OSS2
第2回

岩澤全規

資料置き場

- wbtがお亡くなり中なので、nextCloudに資料置きました。

ブラウザでhttp://10.161.1.253:10080を開く。

id:oss2

pass: admin2020

oss2ディレクトリ以下に置いてます。

ただ、課題提出とかには少し不便なので、別の何かを考えるかも、、

- これ使うと良いよ！ってやつあったら教えてください。
- もしなければ、wbtが復活するように祈りを捧げてください。

今日やる事

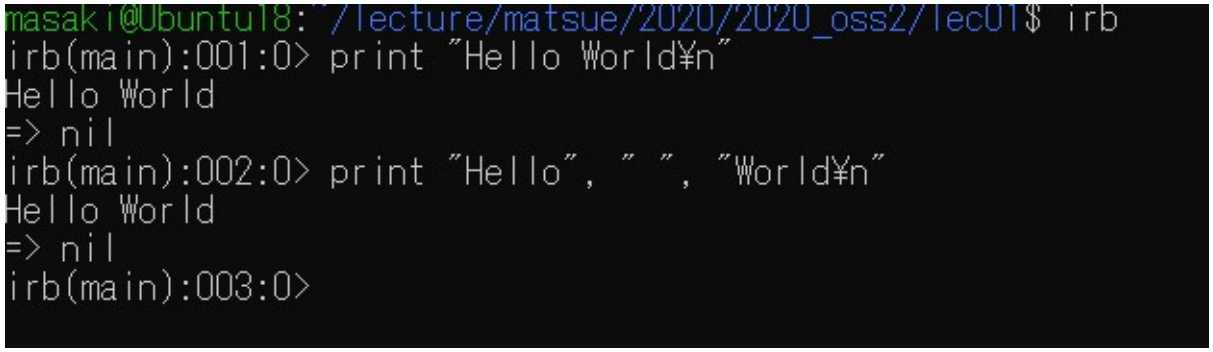
- 環境設定
- Rubyの実行方法
- 標準出力
- コメントの書き方
- 算術演算、数学関連
- 変数
- 標準入力

環境構築

- この講義ではVMware上でUbuntu20を用います。
- 環境構築：
  - プログラムからVMware workstation playerを起動。
  - “仮想マシーンを開く”から、¥C¥vm¥ubuntu20¥ubuntu 64ビット.ovfを選択。
  - インストールできたらログイン
    - ユーザー名: OSS2
    - パスワード: admin2020

Rubyの実行方法1/3

- irb(Interactive Ruby)を使う
  - ターミナルを開き、irbとタイプ。
  - Hello Worldを出力するコードは以下。



  - 終了は、exit もしくは quit
  - 簡単なRubyコードのテストやちょっとした計算に便利。
  - 大規模なプログラム開発には向かない

Rubyの実行方法2/3

- vi などで適当なファイル(hello00.rb)を開く。
- 以下のように記述

```
=begin
hello00.rb
2020/11/07 M. Iwasawa
=end
print "Hello World\n"
print "Hello", " ", " World\n"
```

- =beginから=endで囲まれた部分はコメントアウトされる。
(C言語の'/*'、'*/'と同じ。)

- 以下のコマンドで実行
  - $ ruby hello.rb

Rubyの実行方法3/3

- 適当なファイル(hello01.rb)を開き、以下のように記述

```
#!/usr/bin/ruby
=begin
hello01.rb
2020/11/07 M. Iwasawa
=end
print "Hello World\n"
print "Hello", " ", "World\n"
```

行頭に'#!/usr/bin/ruby'を付ける

- 実行権限を付与
  - $ chmod u+x hello2.rb
- 以下のコマンドで実行
  - $ ./hello2.rb

標準出力

- Rubyには画面に出力するためのメソッド(関数)がたくさんある。
  - Rubyでは関数の事をメソッドと呼ぶ。
  - print, puts, p

printメソッド

- print: 文字や数字を標準出力に表示。
- 下記のコードを作成し実行せよ。

```
print "hoge\n"
print ("fuga\n")
print ("piyo\n"); print ("foo\n");
print 1, 2, 3, "\n"
print 'fuga\n'
```

- メソッドの引数には()を付けてもつけなくても良い。
- 式の区切りは、改行もしくは；(セミコロン)
- ‘’ではエスケープシーケンス(¥nとか)が無視される。

putsメソッド

- printメソッドとほぼ同じだが、文字列の最後に改行が入る。
- 下記のコードを作成し実行せよ。

```
puts "hoge"
puts "fuga\n"
puts 1, 2, 3
puts 'fuga\n'
```

- 末尾に改行がある場合は、putsは改行を出力しない。

printfメソッド

- printf: 書式付き出力。C言語のprintfとほぼ同じ
- 下記のコードを作成し実行せよ。

```
printf "hoge\n"
printf "val0=%d, val1=%.5f, str=%s\n", 1, 3.14259265359, "fuga"
```

pメソッド

- p: 改行は自動で入る。文字列と数値を区別して出力するなど、デバッグに使う。
- 下記のコードを作成し実行せよ。

```
p 100, 200
p "300", "400"
p "500\n"
```

- エスケープシーケンスは無視される。

コメント

- =begin, =end: 囲まれた範囲がコメント。
  - C言語の’/*’, ‘*/’に相当
- #: #より後の行がコメント
  - C言語の’//’に相当

```
=begin
ここはコメント。
ここもコメント。
=end
print "Hello World\n" # ここすらコメント。
```

算術演算、数学関連

- +- * / 等の四則演算子は普通に使える。
- **:べき乗, %:剰余
- sinやsqrt(√)を使うためには、行頭に'include Math'を記述。
  - 'include Math'を書くと数学関連の関数や定数(e.g. π)などが使える。
  - Mathの中には数学関連の関数や定数が記述されている。Math等の事をモジュールという。

```
include Math
print "1+1*15=", 1+1*15, "\n"
print "2^8=#{2**8}\n"
puts  "sqrt(2.0)=#{sqrt(2.0)}"
puts  "sin(PI)=#{sin(PI)}"
```

文字列中の'#{式}'は式の値が展開される。

算術演算、数学関連

- ‘include Math’を書かないで以下のように記述する事もできる。
  - Math.メソッド
  - Math::定数

```
print "1+1*15=", 1+1*15, "\n"
print "2^8=#{2**8}\n"
puts  "sqrt(2.0)=#{Math.sqrt(2.0)}"
puts  "sin(PI)=#{Math.sin(Math::PI)}"
```

変数

- (ローカル)変数名は小文字orアンダースコア('_')で始める
- Rubyの変数宣言には型を指定する必要がない。
  - 右辺の値から型が分かるため。(型推論)

C言語

int num = 10;
char str[] = “hogehoge”;

Ruby

num=10
str = “hogehoge”
さらに、
num, str = 10, “hogehoge”
とも書ける。
さらに、違う型でも再代入できる
val = 10
val = “hoge”

変数

- 以下の(左下の)BMIを求めるプログラムを作成し実行せよ。

Ruby

```ruby
weight, height = 70.0, 1.7
bmi = weight / (height**2.0)
print "BMI is ", bmi, "\n"
print "BMI is #{bmi}\n"
```

C言語

```c
#include<stdio.h>
int main(){
    double weight = 70.0;
    double height = 1.7;
    double bmi = weight / (height*height);
    printf("BMI is %lf\n", bmi);
    return 0;
}
```

文字列の中で'#{変数名}'とすると、変数の値が展開される。

標準入力

- gets: 標準入力から一行読み込むメソッド
- getsの戻り値は文字列で末尾に改行が入る
  - 数値を扱うには文字列を変換する必要がある。後述。
- 以下のコードを作成し実行せよ

```
print "文章を入力してください。\n"
str = gets
print str
p str
```

文字列と数値の変換

- to_i: 文字列を整数に変換
  - “100”.to_i
- to_f: 文字列を浮動小数点に変換
  - “3.14”.to_f
- to_s: 数値を文字列に変換
  - 1.41421356.to_s

左のコードを作成し実行せよ。

```
print "1) 整数を入力してください。\n"
num0 = gets
print "1) あなたが入力した数字:\n"
p num0

print "\n2) 整数を入力してください。\n"
num1 = gets
num1 = num1.to_i
print "2) あなたが入力した数字:\n"
p num1

print "\n3) 整数を入力してください。\n"
num2 = gets.to_i
print "3) あなたが入力した数字:\n"
p num2

print "\n4) 少数を入力してください。\n"
num3 = gets.to_f
print "4) あなたが入力した数字:\n"
p num3
```

gets.to_[if]
とも書ける

- 変換された型を調べるにはclassメソッドが便利
  - “hoge”.class

演習1

- classメソッドを用いて“hoge”, 3.14, “2.71828”, 57の型を調べよ。

演習2

- 標準入力から体重、身長を入力するとBMIを画面に出力するプログラムを作りなさい。

まとめ

- 標準出力
  - print, puts, printf, p
- コメントの書き方
  - =begin, =end, #
- 算術演算、数学関連
  - include Math
  - Math.メソッド、Math::定数
- 変数
  - 型推論
- 標準入力
  - gets
  - to_[sfi]
  - class