

プログラミング1

(第2回) Pythonインタプリタとスクリプトの体験1, ペア・プログラミングの導入

1. Chapter 1 の補足1

1. Calculations and Remembers
2. Computational thinking

欲しい出力を得るためのレシピを考える必要がある。
レシピ≡アルゴリズム。

2. Chapter 2 -- 2.1.2までの補足

1. Glossaries, 用語集1, 2
2. [教科書] 2.1.2 変数と代入
3. Reserved words, 予約語

レシピを記述するための道具(基本的な型・算術演算子・比較演算子・論理演算子・変数・代入文)を覚えよう。

3. 数値演算と文字列演算の補足とprint()の応用編

4. スクリプトの利用

1. スクリプトとは?
2. スクリプトを書いて動かしてみよう
3. スクリプト vs. インタプリタ

基本演算とprint()書式を覚えよう。

インタプリタ実行とファイル実行を使い分けよう。

5. 変数名・ファイル名の命名規則

6. マニュアルの参照

7. 演習

8. 宿題

help()やオンラインマニュアルを活用しよう。

慣習を守ることで「他人が読みやすいコード(readable code)」になる。

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2017/prog1/>

プログラミング1

(第3回) インタプリタとスクリプトの体験2: 文字列とif文, 関数の利用

1. Chapter 2.2, 2.3, 4.1.1の補足

- 2.2 Branching Programs (条件分岐)
- 2.3 Strings and Input (文字列と入力)
- 4.1.1 Function Definitions (関数定義)
- Reserved words, 予約語

2. ペアプロ演習

3. 宿題

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2017/prog1/>

Chapter 2.2, 2.3, 4.1.1の補足

Glossaries, 用語集1, 2
[教科書] 2.1.2 変数と代入
Reserved words, 予約語

2.2 Branching Programs (条件分岐)

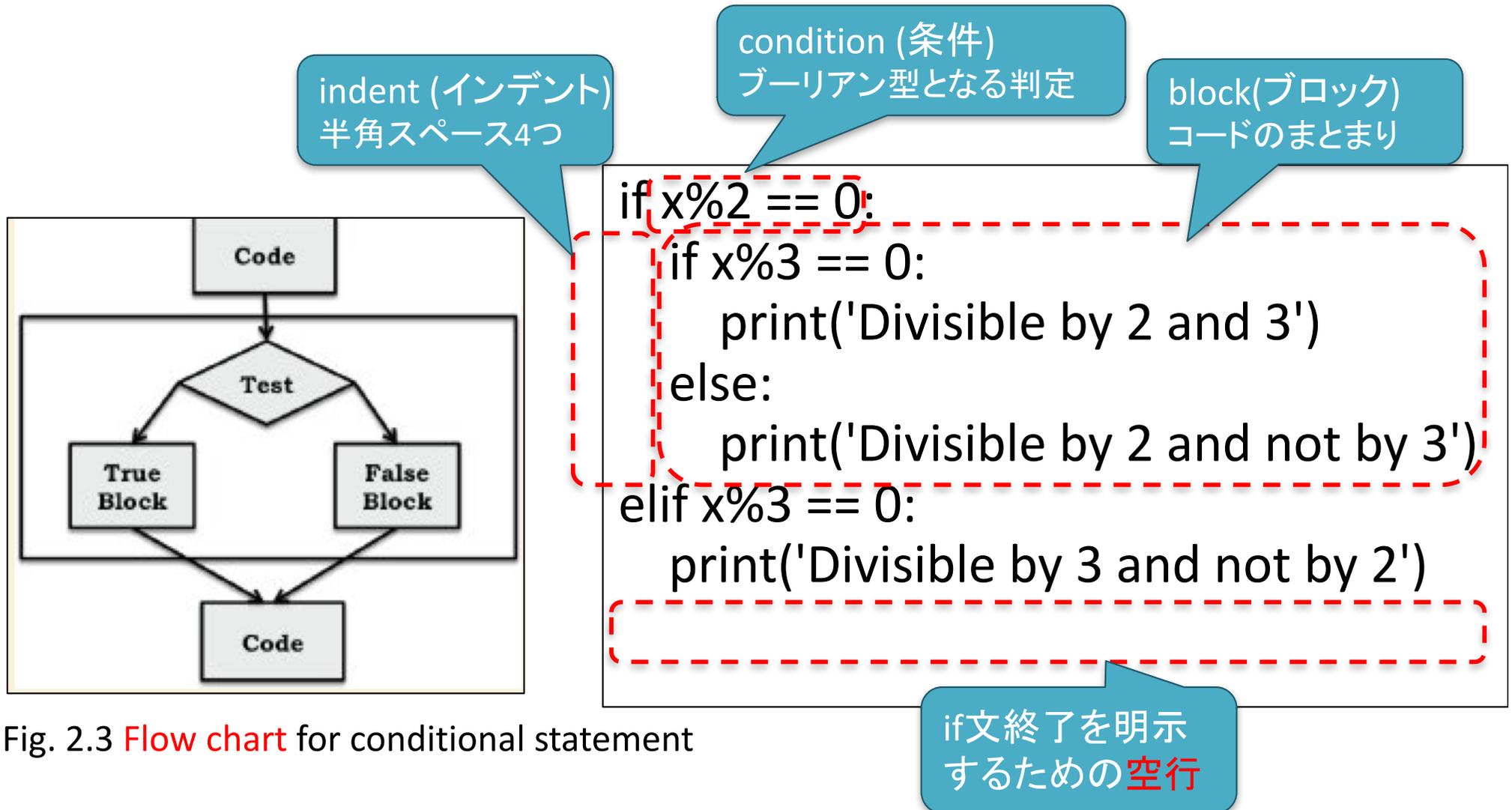


Fig. 2.3 Flow chart for conditional statement

プログラミング1

(第3回) インタプリタとスクリプトの体験2: 文字列とif文, 関数の利用

1. Chapter 2.2, 2.3, 4.1.1の補足

- 2.2 Branching Programs (条件分岐)
- 2.3 Strings and Input (文字列と入力)
- 4.1.1 Function Definitions (関数定義)
- Reserved words, 予約語

2. ペアプロ演習

3. 宿題

if文を使い、条件分岐できるようになる。ブロックを指定するためのインデントを忘れずに。

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2017/prog1/>

2.3 Strings and Input (文字列と入力)

* 数字を入力したつもりでもstr型として受け取る。
* int型として処理したいならint()関数を使ってcast(キャスト, 型変換)しよう。

input()

- 標準入力(≡キーボード)からの入力読み込み、str型を返す。
- 入力を促す文を引数で指定。

```
ターミナル — Python — zsh — ttys000 — 80  
>>> name = input('Enter your student number: ')  
Enter your student number: 945734  
>>> e_mail_address = 'e' + name + '@ie.u-ryukyu.ac.jp'  
>>> print('your e-mail address is ' + e_mail_address)  
your e-mail address is e945734@ie.u-ryukyu.ac.jp  
>>> □
```

2.3 Strings 1 (文字列)

overloaded operator (多重定義)

+演算子

(1) int, float なら加算する。

(2) 文字列同士なら連結する

```
>>> 1 + 1
2
>>> 'a' + 'a'
'aa'
>>> a
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
>>> 'a'*'a'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can't multiply sequence by non-int of type 'str'
>>>
```

「*演算子」に文字列同士の乗算は定義されていないため、TypeError(型エラー)

2.3 Strings 2

len(): 文字列の長さ

'文字列'[0]: 文字列の0番目

- * 指定する位置を Index (インデックス) と呼ぶ。
- * Indexは0番目から数える。
- * 範囲外にはアクセス出来ない (index out of range)。
- * Indexがマイナス指定されると、後ろから数える。

Tips: コマンド実行時の出力を読もう
「IndexError: string index out of range」

'文字列'[x:y]: slicing (スライス処理)
文字列のx番目からy番目までを切り出す

変数に代入されてる
時も同じ操作が可能

```
ターミナル — Python — zsh
>>> len('abc')
3
>>> 'abc'[0]
'a'
>>> 'abc'[2]
'c'
>>> 'abc'[3]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: string index out of range
>>> 'abc'[-1]
'c'
>>> 'abc'[-2]
'b'
>>> 'abc'[1:3]
'bc'
>>> 'abc' == 'abc'[:]
True
>>> 'abc' == 'abc'[0:len('abc')]
True
>>>
>>> string = 'abc'
>>> string[0]
'a'
>>>
```

プログラミング1

(第3回) インタプリタとスクリプトの体験2: 文字列とif文, 関数の利用

1. Chapter 2.2, 2.3, 4.1.1の補足

- 2.2 Branching Programs (条件分岐)
- 2.3 Strings and Input (文字列と入力)
- 4.1.1 Function Definitions (関数定義)
- Reserved words, 予約語

2. ペアプロ演習

3. 宿題

ユーザ入力を受け取るにはinput()。
文字列操作には結合・インデックス指定・スライス操作がある。
型を変更するにはキャストしよう。

4.1.1 Function Definitions

```
def funcion_name(parameters):
```

- **def**: 関数宣言
- **funciton_name**: 関数名
- **parameters**: パラメータ
 - * **arguments**(引数)とも呼ぶ。

Tips: 関数名の命名規則
lower_with_under()

block(ブロック)
コードのまとめ

indent (インデント)
半角スペース4つ

```
>>> def max(x, y):
...     if x > y:
...         return x
...     else:
...         return y
...
[...
[>>> max(1, 2)
2
>>> 
```

return文

- (1) return文に辿り着いたら、その**関数の実行をここで終える**。(それ以降のコードは実行しない)
- (2) 関数の呼び出し元にreturn指定した**オブジェクトを返す**。

def文終了を明示
するための**空行**

プログラミング1

(第3回) インタプリタとスクリプトの体験2: 文字列とif文, 関数の利用

1. Chapter 2.2, 2.3, 4.1.1の補足

- 2.2 Branching Programs (条件分岐)
- 2.3 Strings and Input (文字列と入力)
- 4.1.1 Function Definitions (関数定義)
- Reserved words, 予約語

2. ペアプロ演習

3. 宿題

オリジナルの関数(≒レシピ)
を定義できるようになろう。

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2017/prog1/>

Reserved words, 予約語

<https://goo.gl/4TclUz>

- 一覧(赤丸は今回出てきた予約語)

False
None
True
and
as
assert
break

class
continue
def
del
elif
else
except

finally
for
from
global
if
import
in

is
lambda
nonlocal
not
or
pass
raise

return
try
while
with
yield

演習

前回の続き: 初めてのペア・プログラミング

プログラミング1

(第3回) インタプリタとスクリプトの体験2: 文字列とif文, 関数の利用

1. Chapter 2.2, 2.3, 4.1.1の補足

- 2.2 Branching Programs (条件分岐)
- 2.3 Strings and Input (文字列と入力)
- 4.1.1 Function Definitions (関数定義)
- Reserved words, 予約語

2. ペアプロ演習

3. 宿題

if文を使い、条件分岐できるようになる。ブロックを指定するためのインデントを忘れずに。

ユーザ入力を受け取るにはinput()。
文字列操作には結合・インデックス指定・スライス操作がある。
型を変更するにはキャストしよう。

オリジナルの関数(≒レシピ)を定義できるようになる。

宿題

- 復習: **適宜**(これまでの内容)
 - 課題レポート2「より複雑な判定条件」 * 講義ページ参照。
- 予習: 教科書読み
 - 3章
 - 3.1 Exhaustive Enumeration
 - 3.2 For Loops
- 復習・予習(オススメ): paiza, progate

参考文献

- 教科書: Introduction to Computation and Programming Using Python, Revised And Expanded Edition
- Reserved words, <https://goo.gl/4TclUz>