

Unit 3: 条件, if, while

Application: FizzBuzz

この応用では、素晴らしいFizzBuzz現象を説明するプログラムを作成します。

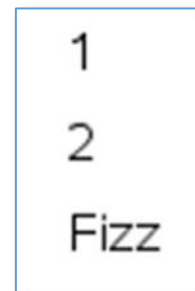
目標

- 2つの方法で終了するwhileループを作成
- 数が割り切れるかどうかを調べる関数を定義

FizzBuzz

自然数1,2,3,...を出力するプログラムを作成しますが、

- 数が3で割り切れるときは、代わりにFizzを表示、
 - 数が5で割り切れるときは、代わりにBuzzを表示、
 - 数が3と5の両方で割り切れるときは、代わりにFizzBuzzを表示、
- Escキーが押されると、プログラムは終了します。



割り切れるか否かは、%演算子(modまたはremainder(剰余))を使ってテストされます。

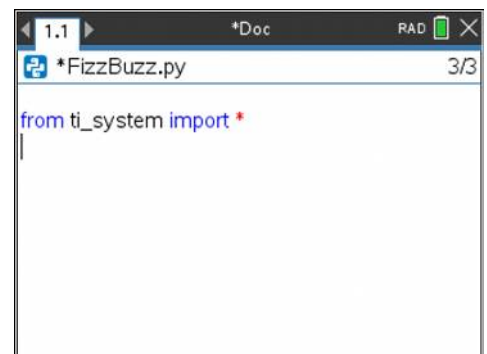
ifステートメントには、もう1つの形式があります。役に立ちます。

- if** <condition>:
 <true block1>
- elif** <another condition>: 複数のelifブロックが存在する可能性
 <true block2> **elif**はelse ifの略
- <さらにelifs?>
- else**:
 <false block> このブロックは他がいずれも真でない場合、処理

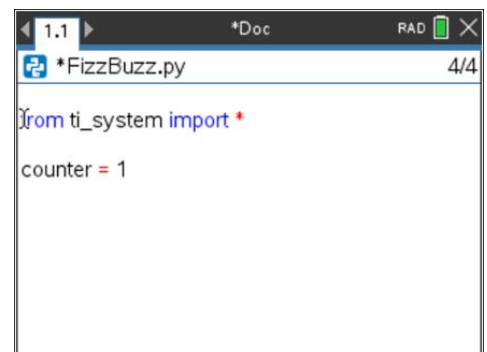
1. 新規のPythonの空白プログラムファイルを開始します。

get_keyを使うので、追加します。
from ti_system import *

これは、**menu > More Modules > TI System**. (メニュー>その他のモジュール>TIシステム)にあります。



2. 1から始まるカウント数を表す変数**counter**を設定します。



3. **while get_key() != 0**を使います。
これは、**menu > More Modules > TI System** (メニュー>その他のモジュール>TIシステム)にあります。



```
*FizzBuzz.py
from ti_system import *
from time import *
counter = 1
while get_key() != "esc":
    block
```

4. **if..elif..else**を使います。
これは、**menu > Built-ins > Control** (メニュー>組み込み>制御)にあります。

覚えておきましょう。それは次のように機能します。

if <this is true>:

 <do this>

elif <this is true>: (else if...の略)

 <do this>

(ここにはもっと多くの**elif**があるかもしれません)

else:

 <do this> (他のどれも当てはまらないとき)

注意：条件を書き込むときは、**=**ではなく**==**を使います。間違った記号を使うと、構文エラーが発生します。**x==5**:のとき、**x=5**:ではありません。

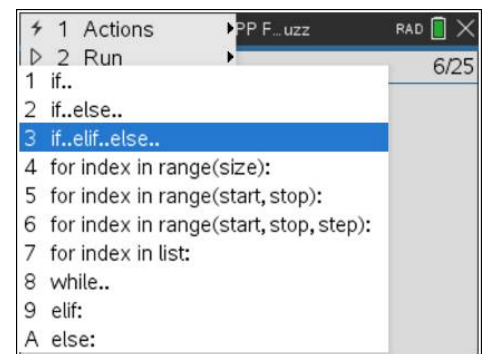
ctrl+=メニューを使うと役立ちます。

5. 右に、**if..elif..else**の構造が記述されています。が、情報は省略されています。

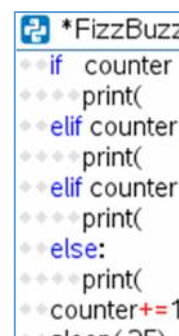
Ifと各**elif**には満たす必要のある条件と、条件が真の場合に処理するブロックがあります。**else**:は条件がなく、他のいずれも真でない場合に、そのブロックが処理されます。

counter+=1は、**counter = counter + 1**の省略形です。

Hint: AをBで割ると、**A%B**は余りを与えることを思い出します。



```
1 Actions
2 Run
3 if..
4 if..else..
5 if..elif..else..
6 for index in range(size):
7 for index in range(start, stop):
8 for index in range(start, stop, step):
9 for index in list:
10 while..
11 elif:
12 else:
```



```
*FizzBuzz
if counter
print(
elif counter
print(
elif counter
print(
else:
print(
counter+=1
sleep(.25)
```

6. プログラムの実行例は、右図のようになります。
数の表示が速すぎるときは、**sleep()**関数を使って速度を落とします。
sleep()はTime(タイム)モジュールに含まれているので、必ず次を含めてください。

from time import sleep

FizzBuzzに置き換えられた最初の数は何ですか。



```
Python Shell 11/11
>>>#Running FizzBuzz.py
>>>from FizzBuzz import *
>>>1
2
Fizz
4
Buzz
Fizz
7
8
>>>|
```