門書の決定版。

書店販売部数第1位のRuby本*

本書は初心者の気持ちを
裏切らない出来になっています

— Rubyの父 まつもとゆきひろ

SB Creative

Software Design plus

Ruby3.0 対応 その後のバージョンも
Webでフォロー

プロを目指す人 言語仕様からテスト駆動開発・
デバッグ技法まで

Ruby

入門

伊藤 淳一 著

<改訂2版>

Rubyを知れば、Railsはもっと楽しくなる

Rubyの基礎知識からプロの現場で必須のテクニックまで、丁寧に解説。Railsアプリの開発が「なんとなくわかる・書ける」から「ちゃんとわかる・書ける」に変わります。

本書が初版で数多くの初心者を助けてきたように、この改訂版が、あなたがプロの壁を乗り越えるお役に立ちますように。
— まつもと ゆきひろ (「本書の刊行に寄せて」より)

技術評論社

授業計画

第1回	4月9日	プログラミングの基礎
第2回	4月16日	変数・制御構造
第3回	4月23日	メソッド、値の型変換
第4回	4月30日	確認テスト
第5回	5月14日	集合処理1（配列）
第6回	5月21日	集合処理2（CSVとデータ処理）
第7回	5月28日	集合処理3（ハッシュ）
第8回	6月11日	正規表現
第9回	6月18日	計算機の内部表現
第10回	6月25日	スタイルとデバッグ、実用的なプログラム
第11回	7月2日	作品の公開
第12回	7月9日	チーム課題作成
第13回	7月16日	チーム課題発表
第14回	日程未定	期末試験

参考書対応表

授業回	たのしいRuby	プロを目指す人のためのRuby入門
1	1章	1.1~2.3節
2	4~5章	2.4~2.5節
3	7章	2.5~2.13節
5	2.1節	4章
6	3章	4章
7	2.2節	5章
8	2.3節	6章
10	-	3章

質問等

- s4グループ（基礎プロIGH）
- メール (yutaka.hirai@koeki-u.ac.jp)
- オフィスアワー（H2研究室木曜2限）

タイピング練習スケジュール

- 第1回 ホームポジション
- 第2回 ローマ字
- 第3回 英語初級
- 第4回 日本国憲法 (trr試験)
- 第5回 ホームポジション
- 第6回 ローマ字
- 第7回 英語初級
- 第8回 日本国憲法 (trr試験)
- 第9回 ホームポジション
- 第10回 ローマ字
- 第11回 英語初級
- 第12回 日本国憲法 (trr試験、合格スコア150)

タイピング練習スケジュール

- 第1回 ホームポジション
- 第2回 ローマ字
- 第3回 英語初級
- 第4回 日本国憲法 (trr試験)
- 第5回 ホームポジション
- 第6回 ローマ字
- 第7回 英語初級
- 第8回 日本国憲法 (trr試験)
- 第9回 ホームポジション
- 第10回 ローマ字
- 第11回 英語初級
- 第12回 日本国憲法 (trr試験、合格スコア150)

ホームポジション

左手でタイプするキー

右手でタイプするキー



左手の人差指から小指までの
ホームポジション

両手の親指の
ホームポジション

右手の人差指から小指までの
ホームポジション

trr起動方法

1. ブラウザを起動し、<https://www.koeki-prj.org/trr/>に繋ぐ。
2. 学籍番号（Cは大文字、省略なし8桁）を入力する。
3. Koeki MAILに届いたパスコードをPasscode: 欄に入力する。

授業計画

第1回	4月9日	プログラミングの基礎
第2回	4月16日	変数・制御構造
第3回	4月23日	メソッド、値の型変換
第4回	4月30日	確認テスト
第5回	5月14日	集合処理1（配列）
第6回	5月21日	集合処理2（CSVとデータ処理）
第7回	5月28日	集合処理3（ハッシュ）
第8回	6月11日	正規表現
第9回	6月18日	計算機の内部表現
第10回	6月25日	スタイルとデバッグ、実用的なプログラム
第11回	7月2日	作品の公開
第12回	7月9日	チーム課題作成
第13回	7月16日	チーム課題発表
第14回	日程未定	期末試験

今回の問い

- プログラミングとは何だろうか？
- なぜプログラミングを学ぶのか？
- Rubyで“Hello , world!”と画面に表示させるにはどのように書けば良いか？

今回の目標

コンピュータに指示を送る基本を理解する。

プログラム作成までの基本操作をできるようになる。

今回学ぶこと

1. プログラミングとは
2. プログラミングの流れ
3. 問題を解くプログラム

今回学ぶこと

1. プログラミングとは
2. プログラミングの流れ
3. 問題を解くプログラム

私たちの生活はコンピュータに 支えられている



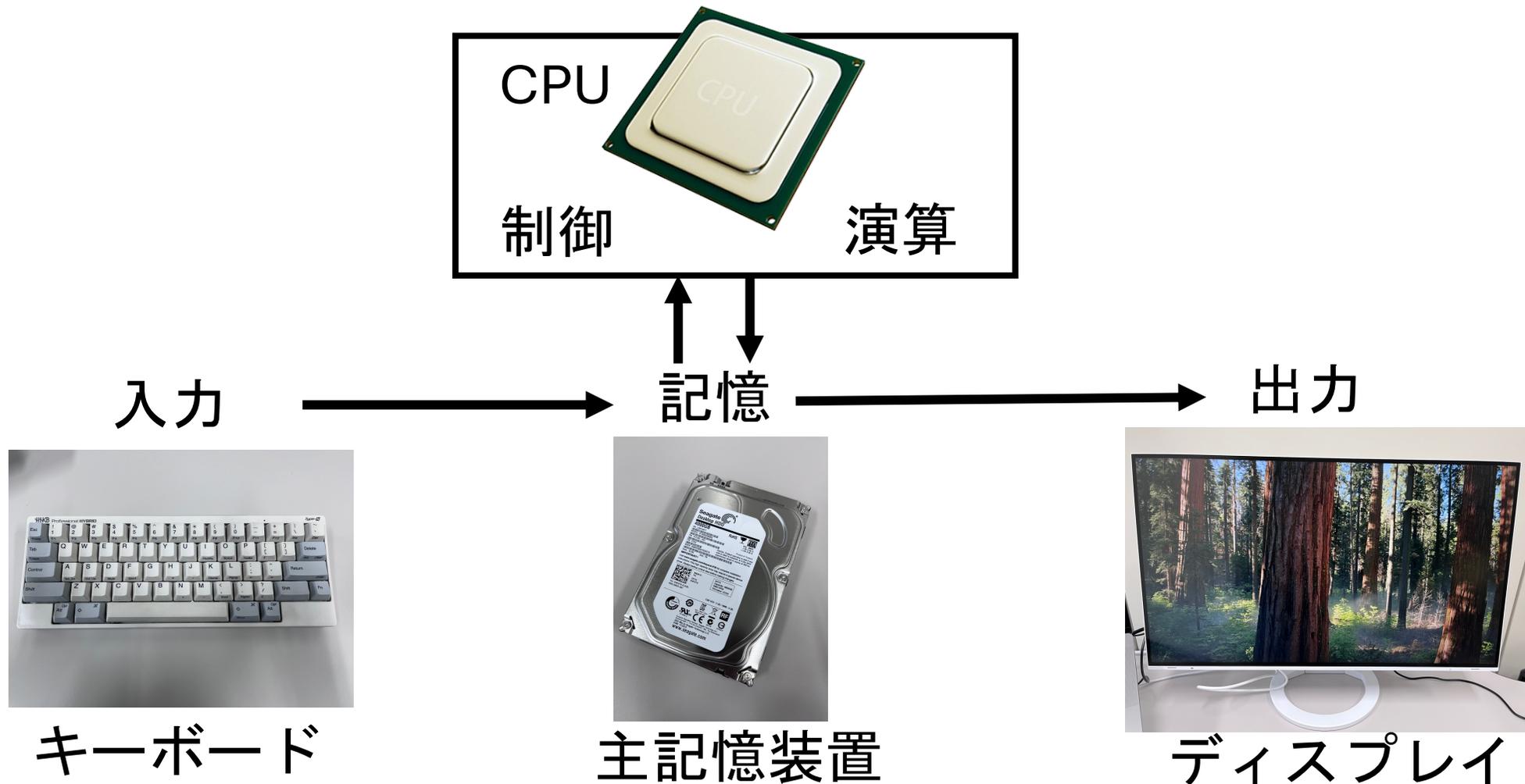
(c) Apple



画像の無断使用・無断転載を禁じます

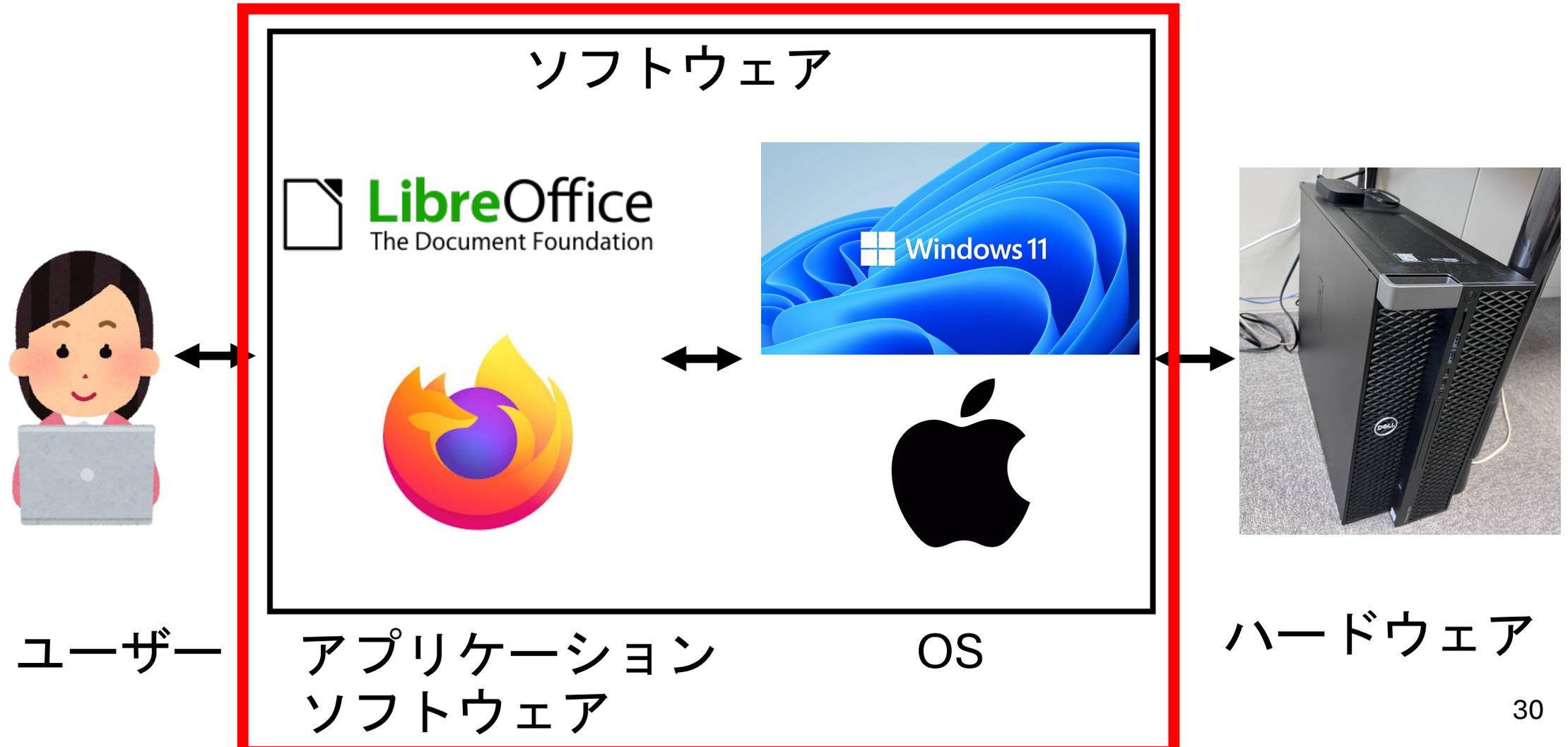
©RIKEN

計算機の仕組み（ハードウェア）



計算機の仕組み（ソフトウェア）

プログラミング



プログラミングとは？

プログラミング言語の文法に従って処理手順を書く工程

コンピュータは日本語を読めない



1 + 1 を
計算して



私たち（普通の人）は機械語を読めない

画面に“Hello”と表示する機械語の例



```
b8 01 00 00 00  
bf 01 00 00 00  
48 8d 35 13 00 00 00  
ba 06 00 00 00  
0f 05  
b8 3c 00 00 00  
31 ff  
0f 05  
48 65 6c 6c 6f 0a
```

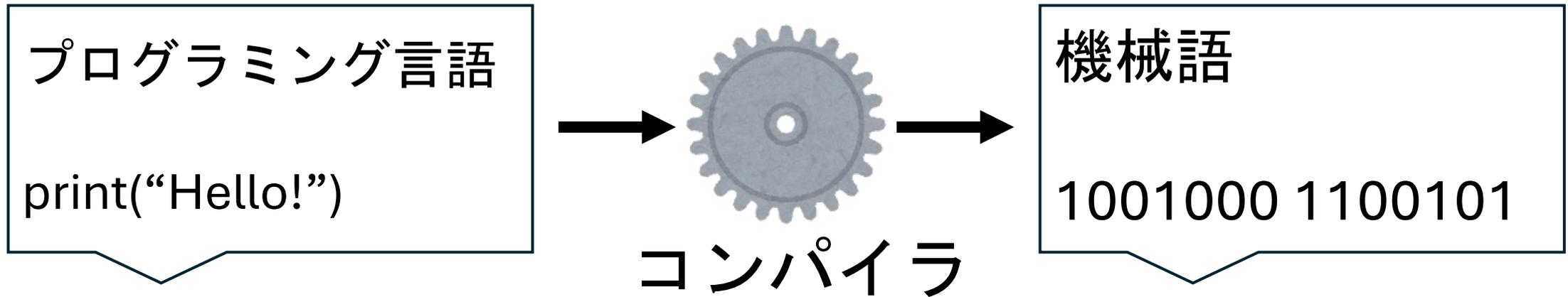
プログラミング言語Rubyを用いた場合

```
print "Hello \n"
```



コンパイル

プログラミング言語をコンピュータが読める機械語に翻訳



今回の問い

- プログラミングとは何だろうか？
 - プログラミング言語の文法に従って処理手順を書く工程
- なぜプログラミングを学ぶのか？
- Rubyで“Hello”と画面に表示させるにはどのように書けば良いか？

なぜプログラミングを
学ぶのか？

グループワーク

身の回りでコンピュータプログラムが使われている物を5つ以上挙げてみよう

プログラミングを学ぶこと
≒世の中の仕組みを知ること

プログラミングで身に付く能力

- 論理構成力
- 状況判断力
- 予測・推理力
- 創造力

職業としてのプログラマー

プログラミング技術を生かして
様々なアプリやシステムを開発

今回の問い

- プログラミングとは何だろうか？
 - プログラミング言語の文法に従って処理手順を書く工程
- **なぜプログラミングを学ぶのか？**
 - **世の中を支えるコンピュータが動く仕組みを学べる。論理構成力、状況判断力などが身に付く。**
- Rubyで“Hello, world!”と画面に表示させるにはどのように書けば良いか？

主なプログラミング言語

Python
(パイソン)



開発者：
グイド・ヴァンロッサム(1991年)

用途：
データ分析、人工知能など

授業：
基礎演習（平居）

Ruby
(ルビー)



まつもとゆきひろ
(1995年)

Web開発など

基礎プログラミング

C言語
(シーげんご)



デニス・リッチー
(1972年)

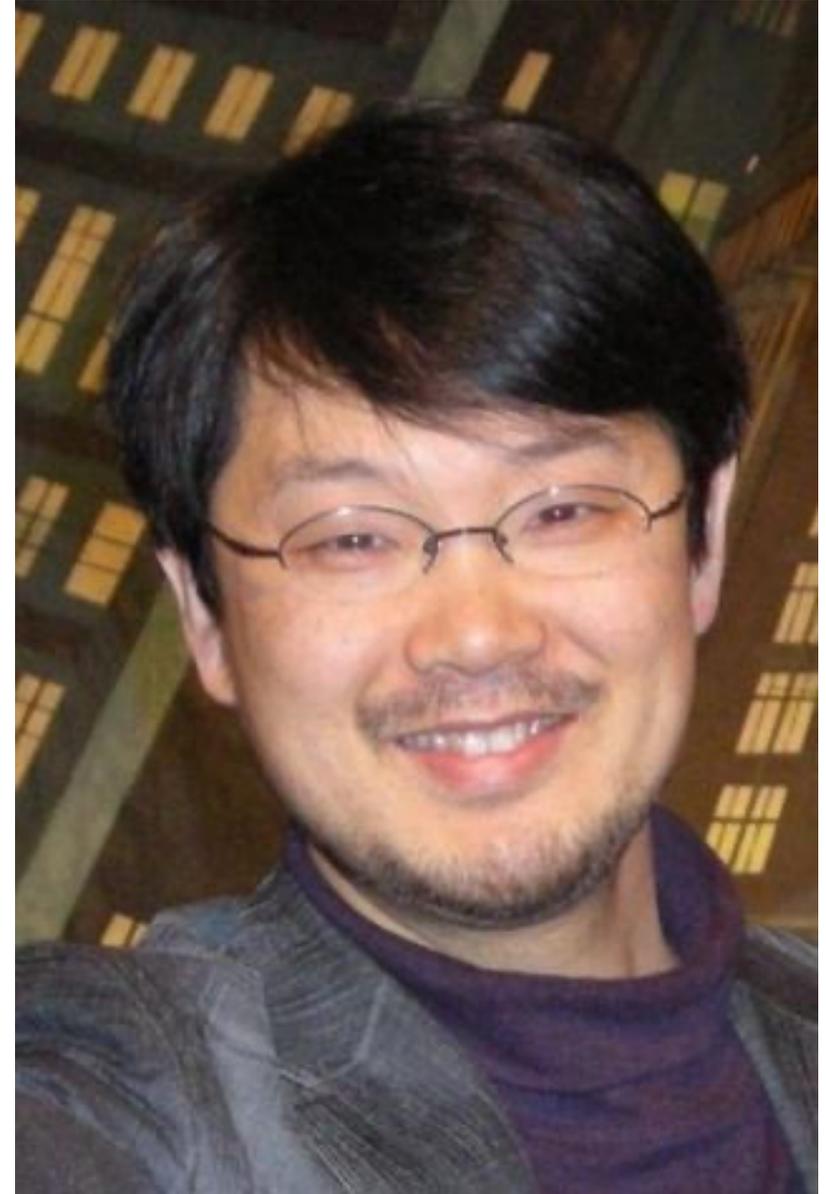
科学技術計算、OSなど

応用プログラミング、
データ構造とアルゴリズム

プログラミング言語Ruby（ルビー）

1993年まつもとゆきひろ氏により開発。1995年公開。

日本製として初めて国際規格として認められた。



なぜRubyを学ぶのか？

Web開発の主要な言語の一つ

Rubyを用いて開発されたアプリの例：

クックパッド

Airbnb



なぜRubyを学ぶのか？

直感的でわかりやすい文法

例：“Hello, world!”と出力するプログラム

Ruby

```
print “Hello, world!\n”
```

C言語

```
#include <stdio.h>

int main (void)
{
    printf(“Hello, world!\n”);
    return 0;
}
```

なぜRubyを学ぶのか？

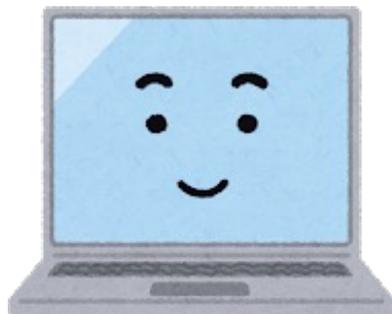
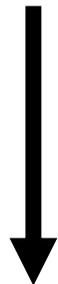
プログラムをすぐ実行できる

Ruby

インタプリタ言語

```
print("Hello, world! \n")
```

プログラム
を1行ずつ機
械語に翻訳



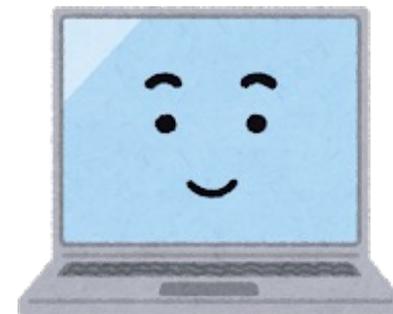
C言語

コンパイラ言語

```
#include <stdio.h>

int main (void)
{
    printf("Hello, world! \n");
    return 0;
}
```

プログラム全体
を機械語に翻訳



なぜRubyを学ぶのか？

「プログラミングの楽しさ」を最大化することを目標に設計・開発

- 簡単に使いこなせる
- 機能的に十分に強力

今回学ぶこと

1. プログラミングとは
- 2. プログラミングの流れ**
3. 問題を解くプログラム

プログラミングの用語

ソースコード :

人間が読みやすいプログラミング言語で書かれたプログラム

機械語 :

コンピュータが読むことができる0と1で表された言語

コンパイラ :

人間が書いたプログラムを機械語に変換する機能

インタプリタ :

ソースコードを1行ずつ機械語翻訳するプログラム

プログラミングを始めてみよう！

プログラミングの流れ

1. エディタ (Emacsなど)でプログラムを作成
2. 保存
3. 実行 (rubyインタプリタの呼び出し)
4. 間違いがあれば修正して2に戻る

ディレクトリの作成

Rubyプログラムをしまおうディレクトリを作成しよう。

“Ruby”というディレクトリを作成する場合：

```
mkdir Ruby
```

ファイルの作成

Emacsで“Ruby”ディレクトリ内にファイルを作るときは、**C-x C-f**で、

```
Find file: ~/Ruby/ファイル名
```

とする。

最初のプログラム

program.rbというファイルを作成して動かしてみよう。C-x, C-fの後以下のように入力してファイルを開く。

```
Find file: ~/Ruby/program.rb
```

最初のプログラム

開いたファイルに右のように書き込んで保存 (**C-x C-s**)。

```
#!/usr/koeki/bin/ruby  
# -*- coding: utf-8 -*-  
print "Hello, world!\n"
```

プログラムの意味

```
#!/usr/koeiki/bin/ruby ←
```

```
# -*- coding: utf-8 -*-
```

```
print "Hello, world!\n"
```

rubyインタプリタ

(翻訳機) がイン

ストールされてい

る場所を指定

自分のPCの場合は異

なる場所を指定

プログラムの意味

```
#!/usr/koeiki/bin/ruby  
# -*- coding: utf-8 -*-  
print "Hello\n"
```

文字コードを
UTF-8に指定
*ここまでは後述
の実行方法2の
場合のみ記述

プログラムの意味

```
#!/usr/koeiki/bin/ruby  
# -*- coding: utf-8 -*-  
print "Hello, world!\n" ←
```

ここがメイン
“Hello, world!”と
出力
\nは改行するための記号

Rubyプログラムの実行

プログラムを実行するときはターミナルの画面で最初に

```
cd Ruby
```

と入力してディレクトリを移動しておく。

Rubyプログラムの実行

実行方法 1

Rubyインタプリタを指定する方法

```
ruby program.rb
```

Rubyプログラムの実行

実行方法 2

プログラム中にRubyインタプリタを埋め込む方法（前述program.rbの1, 2行目）

最初に一度

```
chmod +x program.rb
```

とすると、以後

```
./program.rb
```

で直接実行できる。

chmodコマンド

ファイルの権限（読み込み、書き込み、実行）を変更

```
chmod +x program.rb
```

ファイルに実行権限を与える

今回の問い

- プログラミングとは何だろうか？
 - プログラミング言語の文法に従って処理手順を書く工程
- なぜプログラミングを学ぶのか？
 - 世の中を支えるコンピュータが動く仕組みを学べる。論理構成力、状況判断力などが身に付く。
- Rubyで“Hello, world!”と画面に表示させるにはどのように書けば良いか？
 - `print “Hello, world!\n”`とする。

出席課題

quiz ruby-1を3回以上行い、最も早くクリアした記録の結果をs4基礎プロI (GH)の「#01 pf1出席課題 (quiz ruby-1)」に貼り付ける。

今回学ぶこと

1. プログラミングとは
2. プログラミングの流れ
- 3. 問題を解くプログラム**

問題を解くプログラム

以下の問題を解くプログラムを考える。

人口10万人の都市がある。1年に10%ずつ増加するとした場合、人口が20万人になるのは何年後か。

問題を解くプログラム

これを解く Ruby プログラムは右の通り。

このプログラムを population.rb という名前で作成し、実行してみよう！

```
#!/usr/koeki/bin/ruby
# -*- coding: utf-8 -*-

p=p0=100000
y=0
while p < 2*p0
  p *= 1.1
  printf("%3d年\t%d人\n", y+=1, p)
end
```

実行手順

1. ファイルを開く。Emacsで**C-x C-f**してpopulation.rbを開く。ファイルを作成するディレクトリに注意。

2. 保存する。 **C-x C-s**

3. 実行する。Terminalに移り、

```
chmod +x population.rb
```

```
./population.rb
```

とする。chmodは、一つのプログラムに対して最初の一
回だけやれば良い。

**実行結果とプログラムを見比べ、プログラムの各部分が
どのような意味を持つのかチームで話し合ってみよう！**

今回の目標

コンピュータに指示を送る基本を理解する。

プログラム作成までの基本操作をできるようにする。

今回学んだこと

1. プログラミングとは
2. プログラミングの流れ
3. 問題を解くプログラム

プログラミング上達のポイント

資料などで例に挙げられたプログラムは実際に書いて実行してみよう！

課題

プログラミングの基礎 (<https://www.koeki-prj.org/roy/~yuuji/2025/pf1/02/>)を読み、「課題 1」にあるプログラムを作成せよ。

s4基礎プロJ(GH)の「#01 pf1 提出課題」に指示どおり書き込む。提出後修正して良い。その場合は「編集」リンクから書き換えること。

締切：4月13日(月)

事前課題

1. 第2回解説動画()を視聴する。
2. ターミナルでquiz ruby-2を複数回実行し、最も良い記録をs4基礎プロI(GH)の「#02 pf1 事前課題 (quiz ruby-2)」に貼り付ける。

締切：4月15日 (水)