


cs-10. Python プログラミングの基本

(コンピューターサイエンス)

URL: <https://www.kkaneko.jp/cc/cs/index.html>

金子邦彦



- 
- ① **プログラミングの基礎概念**：オブジェクト、メソッド、引数、変数、代入
 - ② **実践的な演習**. CodeCombatを用いたPythonプログラミングの演習. タートルグラフィックスを使った図形描画の演習
 - ③ **プログラミングの楽しさと達成感**. プログラミングのクリエイティブな側面を体験
 - ④ **プログラミングの基本的な知識とスキルの習得** (モジュールのインポート、オブジェクト生成、メソッドの使用など)

アウトライン

1. Python プログラミングの基礎
2. コードコンバットを用いたプログラミング演習
3. オブジェクト, メソッド
4. プログラミングの楽しさと達成感

- **プログラミング**は**人間の力を増幅**し、私たちができることを大幅に広げる
- **プログラミング**はさまざまな分野で活用されている
 - シミュレーション：複雑な現象をモデル化し、予測や分析を行います
 - 大量データ処理：データの収集、加工、分析
 - AIシステムの開発
 - Webサイト, アプリケーションなどのソフトウェア
- **プログラミング**は**クリエイティブな行為**
- さまざまな**作業を自動化**したいとき、**問題解決**したいときにも役立つ

プログラミングの楽しさと達成感



• 楽しさ

- **未来の技術を学ぶことは楽しい。**
- **プログラミングは自分のアイデアを形にできるクリエイティブな行為。**
- **視覚的なプログラムを書くことで、ゲーム感覚をもって楽しみながら学習することも可能。**

• 達成感

- **プログラミングを通じて問題解決のスキルを身につけることは、大きな達成感につながる**
- **新しいソフトウェアや技術を生み出す。**

プログラミングの基礎



本講義では、プログラミングの基礎として以下の内容を説明

- オブジェクトとメソッド、引数、変数、代入
本資料で説明
- クラス、条件分岐、繰り返し、リスト、パッケージ
今後の講義で説明予定

10-1. Python プログラミングの基礎

変数, 代入

- **変数** : プログラム内で名前を付けて利用する オブジェクト で, 値を保存 し, 後から 参照 できる
- **代入** : 「 $x = 100$ 」のように書くことで, x という名前の変数に、値 100 が保存 される

例

```
x = 100
```


オブジェクトとメソッド



- **オブジェクト**：コンピュータでの**操作や処理の対象となるもの**

t.goto(0,100)

t オブジェクト

goto(0,100) メソッド

間を「.」で区切っている

- **メソッド**：**オブジェクト**に属する機能や操作. オブジェクトがもつ能力に相当する
- **引数**：**メソッド**が行う操作の詳細に関する情報. **メソッド**呼び出しのときに、引数を指定できる

t.goto(0,100)



引数

10-2. コードコンバット (Code Combat) を用いた プログラミング演習

コードコンバット



コードコンバット (CodeCombat) では、**ゲームをクリアするために必要なプログラムを書く**ことで、Pythonの基本を楽しく学ぶことができる

The screenshot displays the CodeCombat game interface. On the left, a character named ANYA is in a dungeon level titled "キースガードのダンジョン". The objective is "トゲをよける 宝石を集める" (Avoid thorns, collect gems). The character is positioned near a blue gem. On the right, a code editor shows Python code for moving the character. The code is as follows:

```
プログラミング言語: Python
1 # 宝石まで移動させよう！
2 # 壁のトゲに当たったらダメだぞ！
3 # 下にコードを打ち込んだらRunボタンを押そう
4
5 hero.moveRight()
6 hero.moveDown()
7
```

Below the code editor, there is an "実行" (Execute) button and a list of available movement methods:

- hero.moveDown(steps)
- hero.moveLeft(steps)
- hero.moveRight(steps)
- hero.moveUp(steps)

The character's health bar at the bottom shows 11 HP.

コードコンバットの授業の範囲



- **無料、クラスコード無し：**

Python の5つのレベルのゲームなど

授業の範囲内

- **有料（会員登録）：**

500以上のレベル

← **興味のある人のみ（各自の判断）**

- **無料、クラスコードあり：**

Python のより多くのレベルのゲーム

（各自で、名前、メールアドレス等の登録が必要です）

← **挑戦は、興味のある人の自習とします（各自の判断）**

この授業では、会員登録しない、クラスコードも使わないとして説明する

コードコンバットの最初の5つのレベルのトピックス



- オブジェクトとメソッド

`hero.moveDown()`

hero は**オブジェクト**

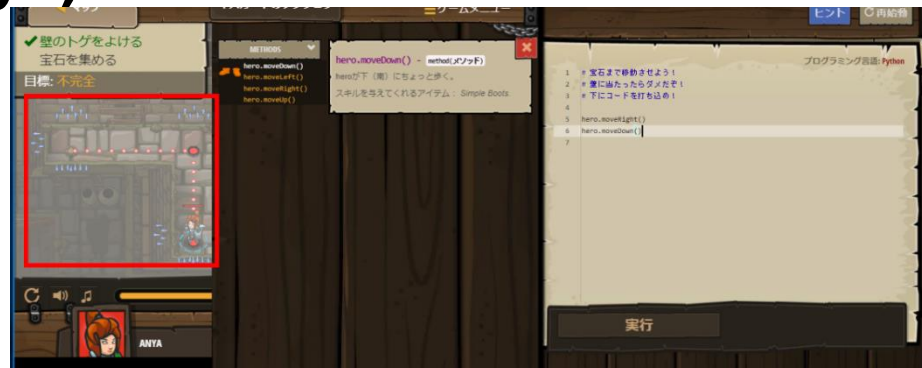
`moveDown()` は**メソッド**

- **文字列**は「"」(ダブルクォーテーション)または「'」(シングルクォーテーション)で囲む

- **メソッド**呼び出しでは, **引数**を指定できる

`hero.attack("fence", 36, 26)`

`hero.attack(enemy1)`



オブジェクトとメソッド



オブジェクトが動く



実行画面

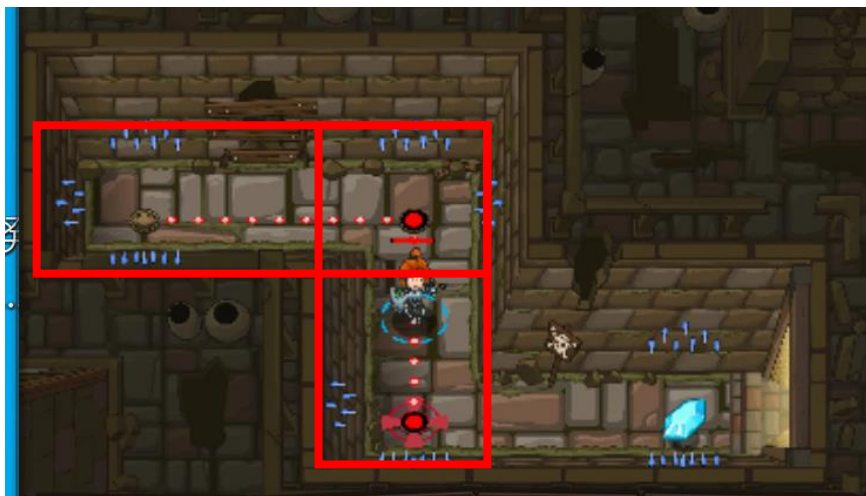
```
1 # 宝石まで移動させよう！  
2 # 壁に当たったらダメだぞ！  
3 # 下にコードを打ち込め！  
4  
5 ▶ hero.moveRight()
```

オブジェクトとメソッド
(Python プログラム)

オブジェクトとメソッド



オブジェクトが動く



実行画面

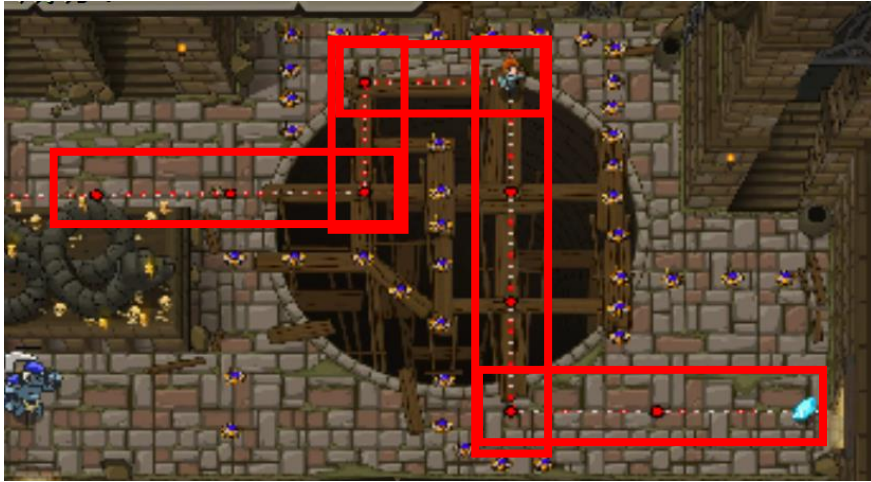
```
1 # 宝石まで移動させよう！  
2 # 壁に当たったらダメだぞ！  
3 # 下にコードを打ち込め！  
4  
✓ 5 hero.moveRight()  
▶ 6 hero.moveDown()
```

オブジェクトとメソッド
(Python プログラム)

引数



オブジェクトが動く



実行画面

```
1 # Use arguments with  
   farther.  
▶ 2 hero.moveRight(3)  
3  hero.moveUp()  
4  hero.moveRight()  
5  hero.moveDown(3)  
6  hero.moveRight(3)
```

オブジェクトとメソッド
(Python プログラム)

引数がある場合もあれば、
ない場合もある。

演習 プログラミングを行う

【内容】

プログラミング学習サイト
CodeCombat 無料の機能、クラス
コード無しで、各自、プログラミ
ングを行う

【トピックス】

- オブジェクト
- メソッド
- 引数
- 文字列

① Web ブラウザを使う



<https://codecombat.com>

「課金のメッセージ」などで心配なことがあるときは、無理に使い続けないこと



課金のメッセージの例

② 「私は学生です」をクリック

- 「日本語」になっていない場合には，日本語に変える
- アカウント登録やログインは行わないことにする



The screenshot shows the Code Combat website homepage. At the top left is the 'CODE COMBAT' logo. To its right are navigation links: '教育者' (Educator), '親' (Parent), and 'Play'. Further right are 'ログイン' (Login) and 'アカウント登録' (Account Registration) buttons. A red box highlights the '日本語' (Japanese) language selector, with a red arrow pointing to it from the right. Below the navigation is a teal banner with the text: '遊びによって推進されるライブオンラインのコーディングクラス！ 無料の体験クラスにSign up！'. The main heading reads '遊びの力を通じてコーディングを学ぶ' (Learn coding through the power of play). On the left side, there are three yellow buttons: '私は教育者です' (I am an educator), 'I'm a Parent', and '私は学生です' (I am a student), with the last one highlighted by a red box. The background features a stylized blue and green pixelated landscape with clouds and a computer monitor displaying a game interface. In the bottom right, there is a photo of three young girls looking at a screen. A small robot character is visible in the bottom left corner.

クラスコードを持っているか



- 「今すぐプレイ」をクリック

クラスコードを持っていますか？

持ってる！

ここに入力しましょう:

持って無いけどやりたい！

すでにアカウントをお持ちですか？ [サインイン](#)

③ キースガードのダンジョンを選んでみる。 「ゲームスタート」をクリック



CODE COMBAT

ゲーム開発 0/21
ロック
自分でレベルを造る事を習おう！

キースガードのダンジョン 0/54
ゲームスタート
1～3時間：文法,メソッド,パラメータ,文字列,ループ,変数

ウェブ開発 0/12
ロック
HTMLをもっと学ぼう！

辺境の森林 0/130
ロック
2～6時間：if/else（条件分岐）、関係演算子、オブジェクト・パラメータ、入力処理

ウェブデベ 0/...

無料で教師アカウントを作成

0 プレイヤーレベル 1 名無しのプレイヤー

④ 「キースガードのダンジョン」の最初のダンジョンを選ぶ



音声が出るので、このとき、各自で「音量」を調整



ゴール

最初のダンジョン

⑤ 「ゲームスタート」をクリック



CODE COMBAT

キースガードのダンジョン 0/44

CODECOMBATには 教室版!

クラスコードを入力:
持っていない? 先生に聞こう!
 [Join](#)

クラスを作りたい?
クラスを設定し生徒を追加して、進歩を見ましょう!
[無料の教師アカウントを作成](#)

キースガードのダンジョン
宝石を集めよう! ただし、壁の針にはぶつからないように。ここでは、主人公を動かす基本的なプログラムをマスターしよう!

Bas

ゲームスタート

0 プレイヤーレベル 1 名無しのプレイヤー ログイン アカウント登録 日本語

⑥ 「Python (デフォルト)」を選び、「次へ」をクリック



ヒーロー選択

キャプテンアーニャ・ウェストン

アーニャはで行くと、それは鬼ベースを襲撃、村を守る、または光沢のあるものを拾うのかどうか、ジョブが、行われ得ることを恐れないできまじめなリーダーです。

難易度	イージー
タイプ	戦士
武器	剣 - 攻撃範囲:短い- 魔法使用不可
攻撃力	
体力	
移動速度	

プログラミング言語: PYTHON (デフォルト) ▼

Code Format: TEXT CODE (デフォルト) ▼

次へ

⑦ 使用可能なアイテムを選ぶ (ダブルクリック)



⑧ 「ゲームスタート」をクリック



The screenshot shows the 'CODE COMBAT' inventory interface. At the top left, the text 'CODE COMBAT' is displayed. Below it, the title 'インベントリー' (Inventory) is shown. A blue gem icon with the number '0' and a red 'X' icon are visible. The main area features a character model on the left, a 'ロック' (Lock) section with three items, and a 'シンプルなお靴' (Simple Shoes) section with a pair of shoes. Below the character model, there are two buttons: 'ヒーローやプログラミング言語を変える' (Change hero or programming language) and 'ゲームスタート' (Game Start), which is highlighted with a red box. To the right of the shoes, there is a text box with instructions: '主人公を上、下、右、左に動かすコマンドが使えるようになる。' (You will be able to use commands to move the hero up, down, right, and left.) and a list of skills: '追加されるスキル' (Additional skills), 'moveDown: heroが下(南)にちょっと歩く。' (moveDown: hero moves a little south), 'moveLeft: heroが左(西)にちょっと歩く。' (moveLeft: hero moves a little west), 'moveRight: heroが右(東)にちょっと歩く。' (moveRight: hero moves a little east), and 'moveUp: heroが上(北)にちょっと歩く。' (moveUp: hero moves a little north). Below this text is a '装備を外す' (Remove equipment) button. On the right side of the screen, there is a sidebar with a 'CODECOMBATには教室版!' (Classroom version for CODECOMBAT!) section, a 'Join' button, and a '無料の教師アカウントを作成' (Create free teacher account) button. At the bottom of the screen, there is a status bar with '0 プレイヤーレベル 1' (0 player level 1), '名無しのプレイヤー' (Nameless player), 'ログイン' (Login), 'アカウント登録' (Account registration), '日本語' (Japanese), and several icons for equipment, a chest, and a trophy.

⑨ 「レベルスタート」をクリック



目標

壁の針をよける
宝石を集める



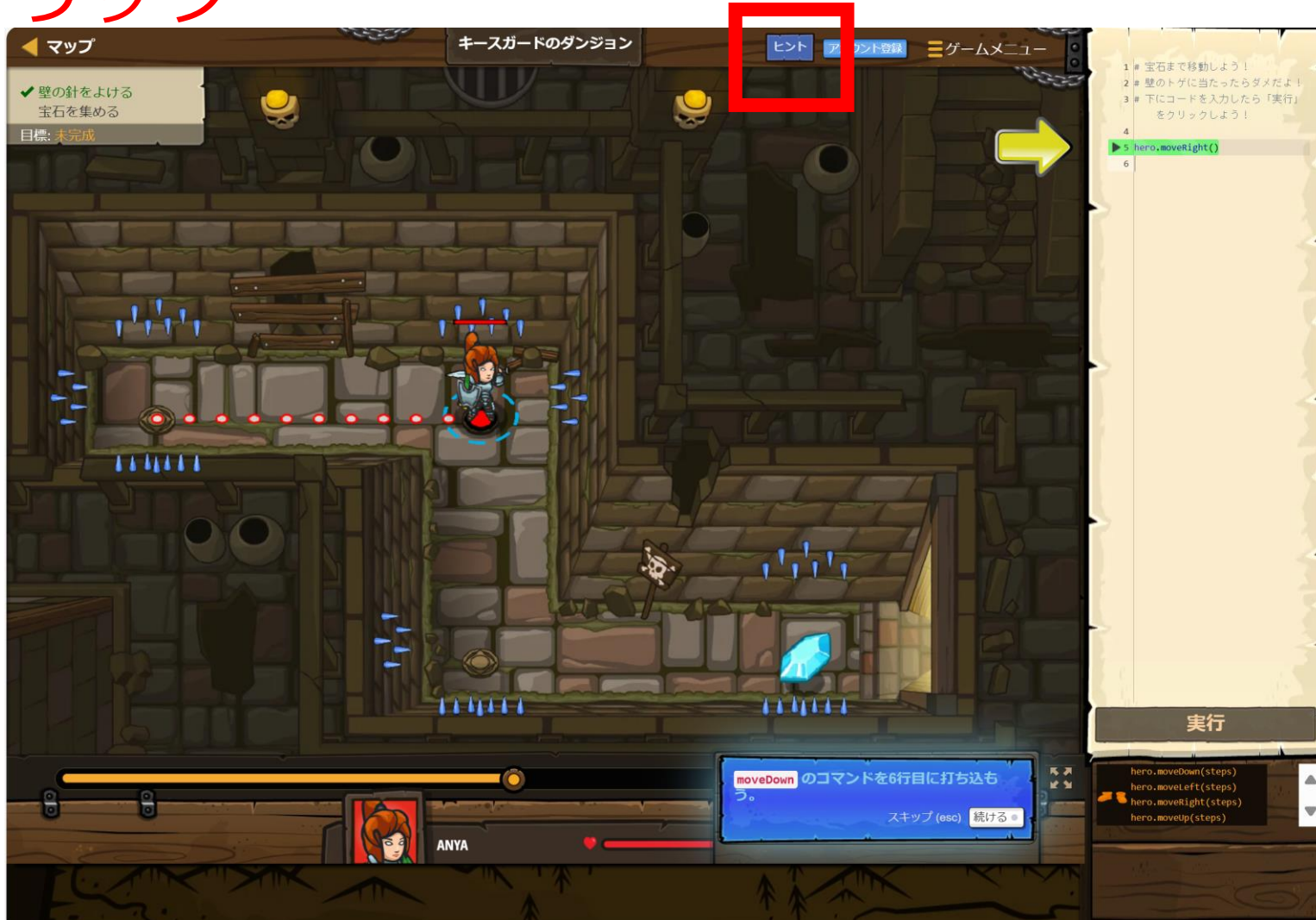
コーディングしたプログラムで、ヒーローを導け！

右側のエディターにコードを書こう。目標をクリアできるコードが書けたら、「実行」ボタンをクリックしよう。ヒーローがコードを読みとって、書かれた指示にしたがって動くぞ。

壁の針に触れないように、ヒーローを通路に沿って動かそう。

レベルスタート

ヒントを見たいときは、「ヒント」をクリック



但し、英語で表示される場合があります。
翻訳が完全でないためです。設定の不備ではありません。

メソッドの説明を見たいときは、「メソッド」のリストの中から、説明を見たいメソッドをクリック



マップ

キースガードのダンジョン

ヒント アカウント登録 ゲームメニュー

壁の針をよける
宝石を集める
目標: 未完成

hero.moveDown(steps) - method
heroが下（南）にちょっと歩く。
Example:
hero.moveDown(2)
Required Parameters:
steps | number (ex: 2)
How many steps to move (default 1 step, 12 meters).
Default value: 1
Granted by Simple Boots.

実行

hero.moveDown(steps)
hero.moveLeft(steps)
hero.moveRight(steps)
hero.moveUp(steps)

但し、英語で表示される場合があります。
翻訳が完全でないためです。設定の不備ではありません。

⑩ 編集画面で、試しに、
「hero.moveDown()」と追加して、「実行」
をクリック



```
プログラミング言語: Python  
1 # 宝石まで移動させよう！  
2 # 壁に当たったらダメだぞ！  
3 # 下にコードを打ち込め！  
4  
5 hero.moveRight()  
6 hero.moveDown()  
7
```

実行

⑪ 「実行」で、キャラクタが動くので確認する



hero.moveRight() で右に動き、hero.moveDown() で下に動く

The screenshot shows a game interface for a dungeon named "キースガードのダンジョン". The main area displays a character named ANYA in a stone-walled dungeon. A red box highlights the character and a red dot on the floor, indicating movement. The right side features a code editor with the following code:

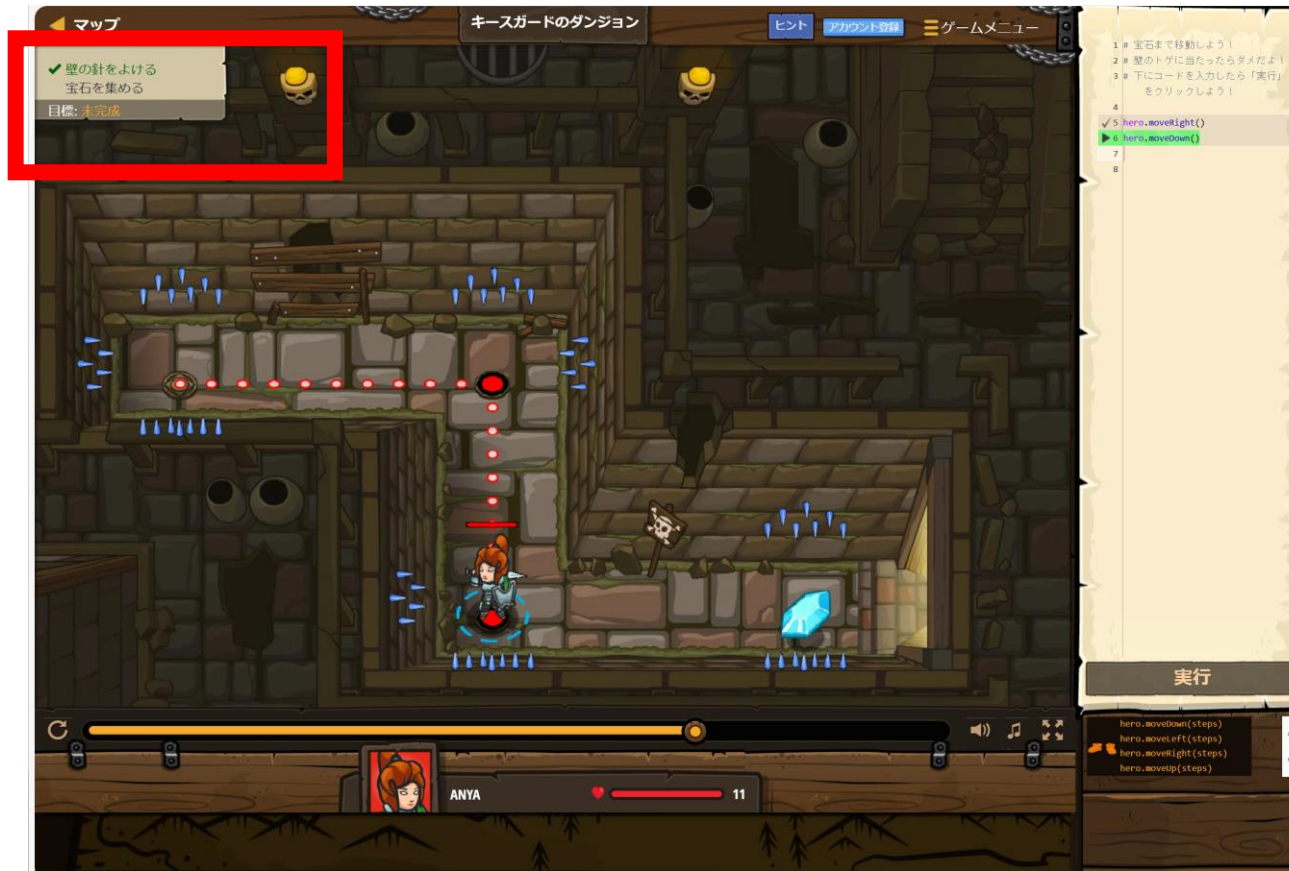
```
1 # 宝石まで移動しよう！  
2 # 壁のトゲに当たったらダメだよ！  
3 # 下にコードを入力したら「実行」をクリックしよう！  
4  
5 hero.moveRight()  
6 hero.moveDown()  
7  
8
```

Below the code editor is a button labeled "実行" (Execute). At the bottom, the character's name "ANYA" and a health bar with the value "11" are visible.

迷ったら、「ミッション（目標）」や「ヒント」を確認する

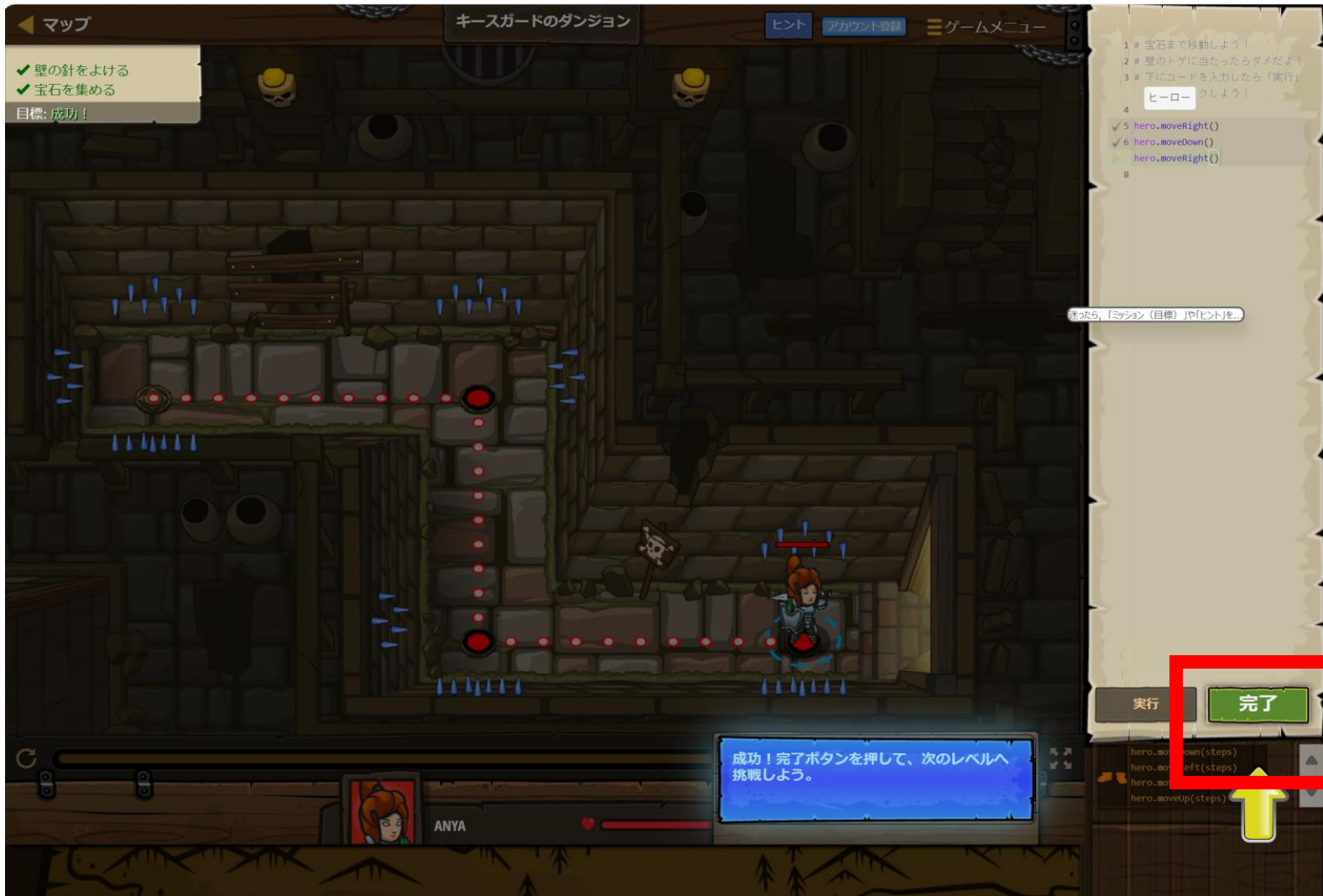
- プログラミングの練習だけでなく、**ゲーム要素（パズル）**もある。楽しんで解く。
- ヒントや説明が、英語で表示される場合がある

すべての
目標達成
を目指す



⑫ 「目標：成功！」になるまで、編集画面を
書き換えて、「実行」を繰り返す。

成功したら、「完了」をクリック



⑬ 完了の確認。「続ける」をクリック



続けてみる

- 赤い旗を選択できる
- 有料のものもある



赤い旗をたどる



青い旗はクリア済み



「(ロック)」と表示される
場合は有料

新しい装備が増える場合がある

ダブルクリックして装備



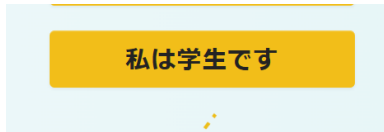


無料で、クラスコードを使わずに、Python
の5つのレベルを学ぶことができる

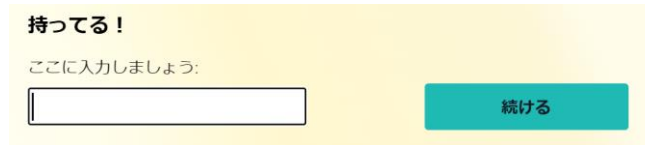
(続きは有料となる。詳しくは次ページ)

次のことは、各自の判断とする（このことは、授業の成績に関係しない。）

1. CODECOMBATプレミアムへの登録（有料）し，ゲームを続ける
2. セレッソの記載のクラスコードで，本格的に使う（無料で学んでいけます）



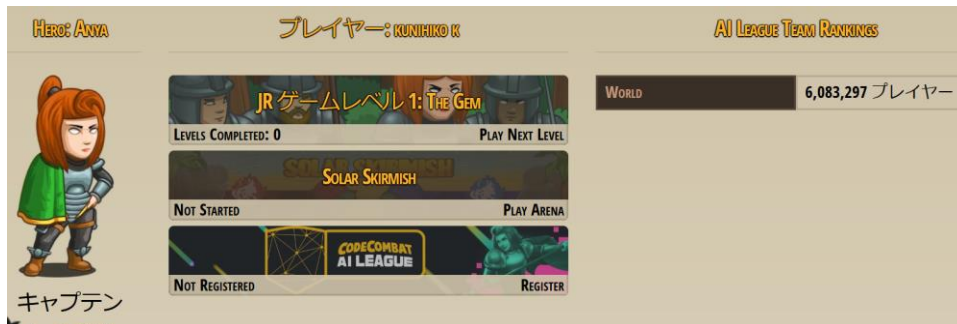
私は学生です



セレッソのクラスコードを入れる



メールアドレスなどの登録が必要。ユーザ名，パスワードは自分で決める



10-3. オブジェクト, メソッド

オブジェクトとメソッド



- **オブジェクト** : コンピュータでの 操作や処理の対象となるもの

t.goto(0,100)

t オブジェクト

goto(0,100) メソッド

間を「.」で区切っている

- **メソッド**: **オブジェクト**に属する機能や操作. オブジェクトがもつ能力に相当する
- **引数** : **メソッド**が行う操作の詳細に関する情報. **メソッド**呼び出しのときに、引数を指定できる

t.goto(0,100)



引数

タートルグラフィックス

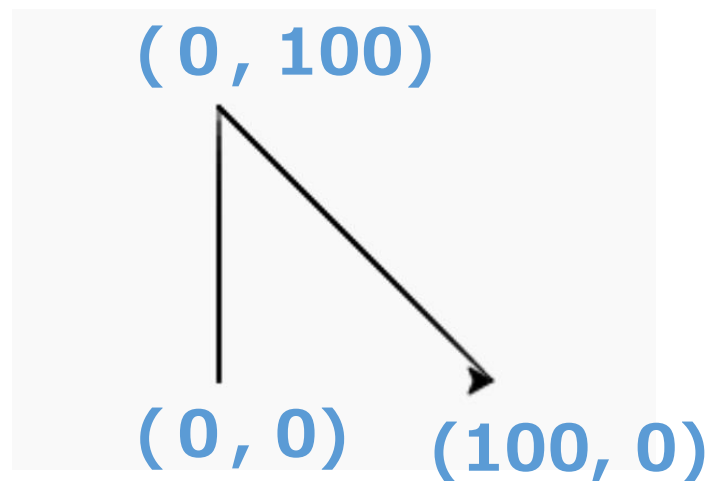


カーソルを使って絵を描く

- **タートルグラフィックス**を用いた演習により、**プログラムによって図形を描画**する。それを通して、プログラムの**動作を視覚的に理解**
- 論理的思考力や課題解決力の向上にもつながる

```
1 import turtle
2 t = turtle.Turtle()
3 t.goto(0,100)
4 t.goto(100,0)
```

タートルグラフィックスの機能をインポートする「import turtle」が必要



タートルグラフィックス入門



```
1 import turtle
2 t = turtle.Turtle()
3 t.goto(0,100)
4 t.goto(100,0)
```

オブジェクト **メソッド**

- **メソッド**は、オブジェクトが持つ機能を呼び出すためのもの
- 「**goto**」は**指定した座標への移動**

主なメソッド

- **goto** (<横方向の値> , <縦方向の値>) **移動**
- **forward**(<移動量>) **前進**
- **backward**(<移動量>) **後退**
- **right**(<角度>) **右回りに回転**
- **left**(<角度>) **左回りに回転**

オブジェクトとメソッド



```
import turtle
```

```
t = turtle.Turtle()
```

```
t.goto(0,100)
```

インポート

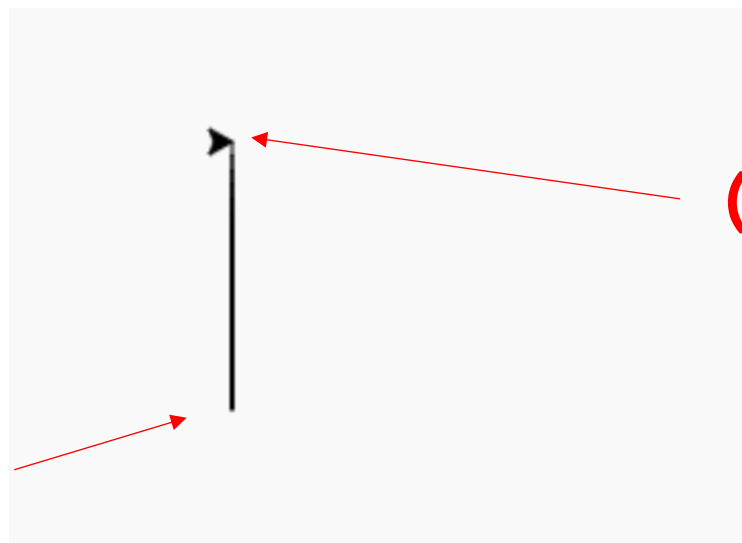
オブジェクト生成。t へのセット。

(0, 100) への移動

実行結果

**最初の位置は
(0, 0)**

(0, 0)



(0, 100)

オブジェクトが動く

オブジェクトとメソッド



```
import turtle
```

```
t = turtle.Turtle()
```

```
t.goto(0,100)
```

```
t.goto(100,0)
```

インポート

オブジェクト生成。t へのセット。

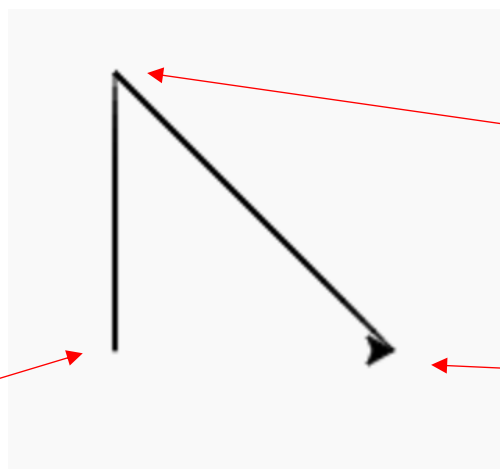
(0, 100) への移動

(100, 0) への移動

実行結果

**最初の位置は
(0, 0)**

(0, 0)

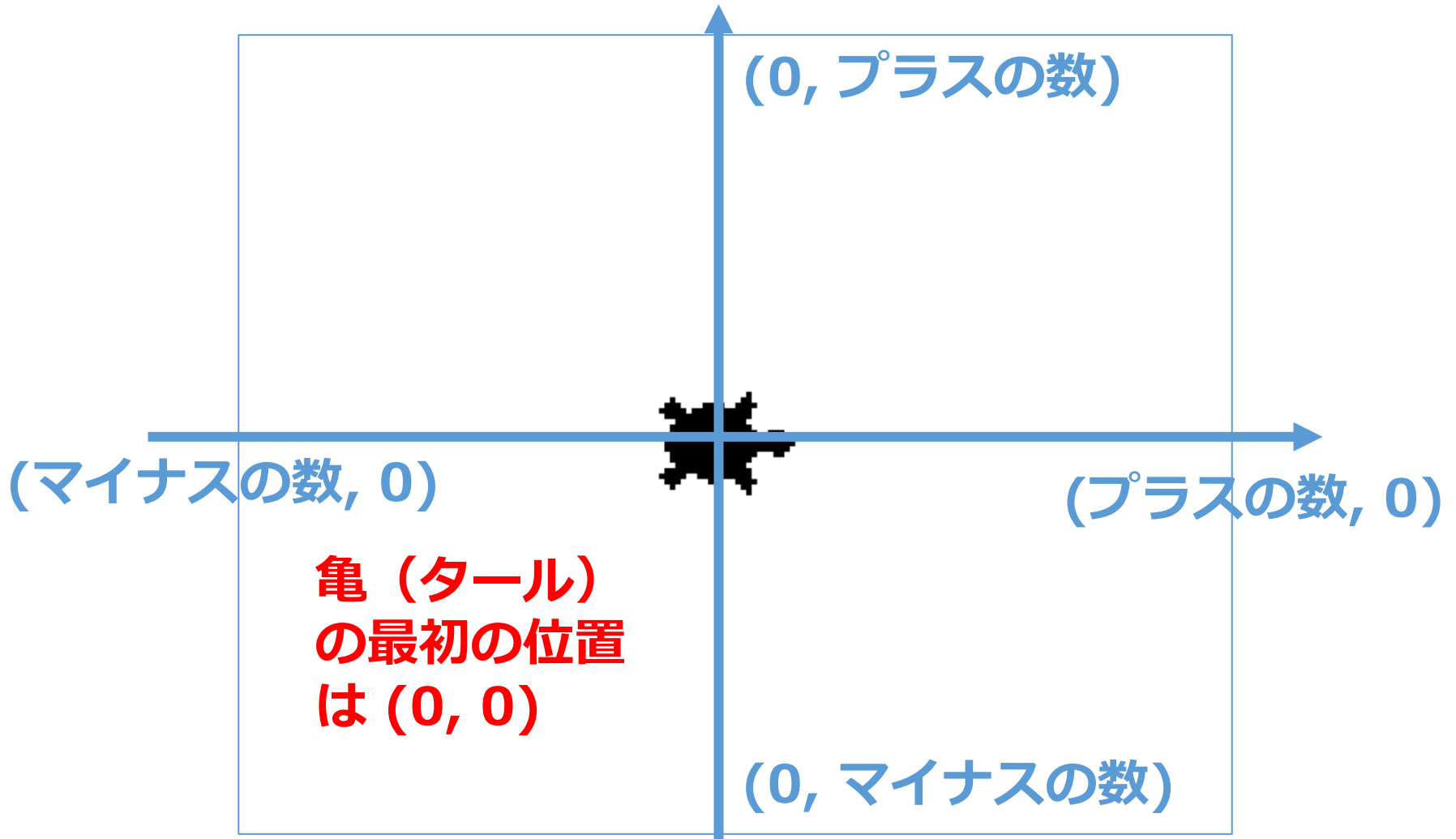


(0, 100)

(100, 0)

オブジェクトが動く

メソッド goto の引数となる 縦方向の値と, 横方向の値



- Trinket は**オンライン**の Python、HTML 等の**学習サイト**
- 有料の機能と無料の機能がある
- **自分が作成した Python プログラムを公開し、他の人に実行してもらうことが可能**（そのとき、書き替えて実行も可能）
- **Python の標準機能**を登載、その他、次のモジュールやパッケージがインストール済み

math, matplotlib.pyplot, numpy, operator, processing, pygal, random, re, string, time, turtle, urllib.request

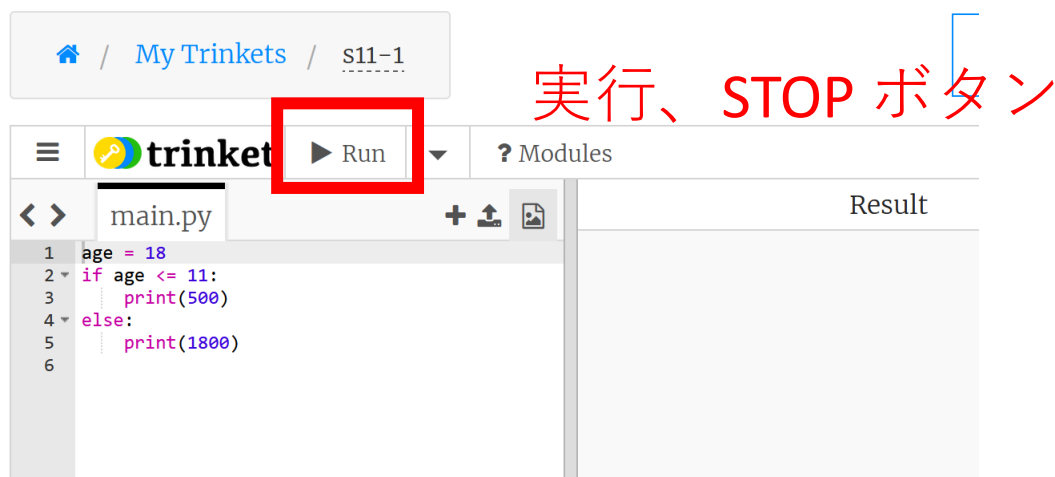


trinket でのプログラム実行

- trinket は Python, HTML などのプログラムを書き実行できるサイト

<https://trinket.io/python/cdc4896571>

のように、違うプログラムには違う URL が割り当てられる



ソースコードの
メイン画面

実行結果

- 実行が開始しないときは、「**実行ボタン**」で**実行**
- ソースコードを書き替えて再度実行することも可能



演習. タートルグラフィックス



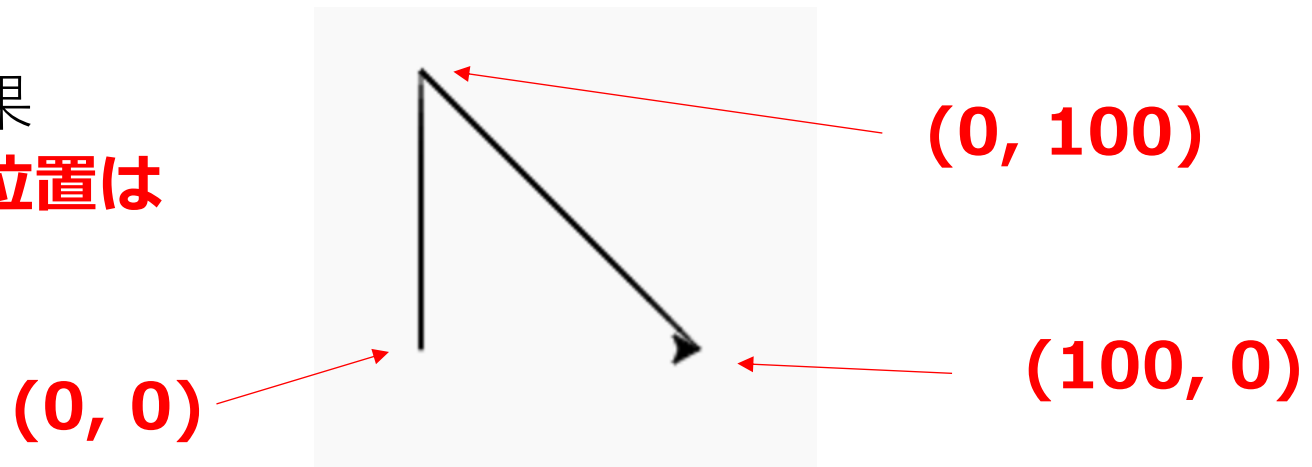

```
import turtle  
t = turtle.Turtle()  
t.goto(0,100)  
t.goto(100,0)
```

ライブラリのインポート
オブジェクト生成。t へのセット。

(0, 100) への移動

(100, 0) への移動

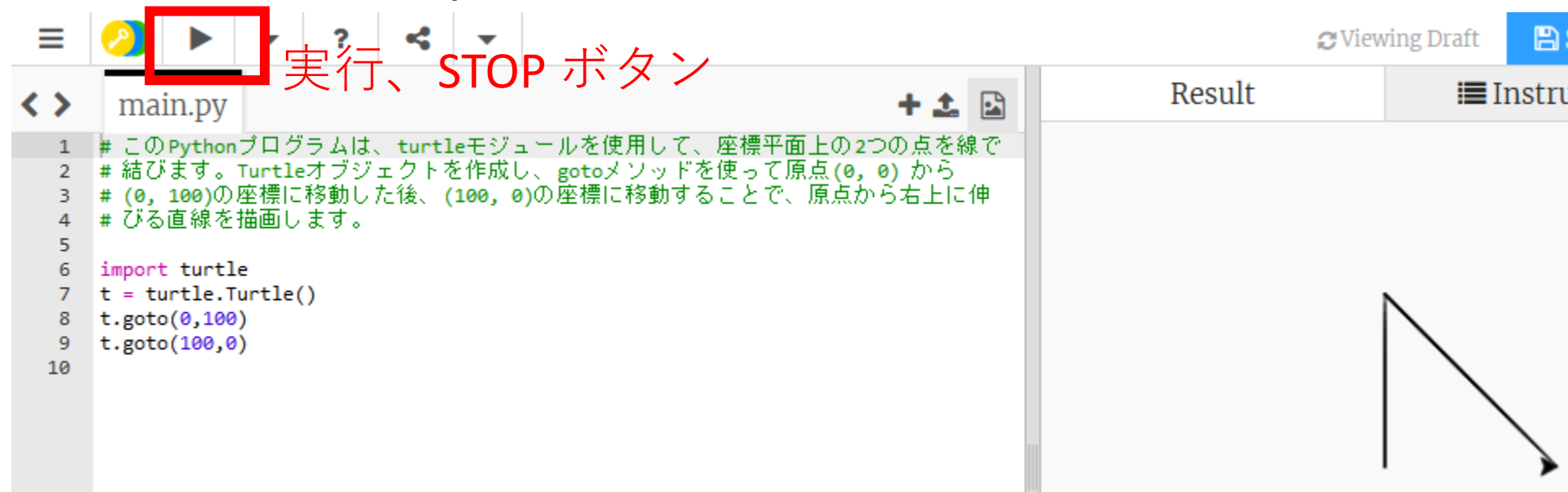
実行結果
**最初の位置は
(0, 0)**



① trinket の次のページを開く

<https://trinket.io/python/f29bfe71cd>

② 実行結果が、次のように表示されることを確認



実行、STOP ボタン

```
1 # このPythonプログラムは、turtleモジュールを使用して、座標平面上の2つの点を線で  
2 # 結びます。Turtleオブジェクトを作成し、gotoメソッドを使って原点(0, 0) から  
3 # (0, 100)の座標に移動した後、(100, 0)の座標に移動することで、原点から右上に伸  
4 # びる直線を描画します。  
5  
6 import turtle  
7 t = turtle.Turtle()  
8 t.goto(0,100)  
9 t.goto(100,0)  
10
```

Result

Instru

- 実行が開始しないときは、「**実行ボタン**」で**実行**
- ソースコードを**書き替えて再度実行**することも可能



③ 次のように書き換えて、再実行し、結果が変わることを確認

```
import turtle  
t = turtle.Turtle()  
t.goto(0,100)  
t.goto(200,0)
```

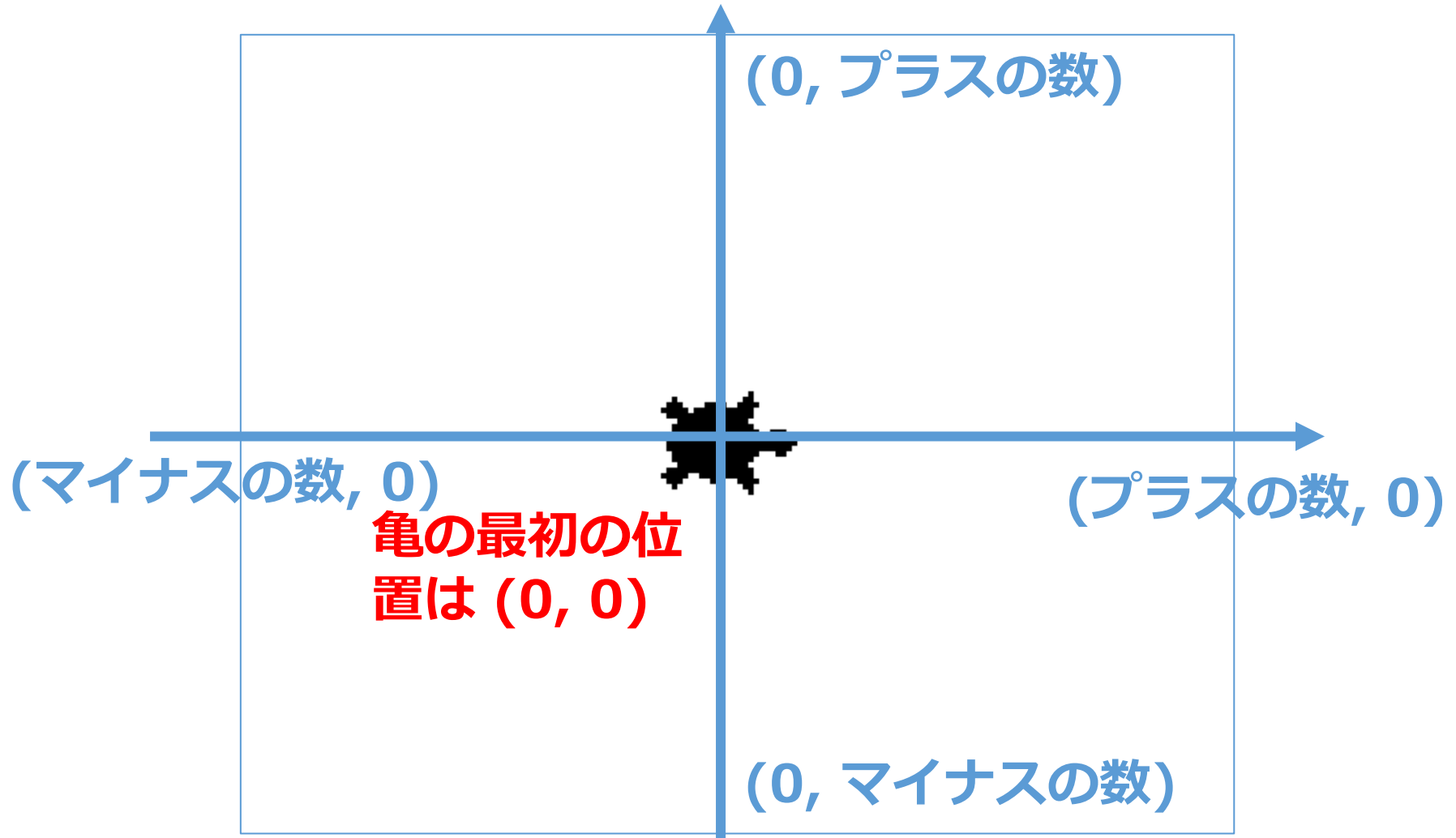
← 書き換え

実行、STOP ボタン

```
main.py  
1 import turtle  
2 t = turtle.Turtle()  
3 t.goto(0,100)  
4 t.goto(100,0)  
5
```

Result

④ 次を参考に、自分で引数を書き替えたり、プログラム内に「t.goto(<引数>)」を増やして、思い通りの図形を目指す。



ここまでのまとめ

- **変数**：プログラム内で名前を付けて値を保存し参照できるオブジェクト
- **オブジェクト**：コンピュータでの操作や処理の対象となるものである。
- **メソッド**：オブジェクトが持つ機能や操作を表すもの。
例: `t.goto(100,0)`では `t` はオブジェクト、`goto` はメソッド。
- **引数**：メソッドには、操作の詳細を指定するための引数を設定できる。
例： `t.goto(100,0)` では、引数は「100,0」
- Python では、**インデント**を適切に使用する。
- プログラムの修正や改良を繰り返し行うことで、目的の動作を実現できる。



10-4. プログラミングの 楽しさと達成感

ソースコード

- ソースコードは、プログラミング言語で書かれたプログラムのもの
- 人間も読み書き，編集できる
- ソースコードにより，プログラムの動作を理解し，必要に応じて改変できる



```
function() {
  //is the element hidden?
  if (!t.is(':visible')) {
    //it became hidden
    t.appeared = false;
    return;
  }

  //is the element inside the visible window?
  var a = w.scrollLeft();
  var b = w.scrollTop();
  var o = t.offset();
  var x = o.left;
  var y = o.top;

  var ax = settings.accX;
  var ay = settings.accY;
  var th = t.height();
  var wh = w.height();
  var tw = t.width();
  var ww = w.width();

  if (y + th + ay >= b &&
      y <= b + wh + ay &&
      x + tw + ax >= a &&
      x <= a + ww + ax) {

    //trigger the custom event
    if (!t.appeared) t.trigger('appear', settings.data);

  } else {

    //it scrolled out of view
    t.appeared = false;
  }
};

//create a modified fn with some additional logic
var modifiedFn = function() {

  //mark the element as visible
  t.appeared = true;

  //is this supposed to happen only once?
  if (settings.one) {

    //remove the check
    w.unbind('scroll', check);
    var i = $.inArray(check, $.fn.appear.checks);
    if (i >= 0) $.fn.appear.checks.splice(i, 1);

  }

  //trigger the original fn
  fn.apply(this, arguments);
};
```

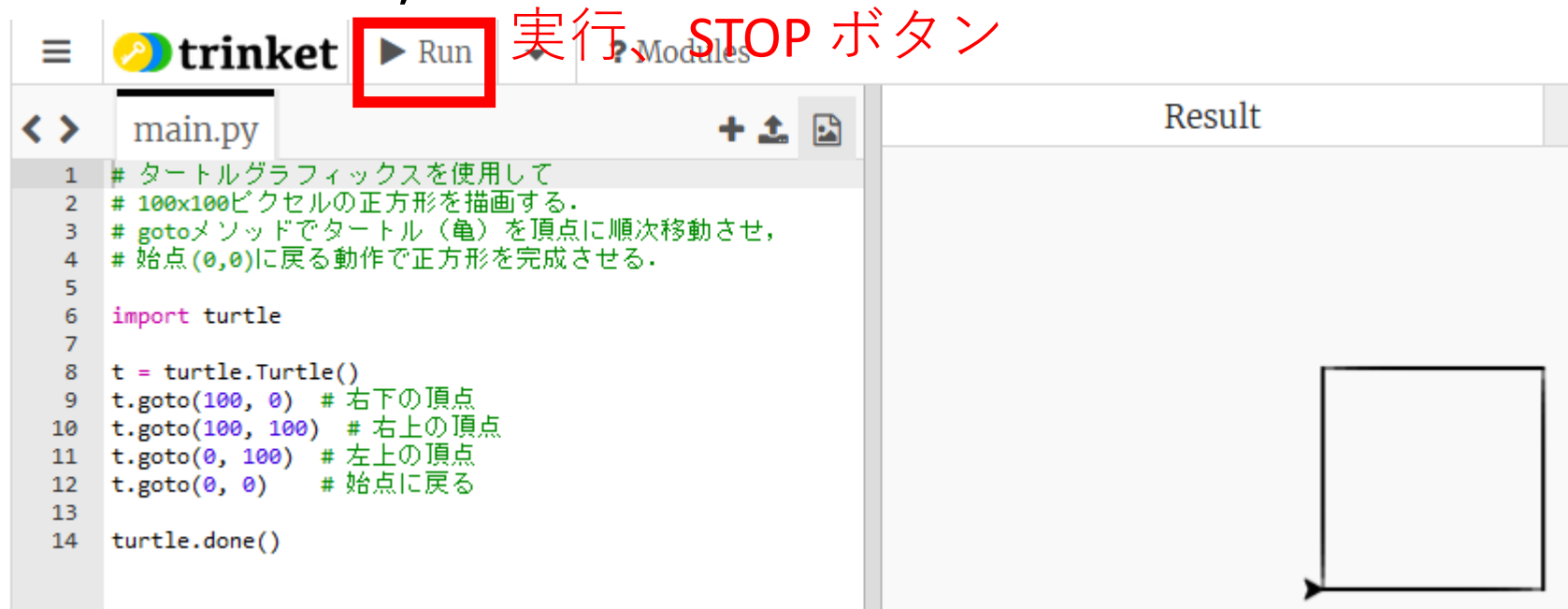
演習 図形のバリエーション



① 別のプログラムを試す. trinket の次のページを開く

<https://trinket.io/python/035810ce8d49>

② 実行結果が, 次のように表示されることを確認



実行、STOP ボタン

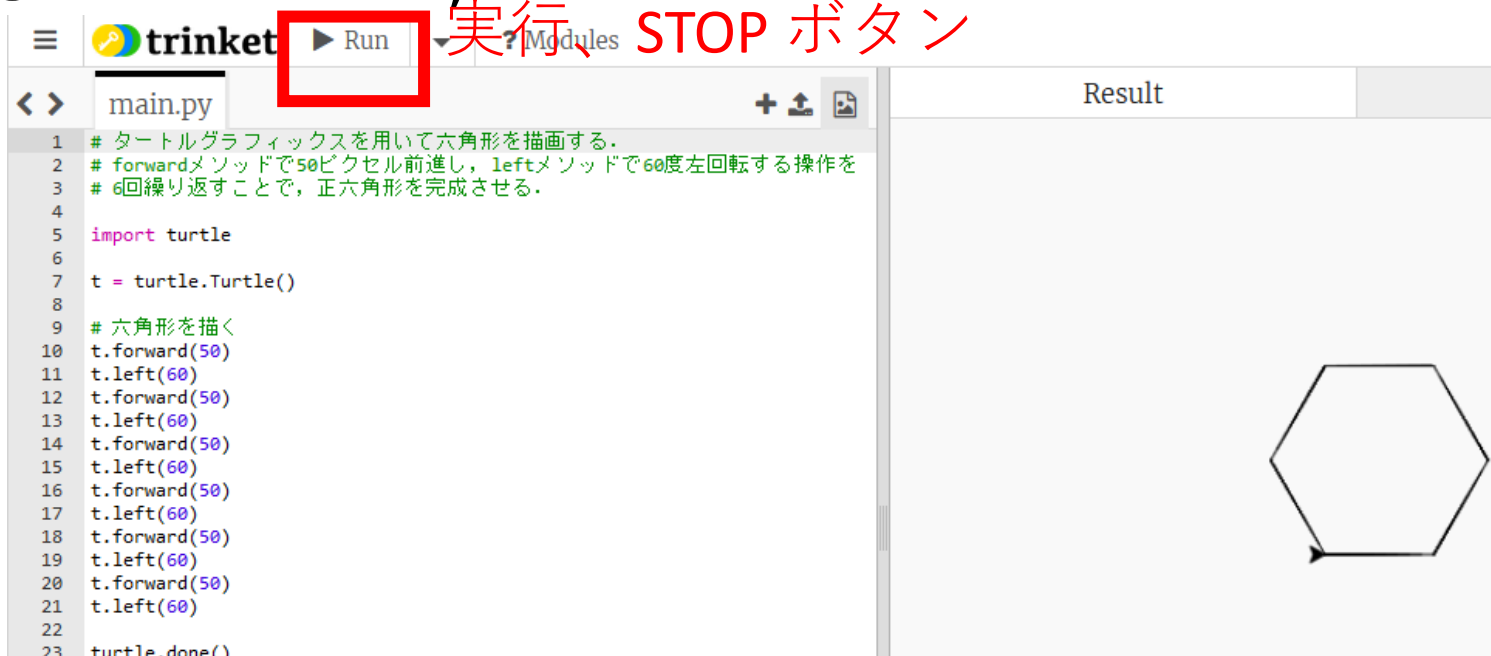
```
1 # タートルグラフィックスを使用して
2 # 100x100ピクセルの正方形を描画する.
3 # gotoメソッドでタートル(亀)を頂点に順次移動させ,
4 # 始点(0,0)に戻る動作で正方形を完成させる.
5
6 import turtle
7
8 t = turtle.Turtle()
9 t.goto(100, 0) # 右下の頂点
10 t.goto(100, 100) # 右上の頂点
11 t.goto(0, 100) # 左上の頂点
12 t.goto(0, 0) # 始点に戻る
13
14 turtle.done()
```

- 実行が開始しないときは、「**実行ボタン**」で**実行**
- ソースコードを**書き替えて再度実行**することも可能

③ 別のプログラムを試す. trinket の次のページを開く

<https://trinket.io/python/ddb861147133>

④ 実行結果が、次のように表示されることを確認



```
1 # タートルグラフィックスを用いて六角形を描画する。
2 # forwardメソッドで50ピクセル前進し、leftメソッドで60度左回転する操作を
3 # 6回繰り返すことで、正六角形を完成させる。
4
5 import turtle
6
7 t = turtle.Turtle()
8
9 # 六角形を描く
10 t.forward(50)
11 t.left(60)
12 t.forward(50)
13 t.left(60)
14 t.forward(50)
15 t.left(60)
16 t.forward(50)
17 t.left(60)
18 t.forward(50)
19 t.left(60)
20 t.forward(50)
21 t.left(60)
22
23 turtle.done()
```

- 実行が開始しないときは、「**実行ボタン**」で**実行**
- ソースコードを書き替えて再度実行することも可能

演習 プログラミングの楽しさと 達成感



各自の自発的な演習、自己研鑽の時間

① Pythonでグラフィックスを描く

資料のプログラムを動かして理解を深める

② Pythonの基本を押さえる

オブジェクト、メソッド、引数

③ 発想力、創造力

turtleモジュールを使用して、あなた自身がデザインした図形を描く。

④ 自主性、自己研鑽力、自分なりに工夫したことを振り返る

説明されなかった機能（他の図形の書き方）などを自主的に調べ、理解し、自分で試してみる。そして、自分なりに工夫したことを振り返り、省察することで、さらに実力アップ。

演習 プログラミングの楽しさと 達成感



各自の自発的な演習、自己研鑽の時間

① Pythonでグラフィックスを描く

資料のプログラムを動かして理解を深める

② Pythonの基本を押さえる

オブジェクト、メソッド、引数

③ 発想力、創造力

turtleライブラリを使用して、あなた自身がデザインした図形を描く。

④ 自主性、自己研鑽力、自分なりに工夫したことを振り返る

説明されなかった機能（他の図形の書き方）などを自主的に調べ、理解し、自分で試してみる。そして、自分なりに工夫したことを振り返り、省察することで、さらに実力アップ。

① 複雑な図形

<https://trinket.io/python/5366def2f4>

② 色, 円

<https://trinket.io/python/f8cd554693>

① <https://trinket.io/python/5366def2f4>



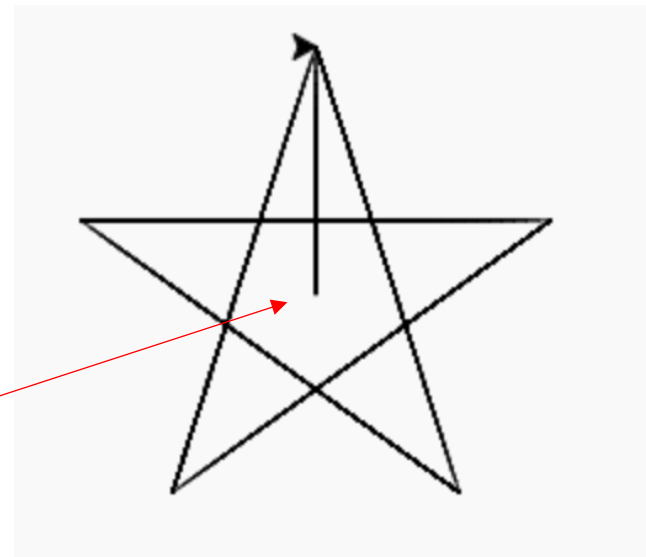
```
import turtle
t = turtle.Turtle()
t.goto(0, 100)
t.goto(58, -80)
t.goto(-95, 30)
t.goto(95, 30)
t.goto(-58, -80)
t.goto(0, 100)
```

ライブラリのインポート
オブジェクト生成。t へのセット。

移動

実行結果
**最初の位置は
(0, 0)**

(0, 0)



② <https://trinket.io/python/f8cd554693>



```
import turtle
```

```
t = turtle.Turtle()
```

```
colors = ["red", "green", "blue"]
```

```
for i in range(3):
```

```
    t.color(colors[i])
```

```
    t.circle(30)
```

```
    t.forward(50)
```

ライブラリのインポート

オブジェクト生成。t へのセット。

色は、赤、緑、青

色を変える

半径 30 の円

前に50進む

実行結果



① プログラミングの基礎

オブジェクト、メソッド、引数などの基本概念を理解。CodeCombatを用いてPythonの実践的スキルを習得

② 論理的思考力の向上とプログラミングの面白さ体験

タートルグラフィックスで論理的思考力を養成。アイデアを形にする楽しさと達成感

③ プログラミングの重要性の理解

シミュレーション、データ処理、AI開発などさまざまな活用

④ 自主的な学習姿勢の育成と視野拡大

自主的な学習姿勢を身につける。プログラミングの基礎を通して視野を広げる