

免責事項

このワークショップは
アトリエのスタッフが作成したものであり
ソフトバンク公式のものではないことを
ご承知ください。



Atelier Akihabara

ワークショップ 番外編

Python入門~Part2~

2017/7/11

Softbank Robotics



実体験とコミュニティで開発を促進する

アトリエ



相互
促進

コミュニティ

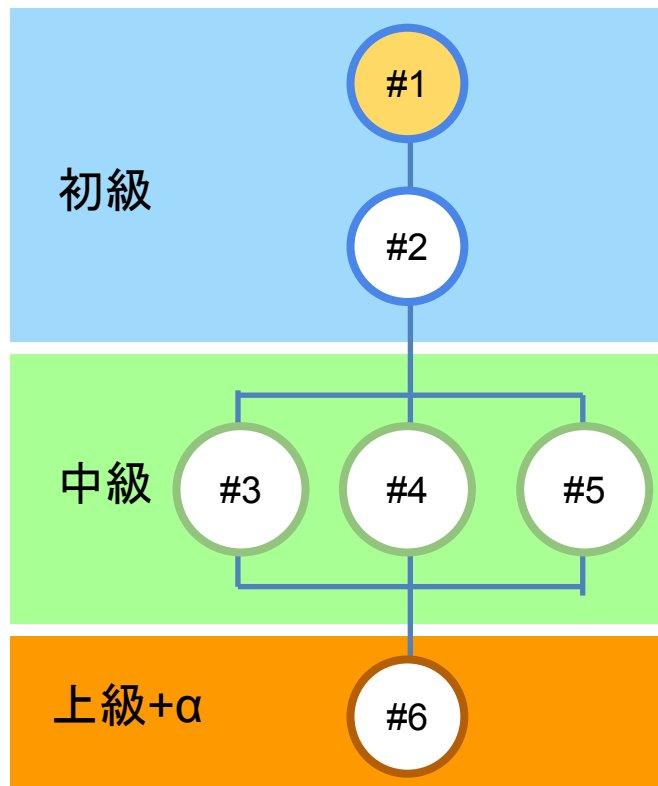


- Pepperのアプリ開発を実体験

- 経験や知見を共有

アトリエ秋葉原について

ワークショップ



タッチアンドトライ

自由に開発
質問はスタッフに
お客様同士の交流
検証や
打ち合わせの利用も可

1週間の予定

月	タッチアンドトライ
火	貸し切り(有料)
水	Pepper for Biz説明会 & タッチアンドトライ
木	貸し切り(有料)
金	タッチアンドトライ & ワークショップ
土日	タッチアンドトライ & ワークショップ

実体験とコミュニティで開発を促進する

アトリエサテライト

有志でPepperと開発スペースを
提供している
企業、大学、コミュニティスペース

秋葉原で回答できない質問は
各サテライトへ



- お名前
- 所属
- 本日の意気込み
- プログラミング経験など

例:

本日の案内を勤めさせていただきます、
大山 まりほ(Mariho Oyama)と申します。



1. 制御文とは？
2. if文
3. for文
4. while文
5. 演習問題



制御文とは？



制御文とは

- プログラムの実行の流れを制御することができる文
 - 条件によって処理を変える
- 複合文とも呼ばれる
 - 他の文を内部に持ち、内部の文を実行するかなどを制御
- Pythonではインデントを用いてブロックを表現



- 真(True): 条件が成り立つ。(「1は0より大きい」「犬は動物だ」)
- 偽(False): 条件が成り立たない。(「1は0より小さい」「動物は犬だ」)
- bool型と呼ばれ、数値との演算をすることもできる
 - True: 1
 - False: 0

型	意味	例
int	整数値	1 , 150 , -30
float	浮動小数点数	1.414 , 0.5 , -1.2
str	文字列	“Apple” , “Hello!” , “50”
bool	真偽値	True,False,1,0



※プログラミングで「=」は代入を意味する

演算子	例	真となる条件
==	a == b	aがbと等しい
!=	a != b	aがbと異なる
>	a > b	aがbより大きい
>=	a >= b	aがbより大きいか等しい
<	a < b	aがbより小さい
<=	a <= b	aがbより小さいか等しい

真となる例

1 < 5

(3 > 2) == 1

4 >= 4

(2 > 5) == False

True != False

偽となる例

1 > 5

(3 > 2) == 0

4 < 4

(2 > 5) == False

True == False



if文
～条件分岐～

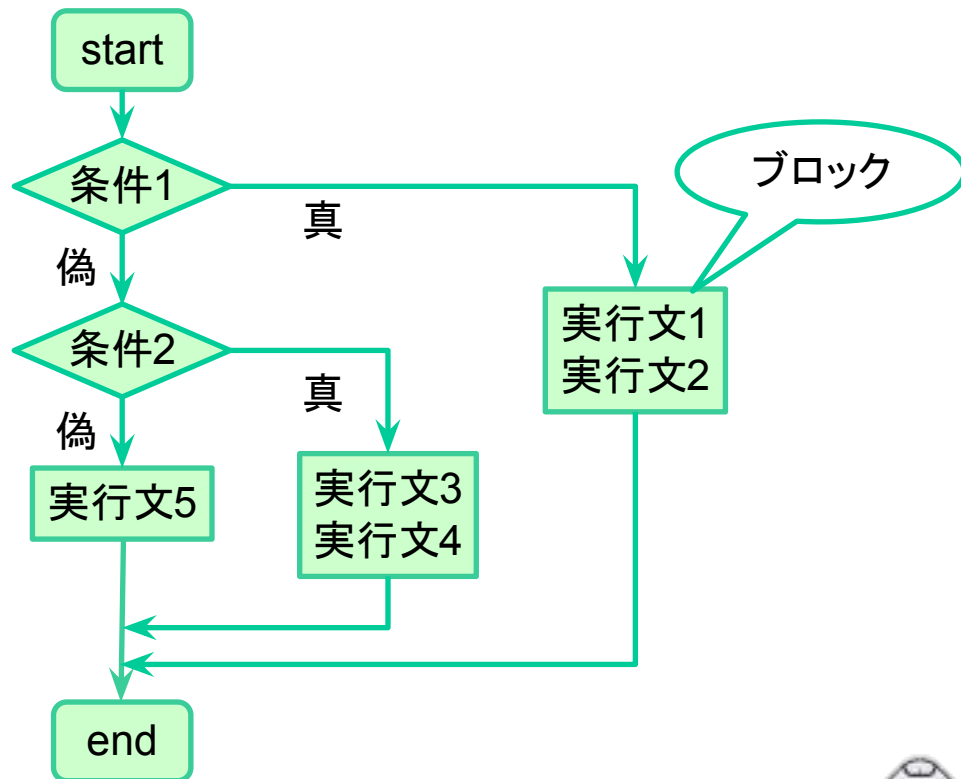


if文 ～条件分岐～

～書き方～

```
if 条件1:  
    #条件1が真の場合  
    実行文1  
    実行文2  
elif 条件2:  
    #条件1が偽の中で  
    条件2が真の場合  
    実行文3  
    実行文4  
else:  
    #条件1も2も偽の場合  
    実行文5
```

～実際の流れ(フローチャート)～



if文 ～条件分岐～

～例1～

```
a = 20 #好きな数字を入力
if a > 10:
    self.logger.info("aは10より大きい")
elif a == 10:
    self.logger.info("aは10")
else:
    self.logger.info("aは10より小さい")
```

aが10より大きい判断するプログラム

～実行結果～

※aが20の時
aは10より大きい

※aが10の時
aは10

※aが3の時
aは10より小さい



if文 ～条件分岐～

～例2(if文の入れ子)～

```
a = 20 #好きな数字を入力
if a > 0:
    if a % 2 == 0:

self.logger.info("aは偶数")
    else:

self.logger.info("aは奇数")
    else:
        self.logger.info("aは負の数")
```

aが正ならば偶数か奇数かを、
負ならば負であると表示するプログラム

～実行結果～

※aが20の時
aは偶数

※aが1の時
aは奇数

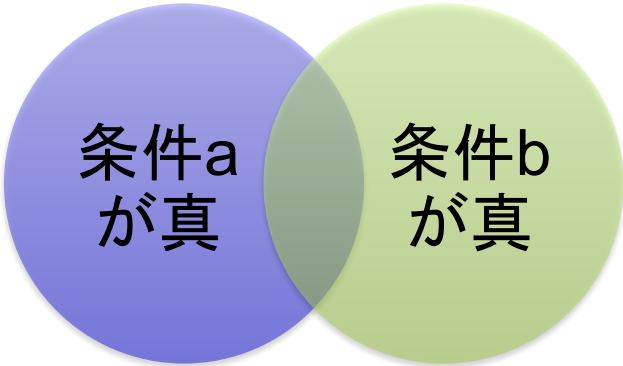
※aが-4の時
aは負の数



論理演算子 ～条件を複数つかう～

演算子	例	真となる条件
and	条件a and 条件b	条件aも条件bも真
or	条件a or 条件b	条件aか条件bが真
not	not 条件a	条件aが偽

条件a	条件b	a and b	a or b
真	真	真	真
真	偽	偽	真
偽	真	偽	真
偽	偽	偽	偽



～例3(論理演算子)～

```
num = 24
if(num % 2 == 0 and num % 3 == 0):
    self.logger.info("numは6の倍数です")
elif(num % 2 == 0 or num % 3 == 0):
    self.logger.info("numは2の倍数または3の
    倍数です")
else:
    self.logger.info("numは2の倍数でも3の倍
    数dでもありません")
```

numが6の倍数なのか、2か3の倍数なのか
それ以外なのかを表示するプログラム

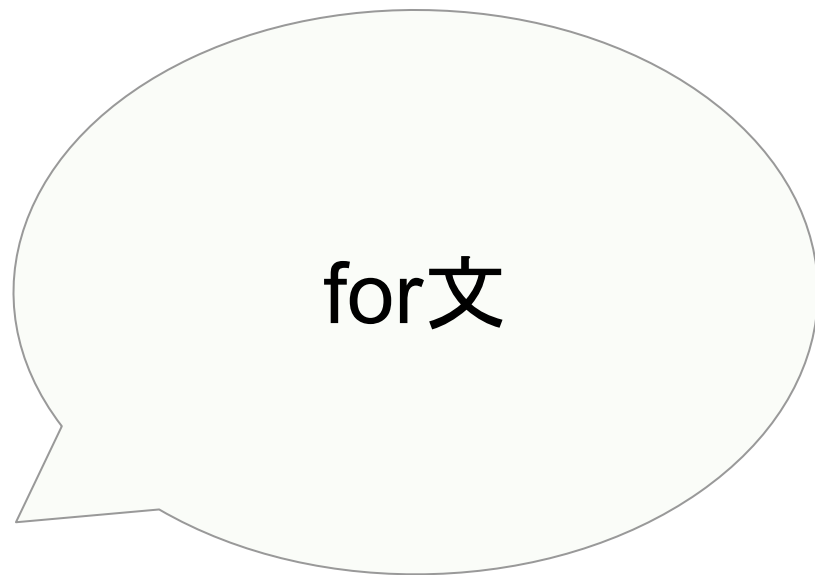
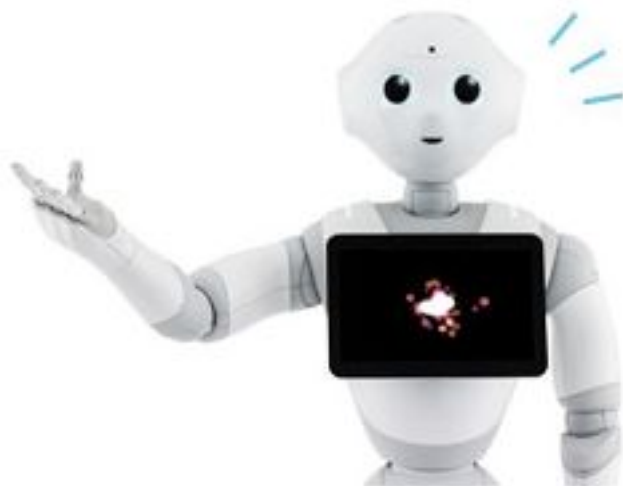
～実行結果～

※numが24の時
numは6の倍数です

※numが27の時
numは2の倍数または
3の倍数です

※numが17の時
numは2の倍数でも
3の倍数でもありません



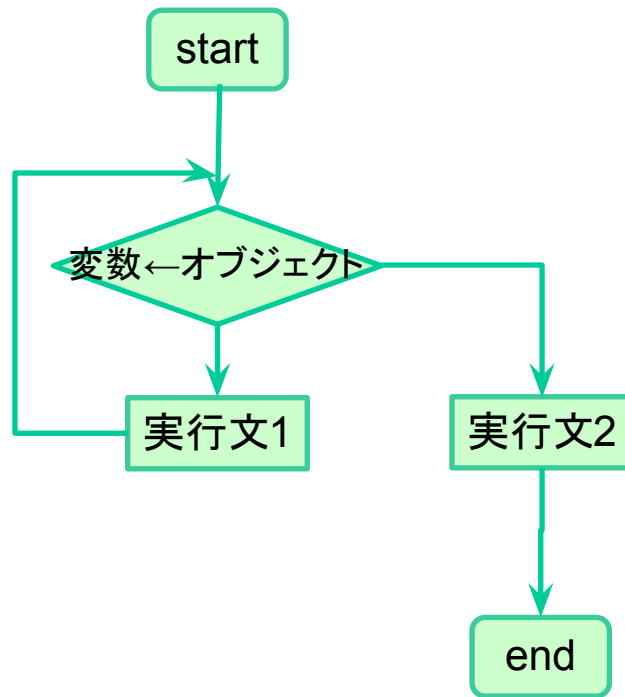


for文 ～繰り返し文1～

～書き方～

```
for 変数 in オブジェクト:  
    実行文1  
    実行文2
```

～実際の流れ(フローチャート)～



～例1～

```
word = "pepper"  
for char in word:  
    self.logger.info(char)
```

wordの中身を1文字ずつ
出力するプログラム

～実行結果～

```
※word = "pepper"
```

p

e

p

p

e

r

決められた回数繰り返したい時はどうしたらいいのか？



- 組み込み関数: Python本体に直接組み込まれている関数
- 指定した長さの連続した整数のリストを自動で生成する関数

書き方	意味
range(num)	0 <u>以上</u> num <u>未満</u> の整数のリスト
range(num1,num2)	num1 <u>以上</u> num2 <u>未満</u> の 整数のリスト
range(num1,num2,num3)	num1 <u>以上</u> num2 <u>未満</u> で 増加量num3の整数のリスト

range(5) – [0,1,2,3,4]
range(2,6) – [2,3,4,5]
range(1,13,3) – [1,4,7,10]



例2～range関数～

```
for i in range(5):  
    self.logger.info(i)
```

決まった回数繰り返すプログラム

～実行結果～

※range(5)

0

1

2

3

4



for文 ～繰り返し文3～

指令: range文とif文を組み合わせて、0～70のうち7の倍数だけを取り出してください。

～指令に対するAさんの回答～

```
for i in range(70):  
    if i%7==0:  
        self.logger.info()
```

おまけ: 回答に1つ間違いが含まれています。
どこでしょうか。

～求められる実行結果～

0
7
14
21
28
35
42
49
63
70



for文 ～繰り返し文3～

range文とif文を組み合わせて、0~70のうち7の倍数だけを取り出してください。

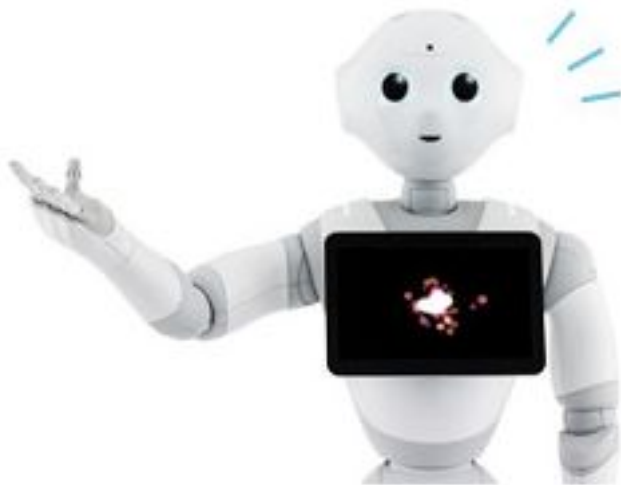
～実行結果～

```
for i in range(71):  
    if i%7==0:  
        self.logger.info()
```

range(70)は70を含まないリストであることに
気をつけましょう

0
7
14
21
28
35
42
49
63
70





while文



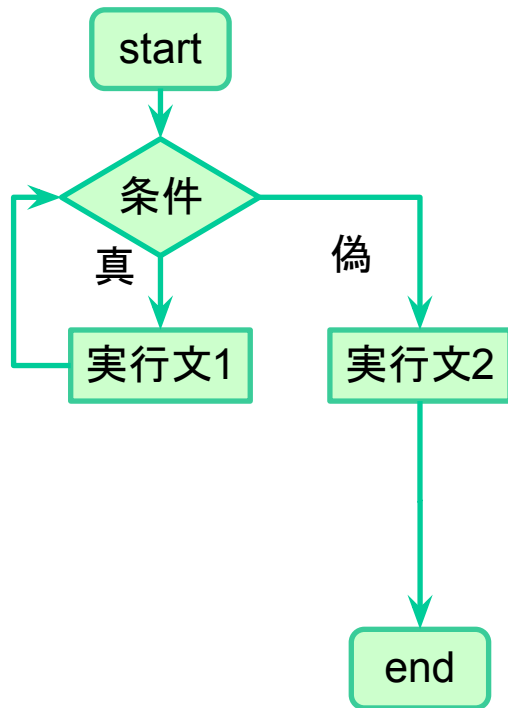
while文 ～繰り返し文～

～書き方～

```
while 条件式:  
    実行文1  
    実行文2
```

条件式が真の間繰り返す

～実際の流れ(フローチャート)～



while文 ～繰り返し文～

例

```
i=0 #事前準備が必要  
while i <5:  
    self.logger.info(i)  
    i+=1
```

決まった回数繰り返すプログラム

～実行結果～

※range(5)

0

1

2

3

4

～インクリメントについて～

i+=1はi=i+1と同義(その他演算も同様)

※C言語やjavaに慣れている方へ。

i++,i--という表現はpythonには存在しない



演習問題1

[0,1,2,3,4,5,6,7,8,9,11,13,15,17,19,20,22,24,26,28,30,40,50,60,70,80,90,100]

というリストを作成しましょう。

ただし、以下の規則性を利用すること。

- 0~9...全ての値を網羅
- 10~20...奇数のみ
- 20~30...偶数のみ
- 30~100...10の倍数のみ

解答形式

```
def onInput_onStart(self):  
    numlist=[] #リストの宣言および初期化  
    #これ以降、numlist.append(追加する内容)で追記する  
    self.logger.info(numlist) #出力
```



演習問題1の解答例

```
def onInput_onStart(self):
    numlist=[]  #リストの宣言および初期化
    #これ以降、numlist.append(追加する内容)で追記する
    for i in range(10):
        numlist.append(i)
    for i in range(10,20):
        if i%2==1:
            numlist.append(i)
    for i in range(20,30):
        if i%2==0:
            numlist.append(i)
    for i in range(30,101):
        if i%10==0:
            numlist.append(i)
    self.logger.info(numlist) #出力
```



演習問題2

[0,2,4,6,8,
11,13,15,17,19,
20,22,24,26,28,
31,33,35,37,39,
40,42,44,46,48,51,53,55,57,59,60,62,64,66,68,71,73,75,77,79,
80,82,84,86,88,91,93,95,97,99]

というリストを作成しましょう。

このリストの要素の規則性は、「10の位と1の位の偶奇が一致すること」です

解答形式

```
def onInput_onStart(self):  
    numlist=[] #リストの宣言および初期化  
    #これ以降、numlist.append(追加する内容)で追記する  
    self.logger.info(numlist) #出力
```



演習問題2の解答例1

```
def onInput_onStart(self):
    numlist=[]  #リストの宣言および初期化
    for i in range(0,10):
        if i %2==0:
            for j in range(0,10):
                if j %2==0:
                    numlist.append(10*i+j)
        if i%2==1:
            for j in range(0,10):
                if j %2==1:
                    numlist.append(10*i+j)

    self.logger.info(numlist)
```



演習問題2の解答例2

```
def onInput_onStart(self):  
    numlist=[]  #リストの宣言および初期化  
    for i in range(0,10):  
        for k in [0,1]:  
            if i %2==k:  
                for j in range(0,10):  
                    if j %2==k:  
                        numlist.append(10*i+j)  
    self.logger.info(numlist)
```



演習問題3

「1つ前の数との差が1ずつ増えていく」規則性を持った以下のようなリスト(階差数列)を作成しましょう。

最初の値は0で、最後の値が300以内に収まるようにしてください。

[0,1,3,6,10,15,21,28,36,.....]

解答形式

```
def onInput_onStart(self):  
    numlist=[]    #リストの宣言および初期化  
    #これ以降、numlist.append(追加する内容)で追記する  
    self.logger.info(numlist) #出力
```



演習問題3の解答

[0,1,3,6,10,15,21,28,36,.....]

```
def onInput_onStart(self):  
    numlist=[]  #リストの宣言および初期化  
    i=0  
    a=1  
    while i<=300:  
        numlist.append(i)  
        i+=a  
        a+=1  
    self.logger.info(numlist)
```

← 足す数が毎回1ずつ増える



おつかれさまでした！
これにてWS番外編は終わりになります。
WSは続けてぜひ受講してみてください
タッチアンドトライで質問もしてみてください

