

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. 振り返り(ミニクイズ)
2. 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
 - Reserved words, 予約語
3. 演習
 1. 前回の続き: 初めてのレポート
 2. 今回の演習: if文, 関数の利用
4. 宿題

講義ページ: <http://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/>

振り返り

- [教科書] 2.1.2 変数と代入
- スクリプト vs. インタプリタ
- 変数名・ファイル名の命名規則
- マニュアル

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)

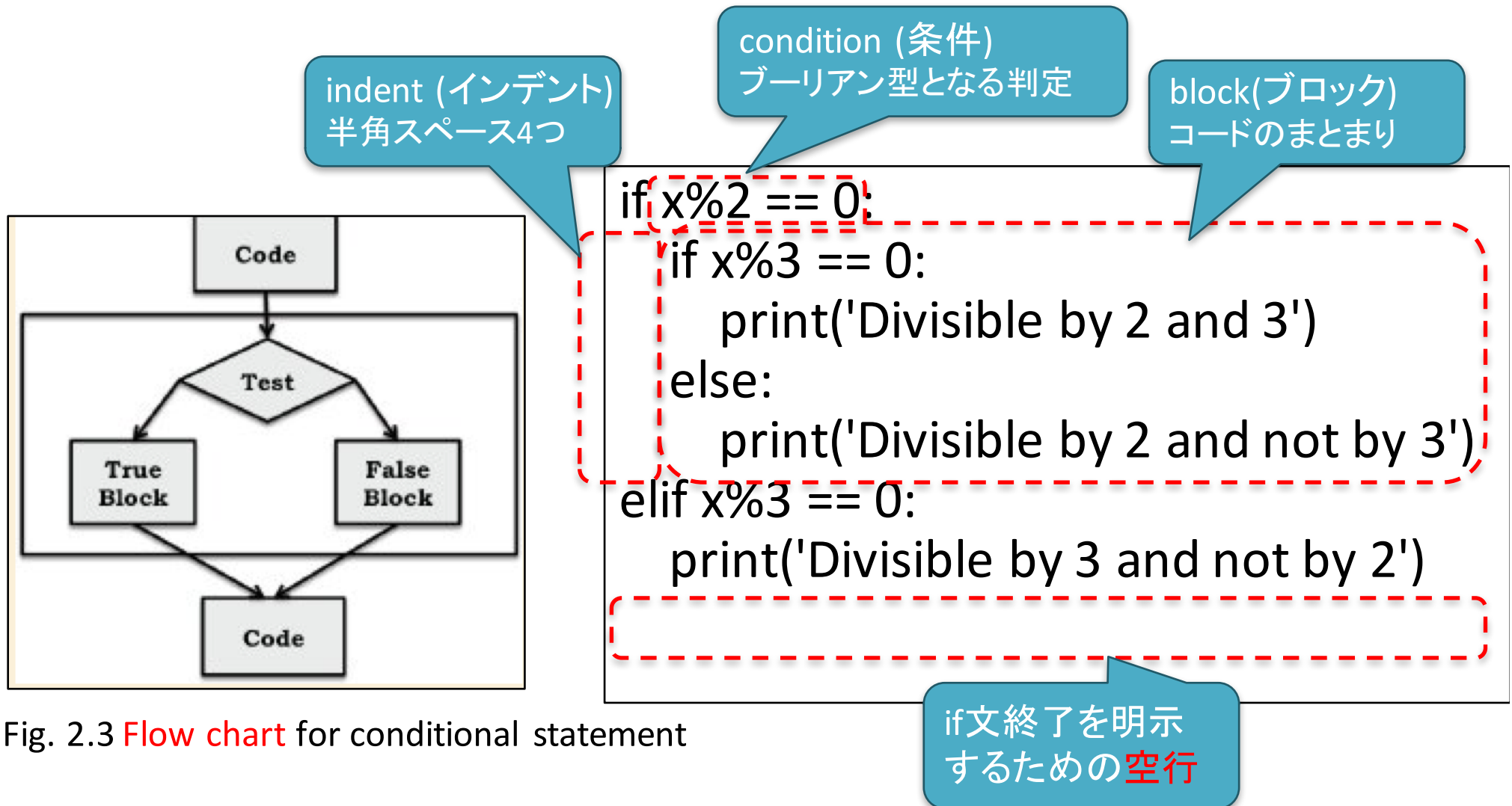


Fig. 2.3 Flow chart for conditional statement

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
>>> □
```

「+演算子」による文字列の連結

- 頭にeを連結
- 後ろに@ie.u-ryukyu.ac.jpを連結

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

```
>>> 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'
>>>
```

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)
コードのまとめり

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
```

```
>>> []
```

if文終了を明示
するための空行

Reserved words, 予約語

<https://goo.gl/rEzdAN>

- 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: 初めてのレポート
今回の演習: if文, 関数の利用

補足1

- ペアプログラミングを始める前に
 - 記入漏れ
 - 「実施日」と「報告者」
 - 前回の復習確認
 - 「何をやったっけ？」
 - 「これはこうやれば良いんだっけ？」

補足2

- ペアプログラミングのやり方

- 7ステップ

- 作業を決める
- 最初の目標を決める
- パートナーを頼りにし、支えてやる
 - driver: 仕事を終わらせることに専念
 - observer: 横から観察し、疑問・改善・簡潔化など大局的な問題について考える
- 喋る
 - 「一人で悩む」のは十秒程度に留める
 - 一緒に相談しながら考える練習
- お互い何をやっているか把握する
 - 頻繁に同期をとる
- 喜ぶ
- 交代する

二人で2,3分考えても分からない場合には、手を上げて質問しよう

演習

- 演習1: 初めてのレポート
 - <https://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/week2-ex.html>
 - レポートの作成手順は授業ページを参照
- 演習2: if文と関数
 - <https://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/week3-ex.html>
- ペア・プログラミング
 - <https://ie.u-ryukyu.ac.jp/~tnal/2016/prog1/week2-pair-programming.html>
 - driver, observer (navigator)

宿題

- 復習: **適宜**(これまでの内容)
 - レポート課題1「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) * 余裕があれば
- 復習・予習(オススメ): paiza
 - Python入門編1:プログラミングを学ぶ (全9回)
 - <https://paiza.jp/works/python3/primer>
 - やれる範囲でok

参考文献

- 教科書: Introduction to Computation and Programming Using Python, Revised And Expanded Edition
- Python 3.5.1 documentation, <https://docs.python.org/3.5/index.html>
- Google Python Style Guide, <https://google.github.io/styleguide/pyguide.html>
- ペアプログラミングのやりかた, <http://goo.gl/ZWtIW>
- (paiza) Python入門編1:プログラミングを学ぶ (全9回), <https://paiza.jp/works/python3/primer>