## Unit 3: 条件，if, while

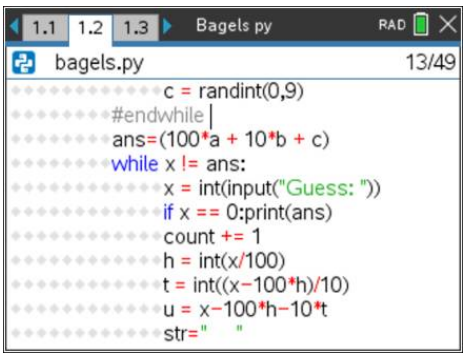
このレッスンでは，分岐(branch)による条件付きプログラミングを扱います。扱う内容はまだ証明されていないことです。

## Skill Builder 1: Collatz(コラッツ)予想

### 目標

- 関係演算子と論理演算子
- 算術演算子(%，//)
- ifステートメントとwhileループを使ったプログラム作成

---




プログラミングにおける意思決定プロセスは，条件に依存する**if**ステートメントと**while**ループで処理されます。右図は，**if**と**while**の例です。

条件は，値が**True**または**False**になる式です。

**X > Y**　　**A+B <= C**　　**Qty > 0**　　**5 != 3**

すべて条件の例です。　　(等しくない)

条件には，次の関係演算子の1つ以上が含まれます。

**==**　　**>**　　**<**　　**!=**　　**>=**　　**<=**

複合条件は，論理演算子を使って作成します。

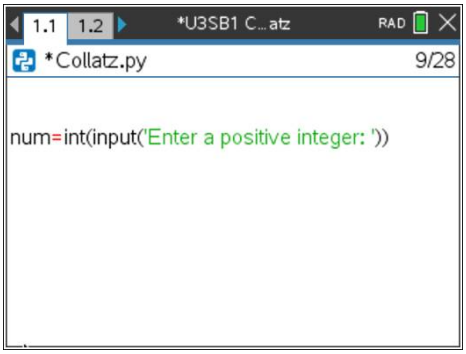
**and**　　**or**　　**not**

条件は，ifステートメントとwhileループで使用されます。

**Note:** 条件を書くときは，=ではなく==を使います。**x==5:**のところを**x=5:**と書いてはいけません(**ctrl+=**メニューを使うと便利です)。間違った記号を使うと，構文エラーが発生します。

このレッスンでは，これらの強力なツールを紹介します。

**Teacher Tip:** イコールは==を使い，“等しくない”は!=を使います。
このレッスンでは，2つの新しいPython算術演算子%と//を紹介します。
**a % b**は，aをbで割ったときの整数の余りを求めます
**a // b**は，aをbで割った整数の商を求めます。int(a/b)と同じです。




1. コラッツ予想

アルゴリズム：1つの正の整数を取り上げます。
偶数の場合は，2で割ります。
奇数の場合は，3を掛けて1を足します。
その結果を繰り返します。
最後にはどうなりますか。

空白のPythonファイル(名前Collatzを付けます)から始めます。

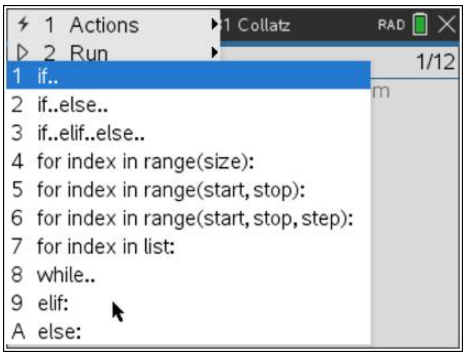
次のステートメントを使って，変数numに整数を入力します。

```
num = int( input( "Enter a positive integer: "))
```

**int(**は，**menu > Built-ins > Type** (メニュー>組み込み>タイプ)から選択します。
**input(**は，**menu > Built-ins > I/O** (メニュー>組み込み>I/O)から選択します。

**Note:** キーボードから入力しても構いません。”：”は英字Gの右側のキーを押して選択します。空白は文字Zの右横のキーです。

2. Ifステートメントには，if..，if..else..，if..elif..else..の3つステートメントがあります。これらはすべて，**menu > Built-ins > Control** (メニュー>組み込み>制御)で表示される一覧にあります。Pythonには'then'がないことに注意してください。




(これらのステートメントはプログラムのフローを制御する**Control**メニューにあります。)

**If..** "その他"のアクションがない場合に使います。

**if..else..** **True**と**False**の2つのアクションがある場合に使います。
(これはすぐに使います。)

**if..elif..else..** 条件に基づいて実行するアクションが3つ以上ある場合に使います。

**elif**は"elseif…"の略で，**if**のような条件が必要です。

必要な数の**elif**を追加できます。
(このユニットの応用で使います。)

**Teacher Tip: if**, **elif**, **else**の最後にあるコロン(:)に注意します。これらは必須であり，条件がTrueまたはFalseのとき実行されるアクションが：以下であることを示しています。BooleanExprと，インデントされた各ブロックは，アクションに置き換えられます。
**else:** 条件を取りません。
**else:** と **elif:** Controlメニューでは単独でも利用可能ですが，**if.**と組み合わせて使う必要があります。

**True**と**False** (大文字で表記されています)はPythonの予約語(ユーザーやプログラム開発者が自分で付ける識別名としては使えない特定の文字列)です。
**while True:**は無限ループを作成することができます。無限ループから抜け出せなくなったプログラムを中断するには，次のようにします。
Handheld: ON　　Windows: [F12] or Fn+[F12]　　Mac: [F5]

3. **menu > Built-ins > Control** (メニュー>組み込み>制御)から**if..else**ステートメントを挿入します。
**tab**または右矢印キーを押して，プロンプトからプロンプトに移動します。

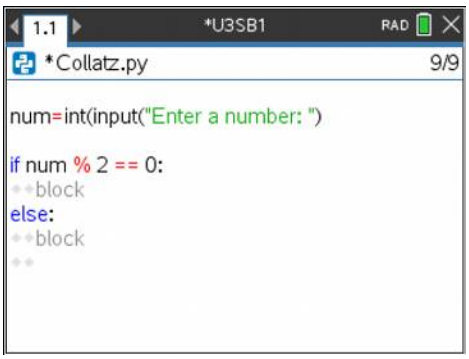

**Teacher Tip:** ステートメントを自分で入力した場合，インラインプロンプト(BooleanExpr, 各ブロック)は表示されません。
プロンプトからプロンプトへの移動は，**tab**を押します。

4. 条件(BooleanExpr)は，

   **if num % 2 == 0**:

   **%**はmodと呼ばれる数学の演算子で(+，-*，/と同じ)，入力された数値を%の後ろの数値(ここでは2)で割ったときの余りが得られます。modはmodulusの略です。
   **%**は句読点キー(文字Gの横)にあります。

   numを2で割ったときの余りがゼロの場合，numは偶数です。
   2つの等号に注意します。
   ==は=を2回押すだけです。すべての関係演算子は**ctrl+=**にあります。

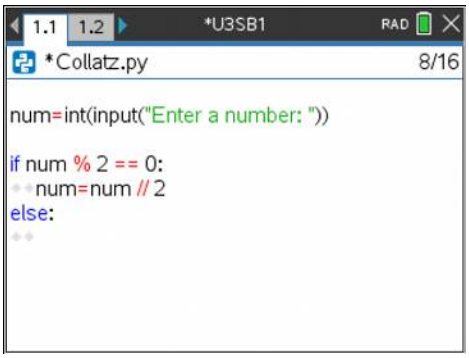



**Teacher Tip:** Pythonにはthenがないことに注意してください。スペースの無駄遣いです。

5. **if** : ( 数が偶数のとき)
   block は，以下のようになります。

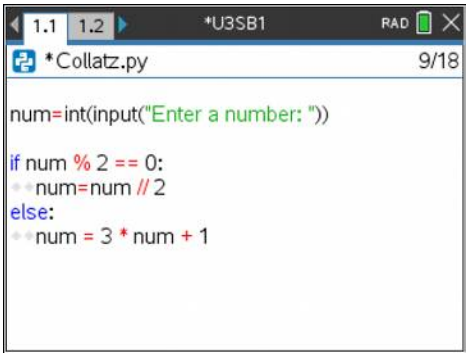
   **num = num // 2**

   // (2つの除法記号)は整数の除算です(結果は10進数ではなく整数のみになります)。/を使うと，数値が整数であっても小数点が表示されます。2つの/記号(割る÷キー)を使います。



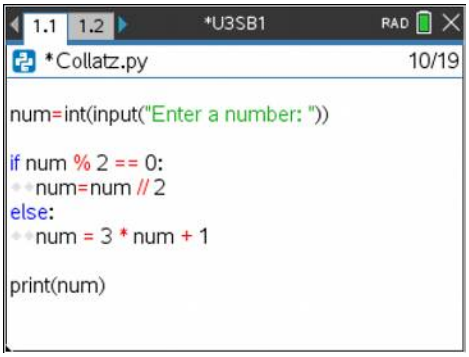

6. **else:** (数が奇数のとき)
   block は，以下のようになります。

   **num = 3 * num + 1**




7. **if..else:** ステートメントブロックの後に，行の先頭にバックスペースして(インデント(字下げ)文字を消去し)，**print**ステートメントを記述します。

   **print(num)**



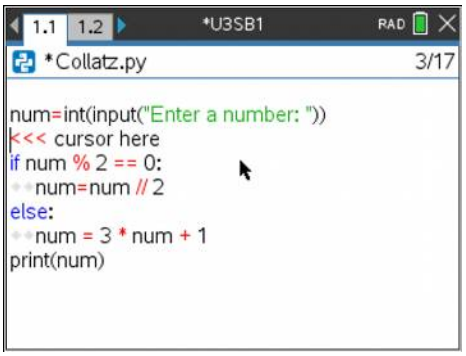
8. プログラムの実行


**ctrl+R**を押して，プログラムを実行します。正の整数を入力してください。答えが表示されます。
もう一度**ctrl+R**を押して，表示された計算結果を今度は入力します。同じようにプログラムの実行を繰り返します。

いつまでも**ctrl+R**で実行を繰り返すのは大変ですから，プログラムにループを追加して，繰り返し実行されるようにします。

**Teacher Tip:** 何が起こっていますか。数値は最終的に1になり，パターンは1, 4, 2, 1, 4, 2, 1…を繰り返すだけです。

9. プログラムはシェルページ(ここでは1.2)で実行されます。エディタページ(ここでは1.1)に**Ctrl+◀**で戻ります。
右図で示すように，カーソルを**if**ステートメントの上**input**ステートメントのすぐ下に置きます(”<<< cursor here” と入力しないでください)。
**Note:** ifの先頭でもかまいません。

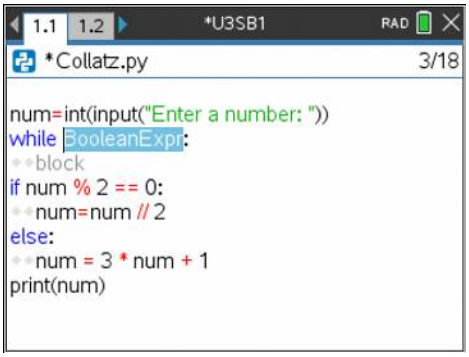



10. この空白行に**while**ステートメントを次の手順で追加します。

**menu > Built-ins > Control** (メニュー>組み込み>コントロール)

次のようにプログラムに貼り付けられます。

```
while BooleanExpr:
    block
```

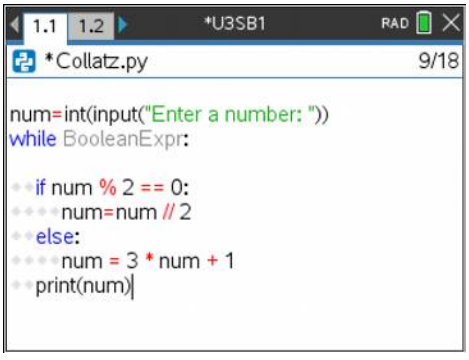



**Teacher Tip:** ‘block’はインデントされていますが，実際のブロックはすでにプログラムに含まれています。**If**ステートメントと**print**ステートメントです。次のセクションでは，テキストのセクション全体をインデントする方法を示します。

11. ‘block’と書かれている行全体を消去します。

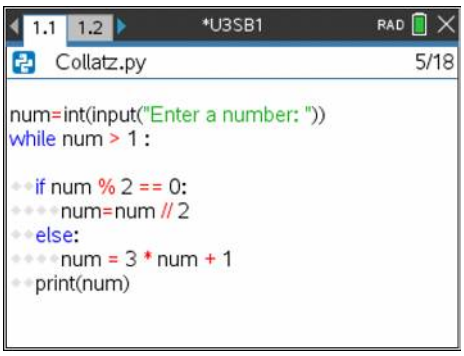
**shift+下矢印▼** (**Shift**を押しながら下矢印カーソル)を使って，**if**文ステートメントと**print**ステートメントを選択します。**Tab**キーを押して，**while**ブロックにするため，これらすべての行を2スペースインデントにします。

**Note:** 選択した範囲の行の頭に◆◆が2つ挿入されます。




**Teacher Tip: tab**を押す別のオプションは，**menu > Edit > Indent** (メニュー>編集>インデント)です。**Dedent**(デデント)はインデントの反対であり，そのショートカットは**shif+tab**です。

12. BooleanExprを置き換えて，条件を記述します。
コラッツの予想は，すべての数列が最終的に1になると述べています。数値が1より大きい限り，処理を続行します。よって，以下のように記述します。




while **num > 1 :**

**>**は**ctrl+=**で選択します。
行末に必ずコロン(:)を残してください。

13. プログラムを実行します。数値は20を入力します。以下プログラムのロジックに従います。奇数は大きくなり，偶数は小さくなります。


20 は偶数 → 10
10 は偶数 → 5
5 は奇数 → 16
16 は偶数 → 8 など…
…プログラムは数値が1に達すると終了します。
ループを作成するのに1行だったことに注意します。

プログラムが終了しない数値を見つけることができますか。大きな数値を試してみてください。数値が収束する速さに注目してください。シェル履歴を上にスクロールすれば，数値の変化を調べることができます。

**Teacher Tip:** シェル履歴はプレーンテキスト(文字だけで構成され，レイアウト情報や装飾情報などを持たないデータ)であり，TI-Nspireドキュメントと一緒には保存されません。シェル履歴の一部を保存したいときは，選択してコピーし，Notesアプリに貼り付けます。シェル履歴をクリアするには，**menu > Tools > Clear History**(メニュー>ツール>履歴クリア)を選択します。シェル履歴をクリアするプログラミングコマンドもあります(画面クリア機能と同様)。

コラッツの予想(1937年に提示)はまだ証明されていません。アルゴリズムの3，1，2の数値について何か特別なことはありますか。別の数値をやってみてください。
プログラムに追加するもう1つのオプションは，1に到達するまでのステップ数をカウントすることです。