

Unit 3: 条件, if, while

Skill Builder 3: キーを押す

このレッスンでは、電卓のキーを押すことによりループを終了する関数を使います。この機能はTI-Nspire固有なため、1つ(または複数)の特別なモジュールをインポートする必要があります。

目標

- **get_key()**を使ってループ終了
- 2の累乗と1/2の累乗
- **sleep(n)**を使ってコードの実行をn秒間停止

whileループは体験したので、TI-Nspire Pythonシステムの特別で強力な機能である**get_key()**にアクセスしてみましょう。2の累乗(2, 4, 8, 16, 32, 64, ...)を連続して表示し、**esc**キーが押されたとき終了するプログラムを作成します。

Teacher Tip: **get_key()**は、**ti_system**モジュールに含まれている関数です。これは、描画プログラム(後のユニット)とTI-Innovator™ Hubプログラム(別のコース)での作業に多く使用されます。

ti_systemからインポートした**get_key**は、任意のプログラムで使用できるようにするために必要です。そのモジュールには他にも多くの機能があります。オンラインドキュメントを確認してください。

get_key()または**get_key(0)**は、プログラムを停止せずキーが押されているかどうかを確認します。

get_key(1)は1秒間プログラムを一時停止し、キーが押されるのを待ちます。

1. 新規のPythonの空白プログラムを開始し、**powers_of_2**という名前を付けます。
2. **menu > More Modules > TI System** (メニュー>その他のモジュール> TIシステム)より、次のステートメントを取得します。

from ti_system import *

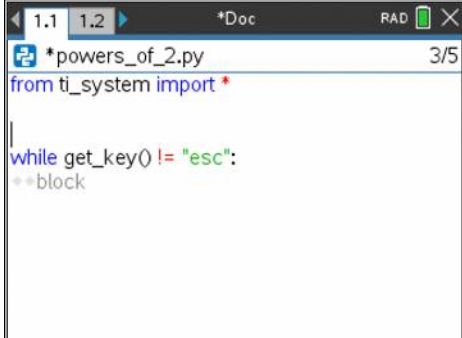
これはリストの一番上にあります。

同じメニューで、次の特別なステートメントを選択します。

while get_key() != "esc":

block

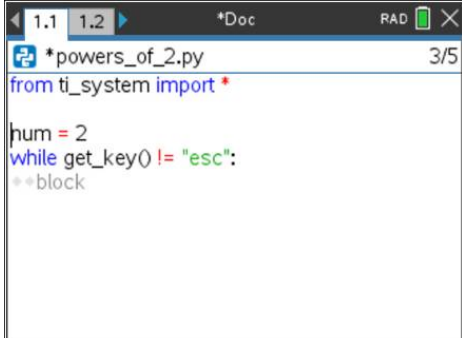
3. ここで、**while**ステートメントの直前に変数**num = 2**を挿入します。



```

1.1 1.2 *Doc RAD
*powers_of_2.py 3/5
from ti_system import *

while get_key() != "esc":
    block
    
```

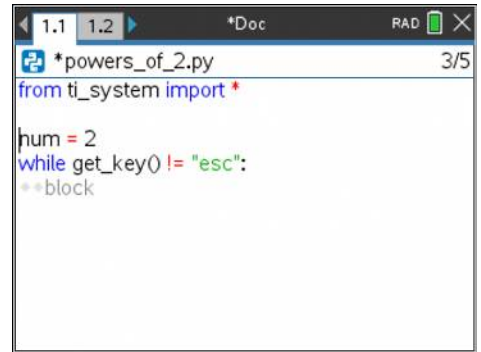


```

1.1 1.2 *Doc RAD
*powers_of_2.py 3/5
from ti_system import *

num = 2
while get_key() != "esc":
    block
    
```

4. ループブロックで、数値を**print**し、それに2を掛ける代入ステートメントを作成します。ステートメントがインデント(字下げ)されていることを確認してください。
- 次のステップでコードをチェックする前に、自分で試してみてください。



```
*powers_of_2.py 3/5
from ti_system import *

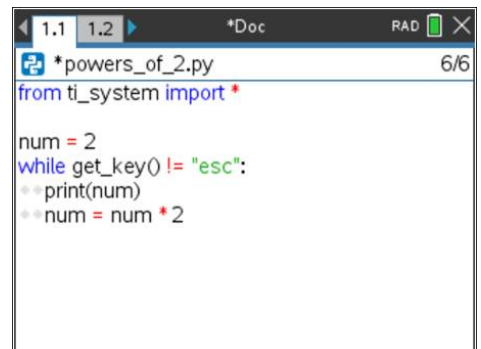
num = 2
while get_key() != "esc":
    *block
```

Teacher Tip: num=num*2 または num*=2を使います。

5. ここで、やってみましょう。

```
while get_key() != "esc":
    print(num)
    num = num * 2
```

プログラムを実行し、**esc**を押して停止します。



```
*powers_of_2.py 6/6
from ti_system import *

num = 2
while get_key() != "esc":
    *print(num)
    *num = num * 2
```

6. 数は本当に速く作成されます。

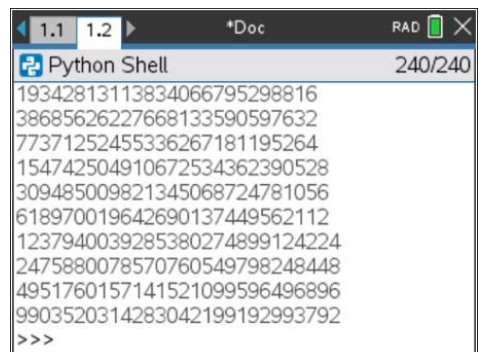
Pythonの2つの興味深い機能：

本当に速くて、

整数に上限はありません。

ただし、コンピュータのメモリの容量により制限されます。

次の仕事は、数字が読めるように遅くすることです。



```
Python Shell 240/240
19342813113834066795298816
38685626227668133590597632
77371252455336267181195264
154742504910672534362390528
309485009821345068724781056
618970019642690137449562112
1237940039285380274899124224
2475880078570760549798248448
4951760157141521099596496896
9903520314283042199192993792
>>>
```

7. Python **Time**(時間)モジュールにある**sleep()**という関数が必要です。

プログラムの先頭に、次のステートメントを追加します。

```
from time import sleep
```

これは、**menu > More Modules > Time** (メニュー>その他のモジュール>時間)のトップにあります。



```
*powers_of_2.py 3/7
from ti_system import *
from time import sleep

num = 2
while get_key() != "esc":
    *print(num)
    *num = num * 2
```

8. `num = num * 2`の下に、次のステートメントを追加します。

sleep(1)

whileブロック内です(その上の2つのステートメントのようにインデントされます)。

プログラムを再度、実行してください。今回は、各数値が表示された後、コンピュータは1秒間スリープ(待機)してから、ループの次のステップに進みます。スリープ数を変更して、処理速度を少し上げます。もう一度`esc`を押して、プログラムを終了します。他の`sleep()`値を試して、表示速度を変更してください。

プログラムの最後に、“Done”(完了)と印刷します。

9. プログラムの各**2**を**0.5**に変更します。

何が起こるでしょう。

他の数値を試してみましょう。

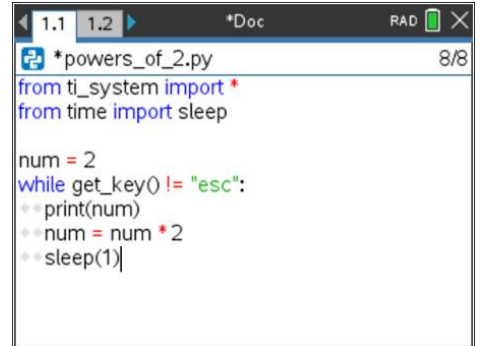
数はいつ増加し、いつ減少しますか。

負の数はどうですか。

2ではなく0または1を試すのは理にかなっていますか。

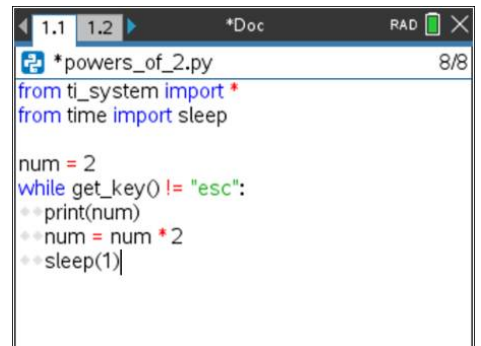
次にこれを試してください：`num = num * 2`を**`num *= 2`**に変更

このショートカットは同じことを行い、他の多くの数学演算子で機能します。



```
1.1 1.2 *Doc RAD 8/8
*powers_of_2.py
from ti_system import *
from time import sleep

num = 2
while get_key() != "esc":
    print(num)
    num = num * 2
    sleep(1)
```



```
1.1 1.2 *Doc RAD 8/8
*powers_of_2.py
from ti_system import *
from time import sleep

num = 2
while get_key() != "esc":
    print(num)
    num = num * 2
    sleep(1)
```

Teacher Tip: `get_key()` と `sleep()` は、Pythonプログラミングを非常に面白くする追加の関数です。`time`は、他の時間関係の関数を持つPython標準モジュールです。`ti_system`は、TI-Nspire CX IIオペレーティングシステムでのみ動作するように設計されたTexas Instrumentsモジュールです。