cs-10. プログラミング入門: プログラミング思考の基礎

（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

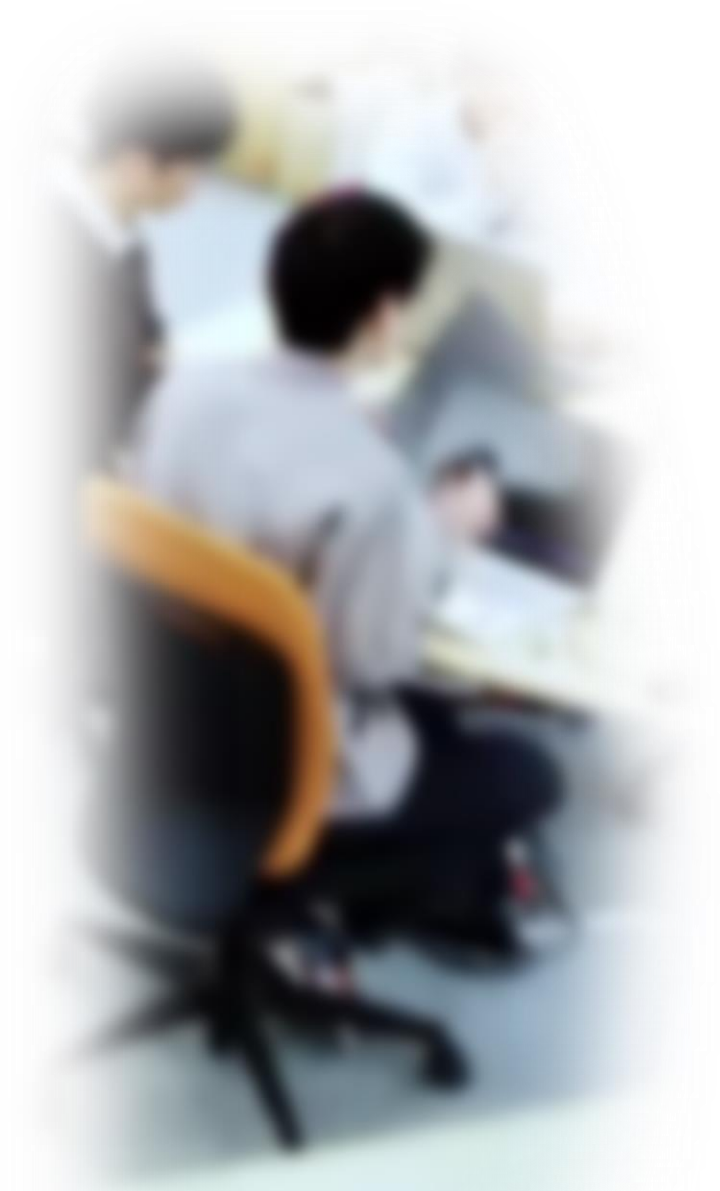
CC BY NC SA

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（汎用プログラミング言語）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（汎用プログラミング言語）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

- **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，予測や分析を行いる
  - 大量データ処理：データの収集，加工，分析
  - AI（人工知能）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（汎用プログラミング言語）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（汎用プログラミング言語）の基本を楽しく学ぶことができる

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（汎用プログラミング言語）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（汎用プログラミング言語）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ある）

← **挑戦は，興味のある人の自習とする（各自の判断）**

この授業では，**会員登録をしない，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()

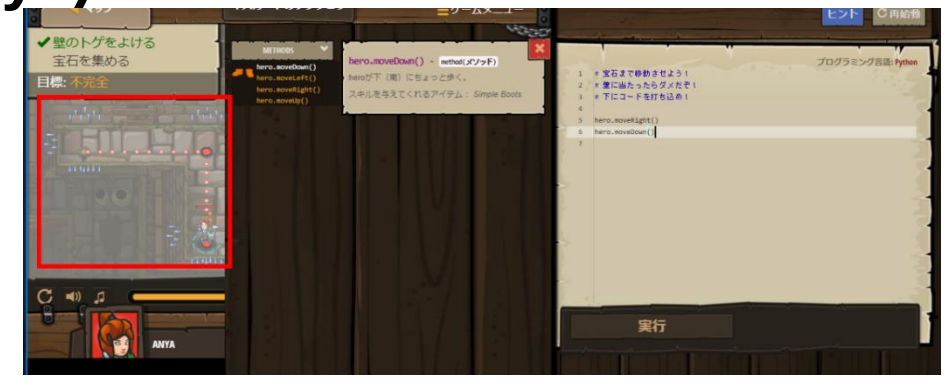
  heroは**オブジェクト**

  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（汎用プログラミ

オブジェクトとメソッド

オブジェクトが動く

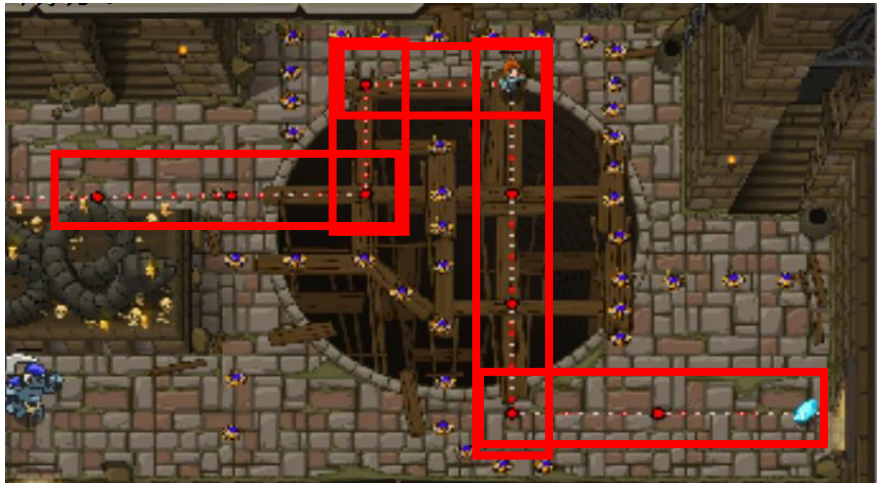
実行画面

```
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

①Web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「私は学生である」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

クラスコードを持っているか

- 「**今すぐプレイ**」をクリック

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

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

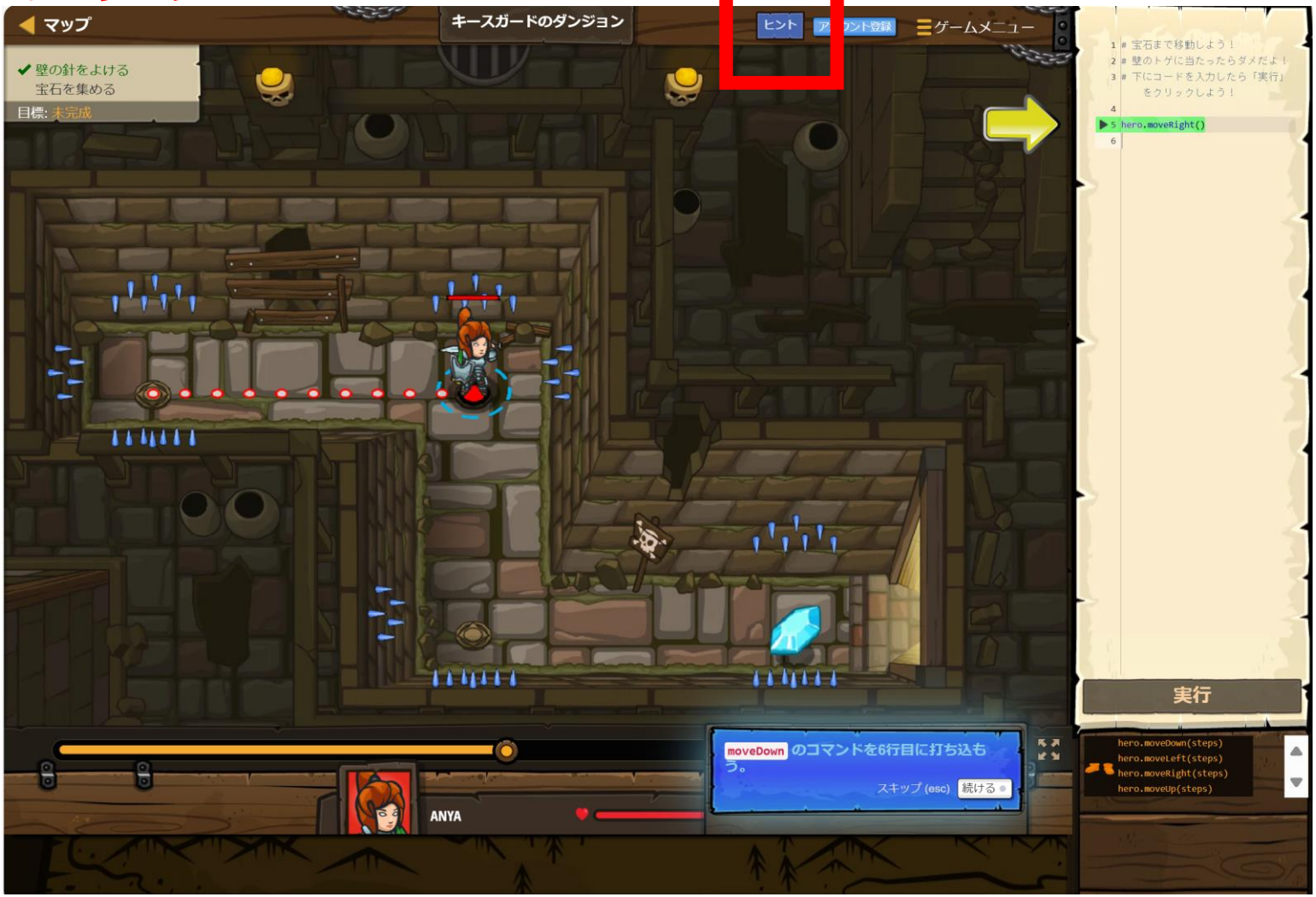
⑥「Python（汎用プログラミング言語）（デフォルト）」を選び，「次へ」をクリック

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

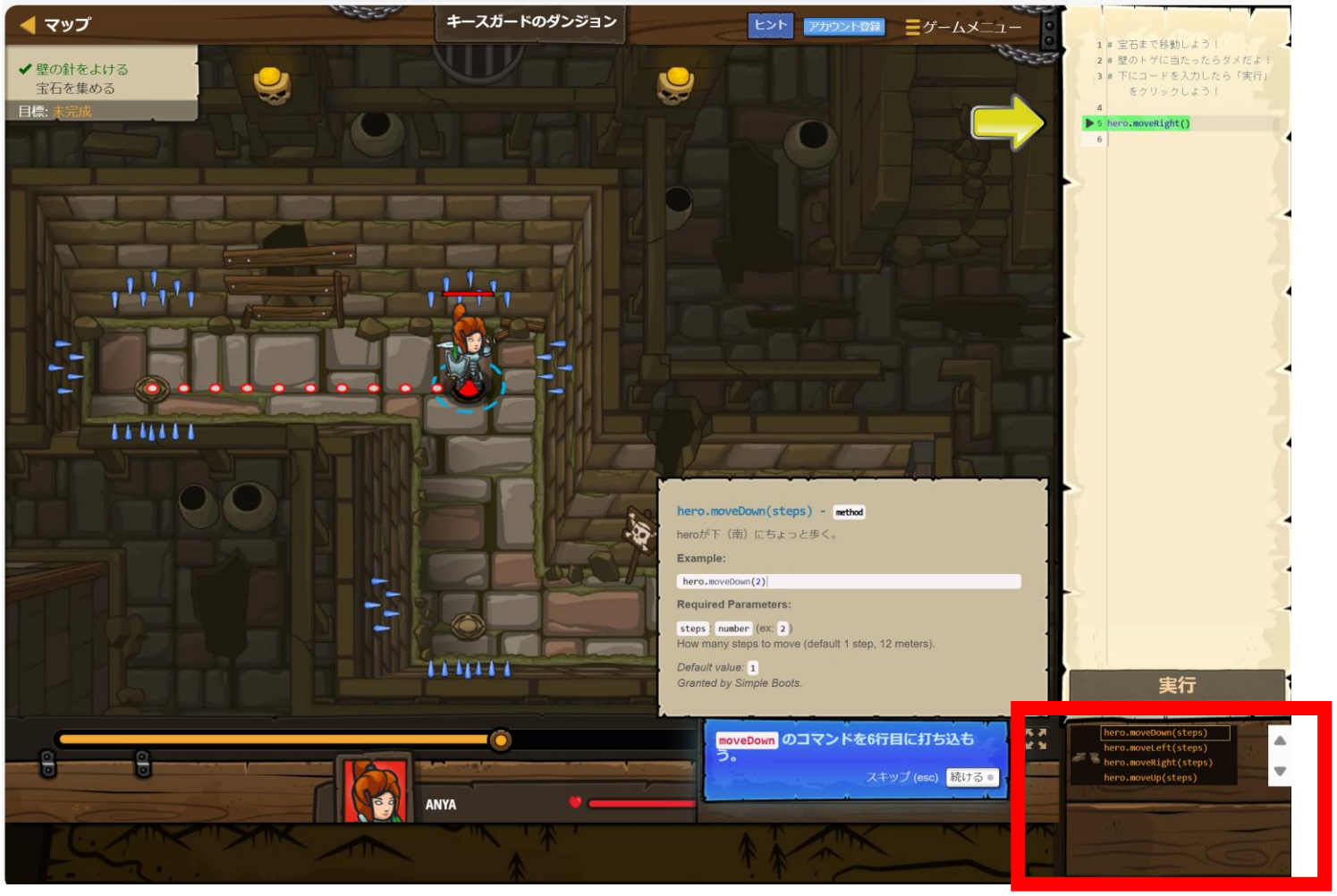
⑨「レベルスタート」をクリック

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る．
翻訳が完全でないためである． 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



但し，英語で表示される場合がありる．
翻訳が完全でないためである． 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

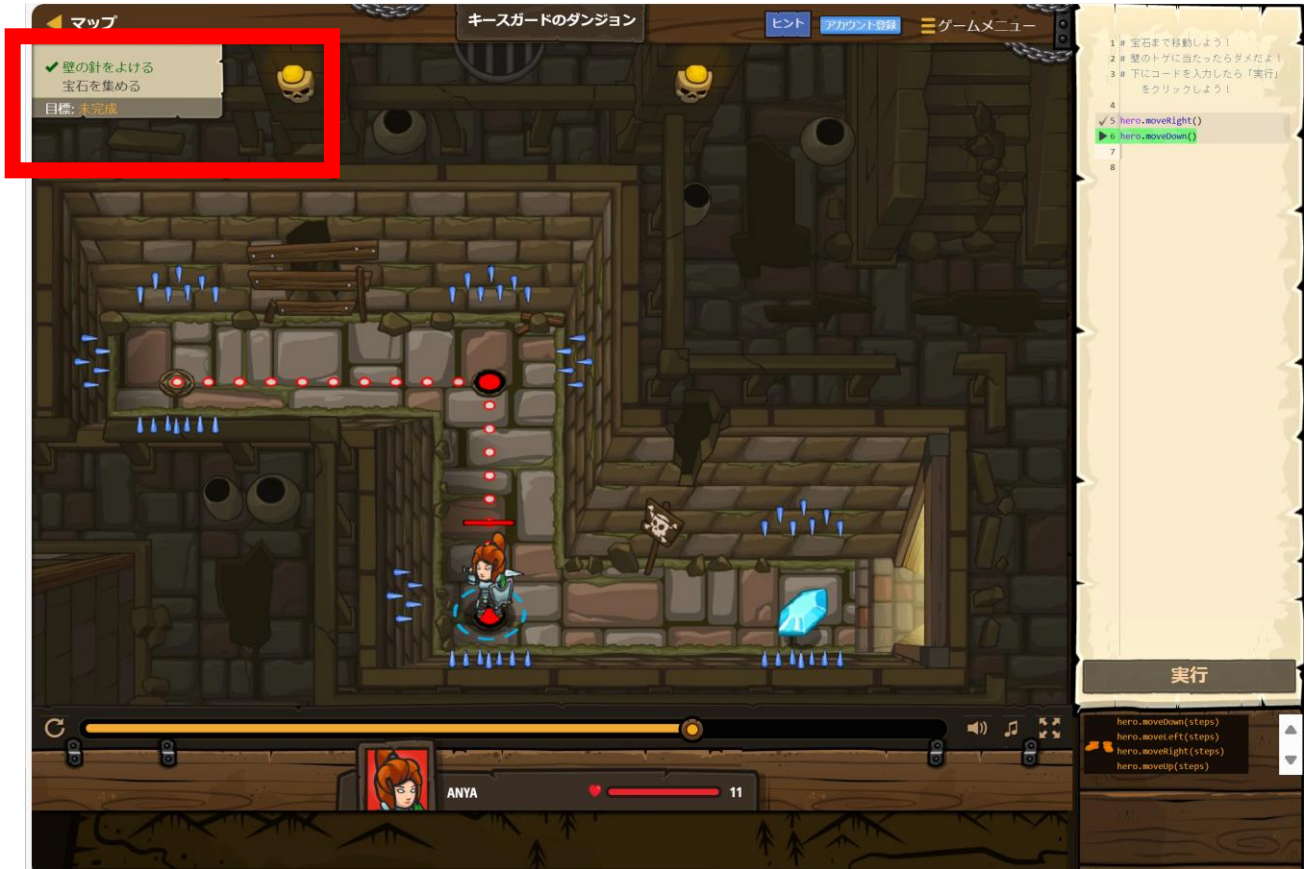
⑪ 「実行」で，キャラクタが動くので確認する

hero.moveRight()で**右に動き**，hero.moveDown()で**下に動く**

迷ったら，「ミッション（目標）」や「ヒント」を確認する

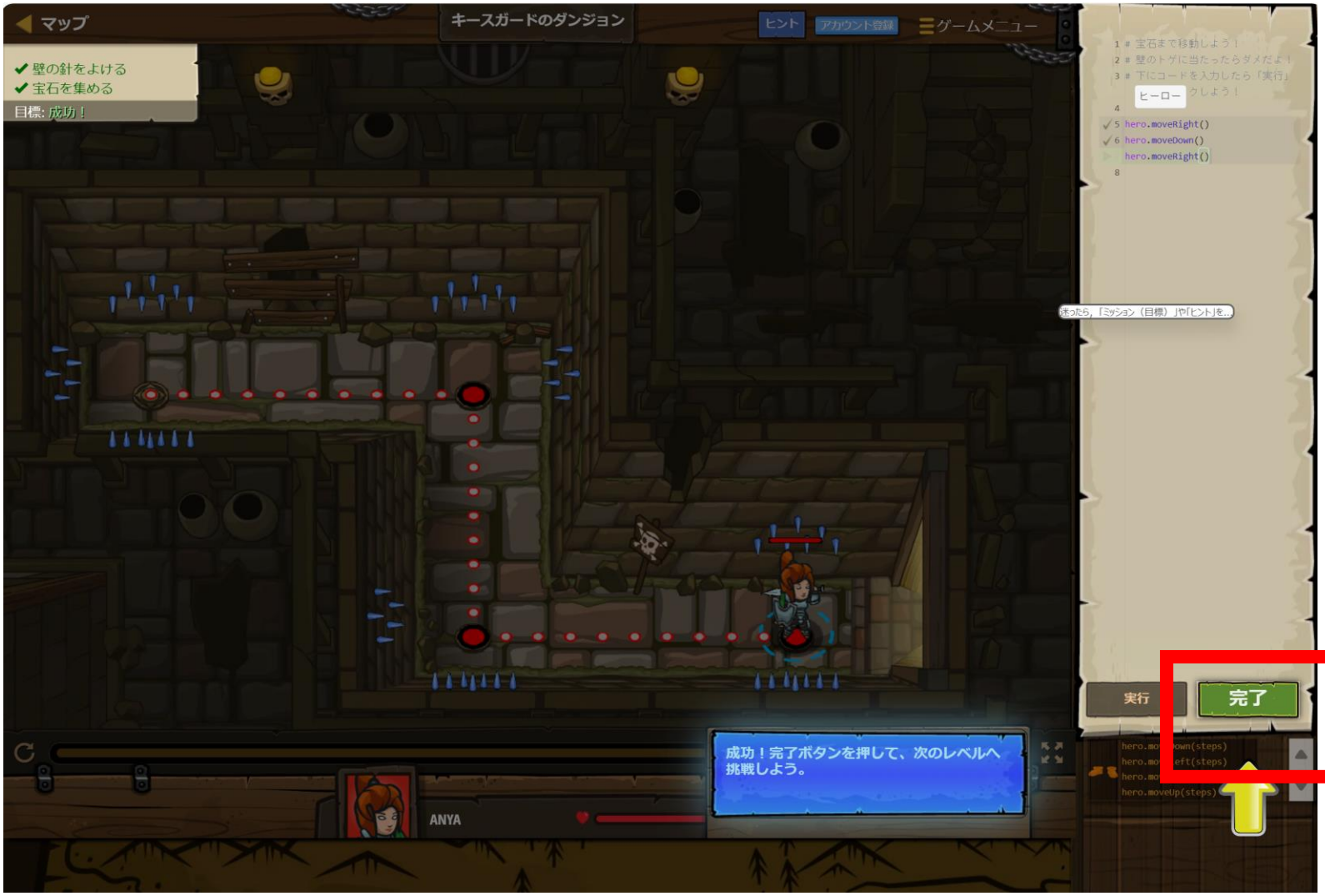
- プログラミングの練習だけでなく，**ゲーム要素（パズル）**もある． 楽しんで解く．
- ヒントや説明が，英語で表示される場合がある

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



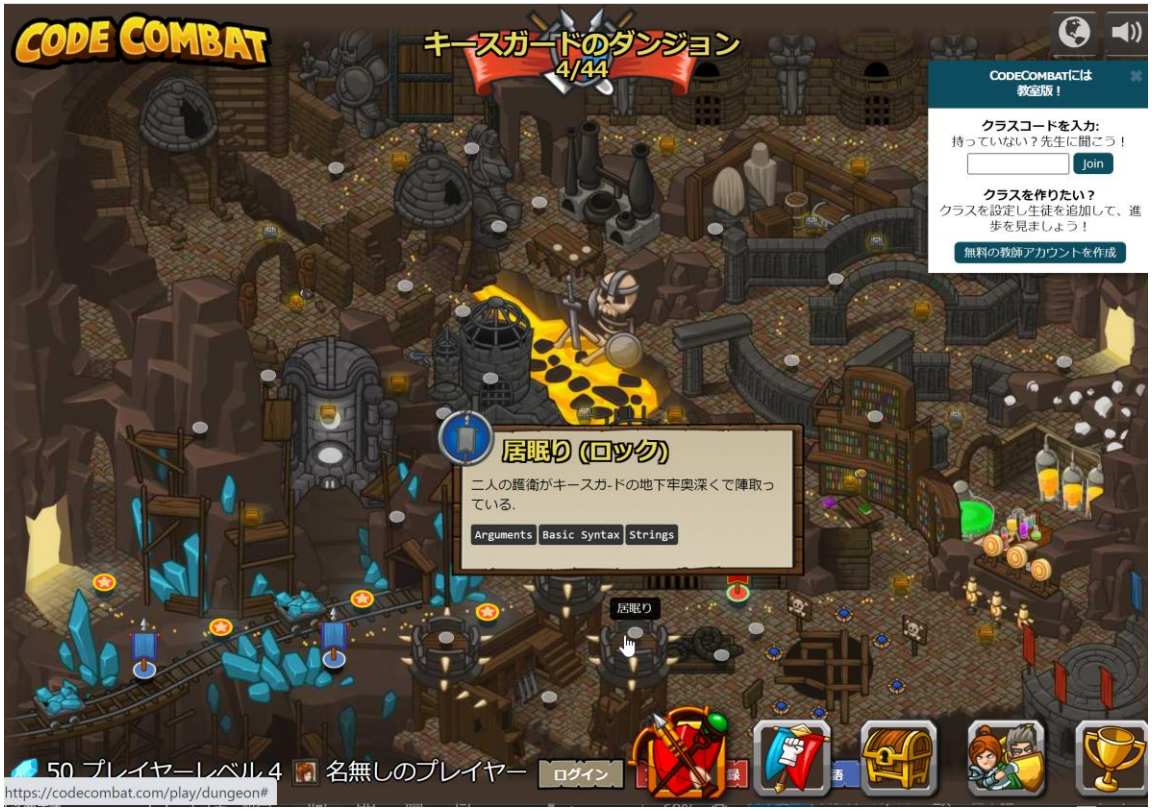
⑬ 完了の確認.　「**続ける**」をクリック

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

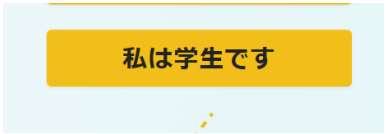
無料で，クラスコードを使わずに，Python（汎用プログラミング言語）の5つのレベルを学ぶことができる

（続きは有料となる． 詳しくは次ページ）

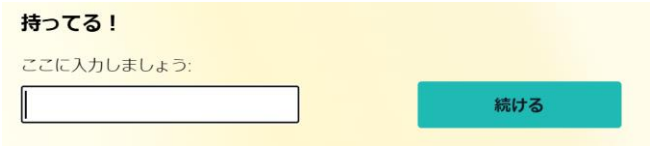
次のことは，各自の判断とする（このことは，授業の成績に関係しない．）

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（無料で学んでいける）

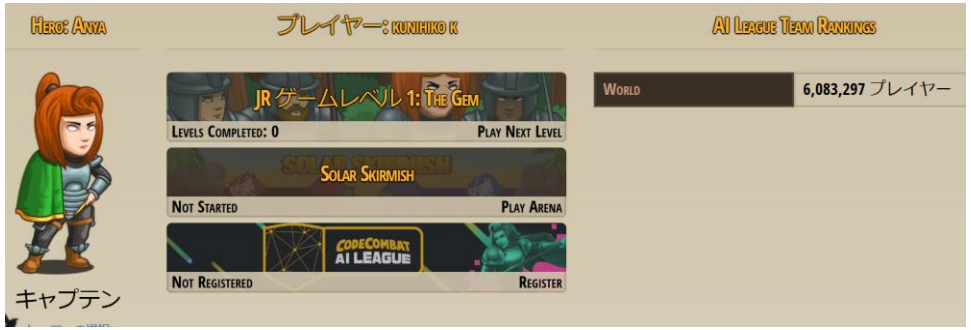


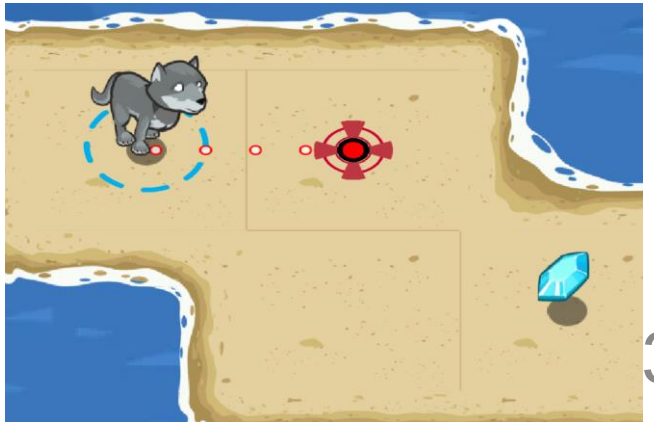
私は学生である



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

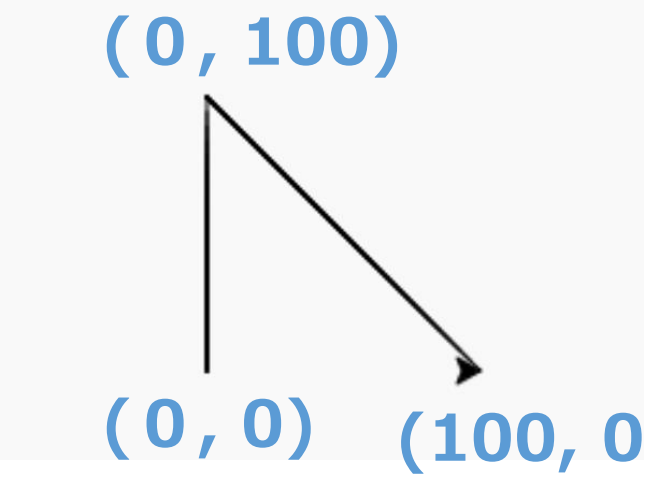
タートルグラフィックス

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**
- **backword**(＜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)
```

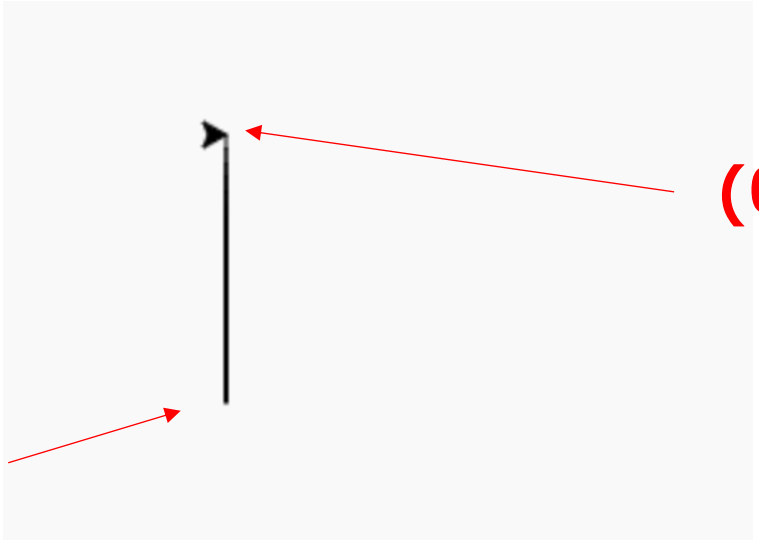
インポート

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)



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

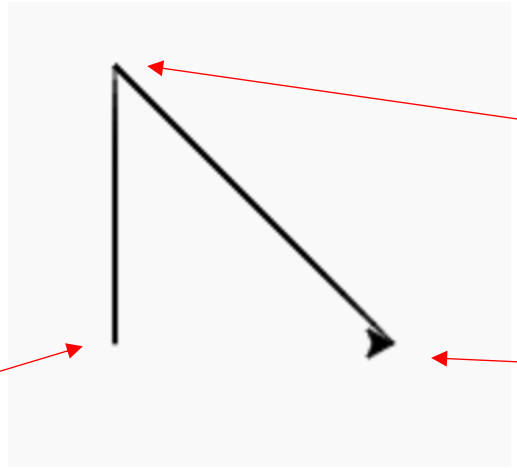
オブジェクト生成．tへのセット．

(0, 100)への移動

(100, 0)への移動

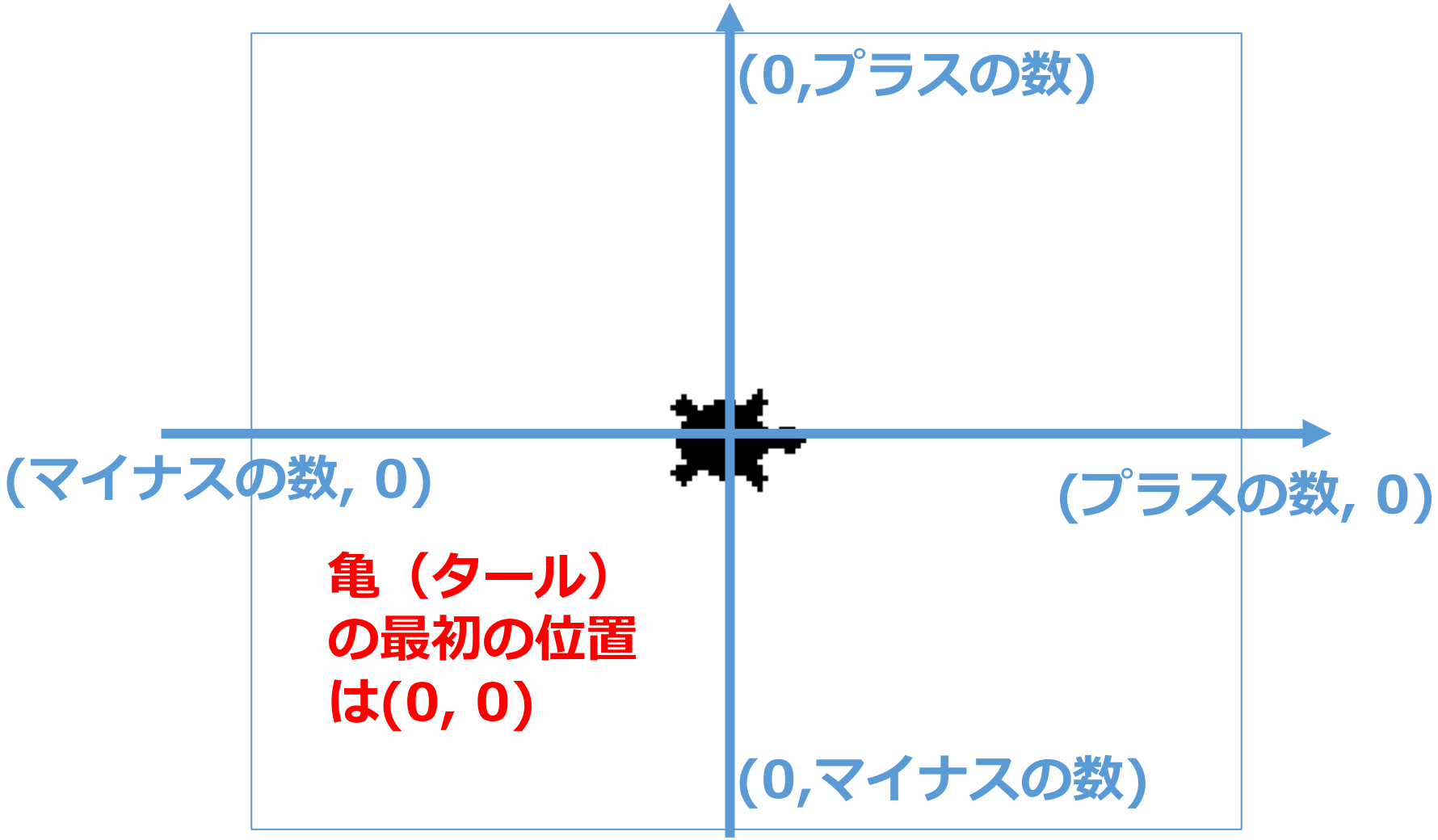
実行結果

最初の位置は
(0, 0)



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

- **Trinket（Python などを実行できるオンライン｜学習環境）**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

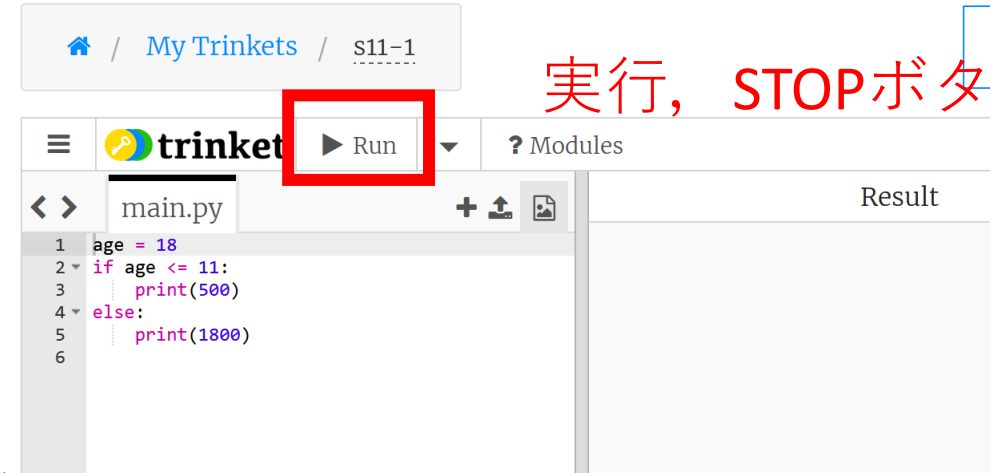
math,matplotlib.pyplot,numpy, operator, processing,pygal, random, re, string, time, turtle,urllib.request

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



- 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

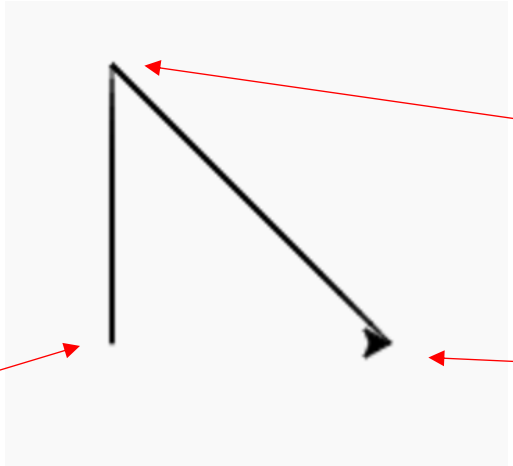
オブジェクト生成．tへのセット．

(0, 100)への移動

(100, 0)への移動

実行結果

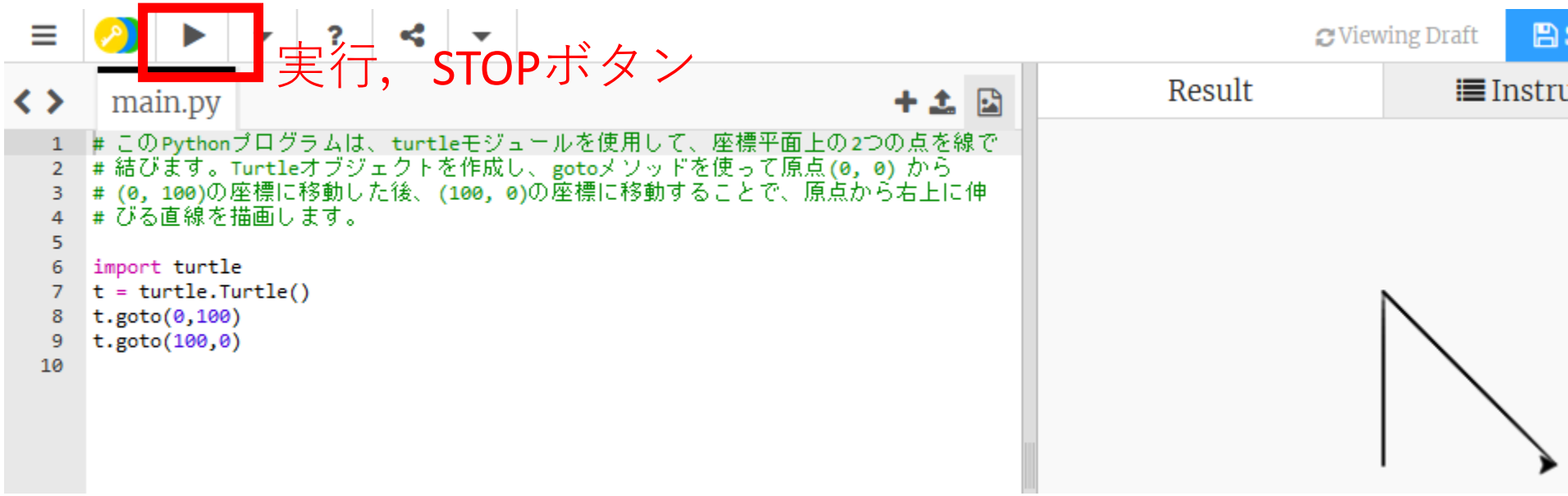
最初の位置は
(0, 0)



①trinket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

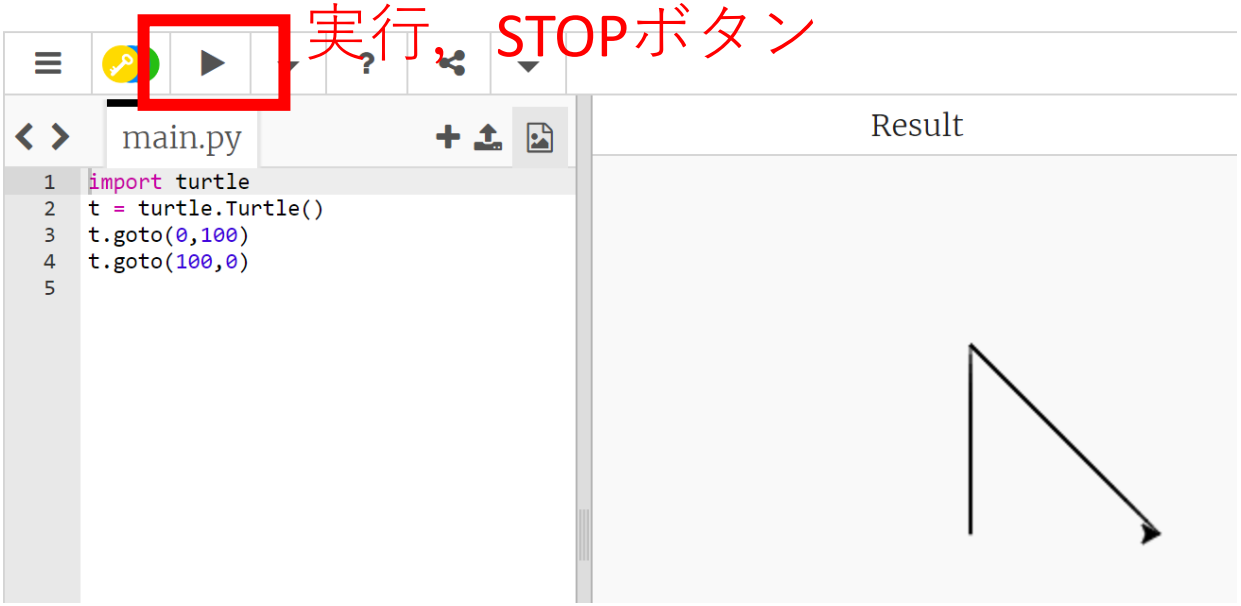


- 実行が開始しないときは，「**実行ボタン**」で**実行**
- ソースコードを**書き替えて再度実行**することも可能

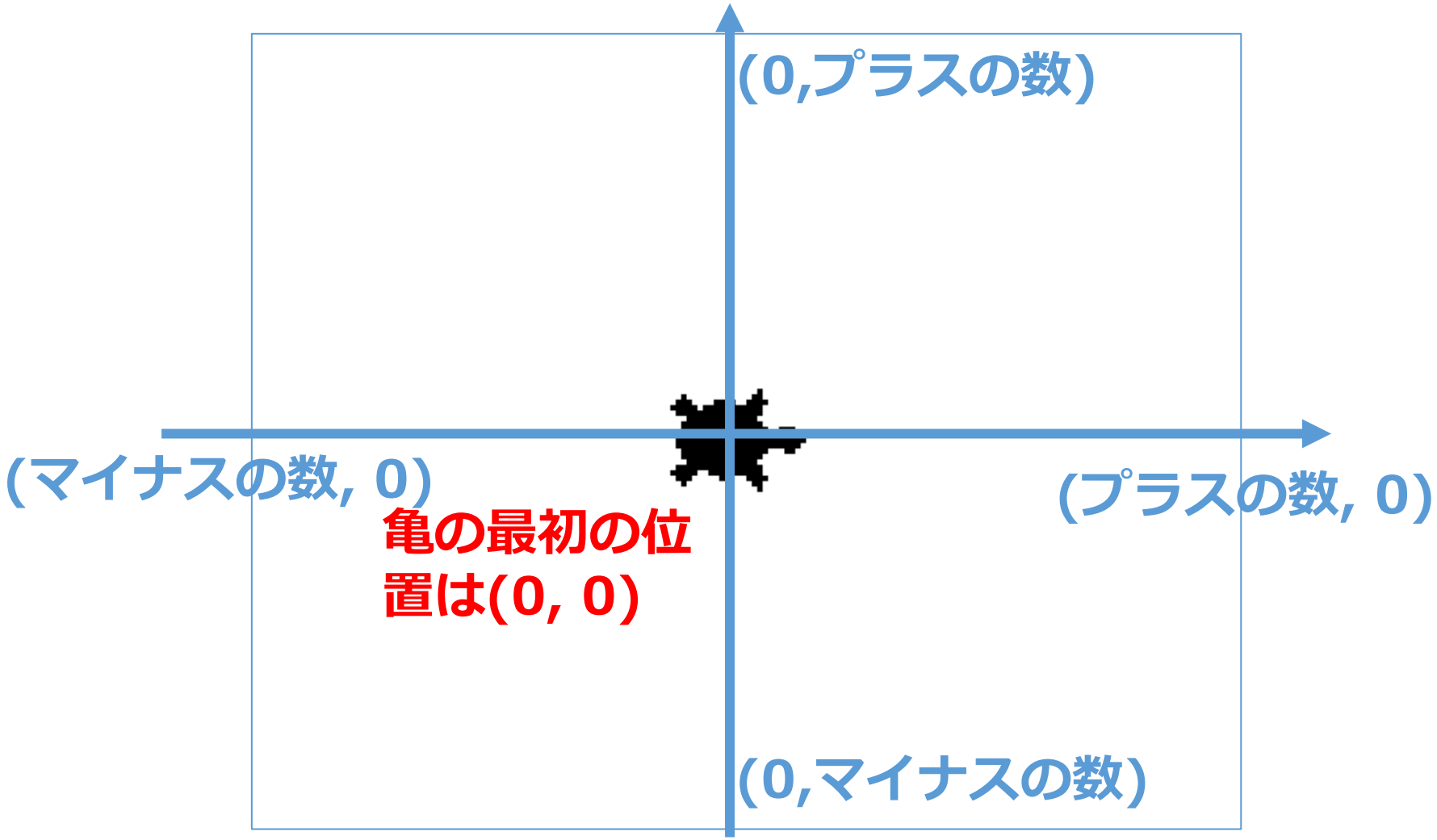
③ 次のように**書き換えて，再実行し，結果が変わる**ことを確認

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

書き換え



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（汎用プログラミング言語）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

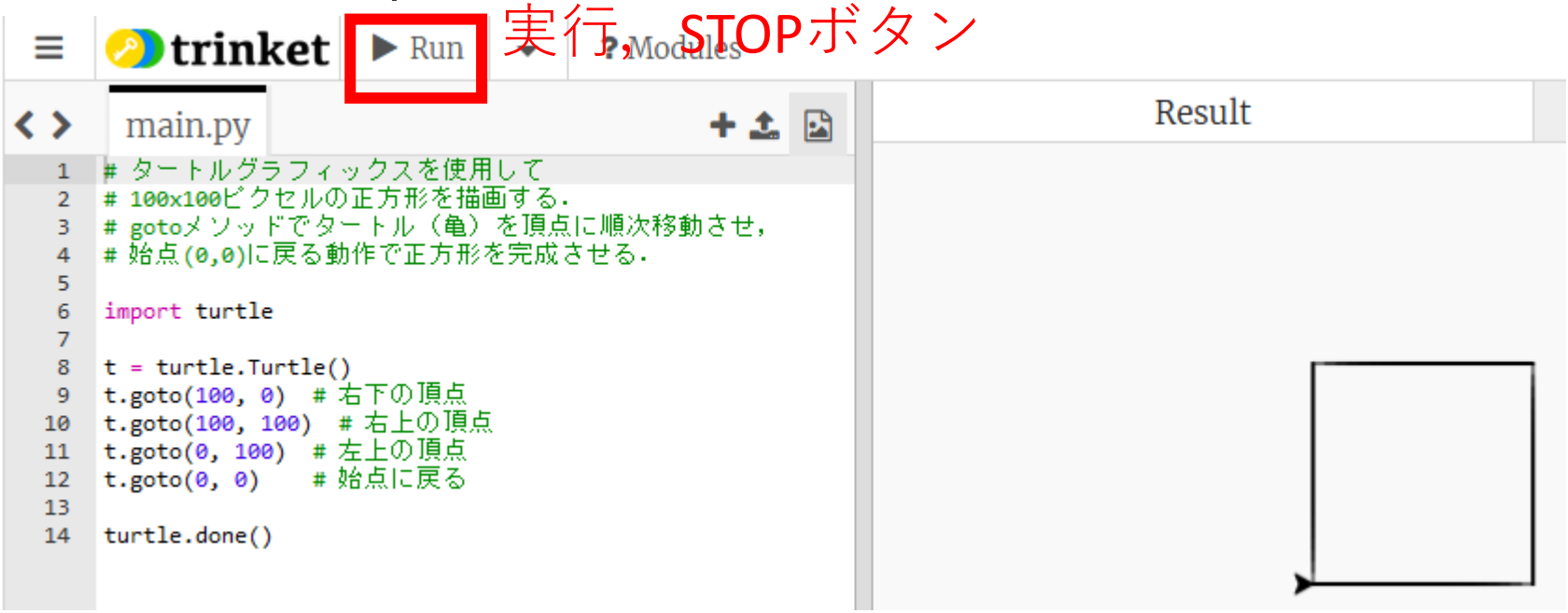
- **ソースコード**は，**プログラミング言語**で書かれた**プログラム**のもの
- 人間も**読み書き**，**編集**できる
- **ソースコード**により，**プログラムの動作を理解**し，必要に応じて**改変**できる

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

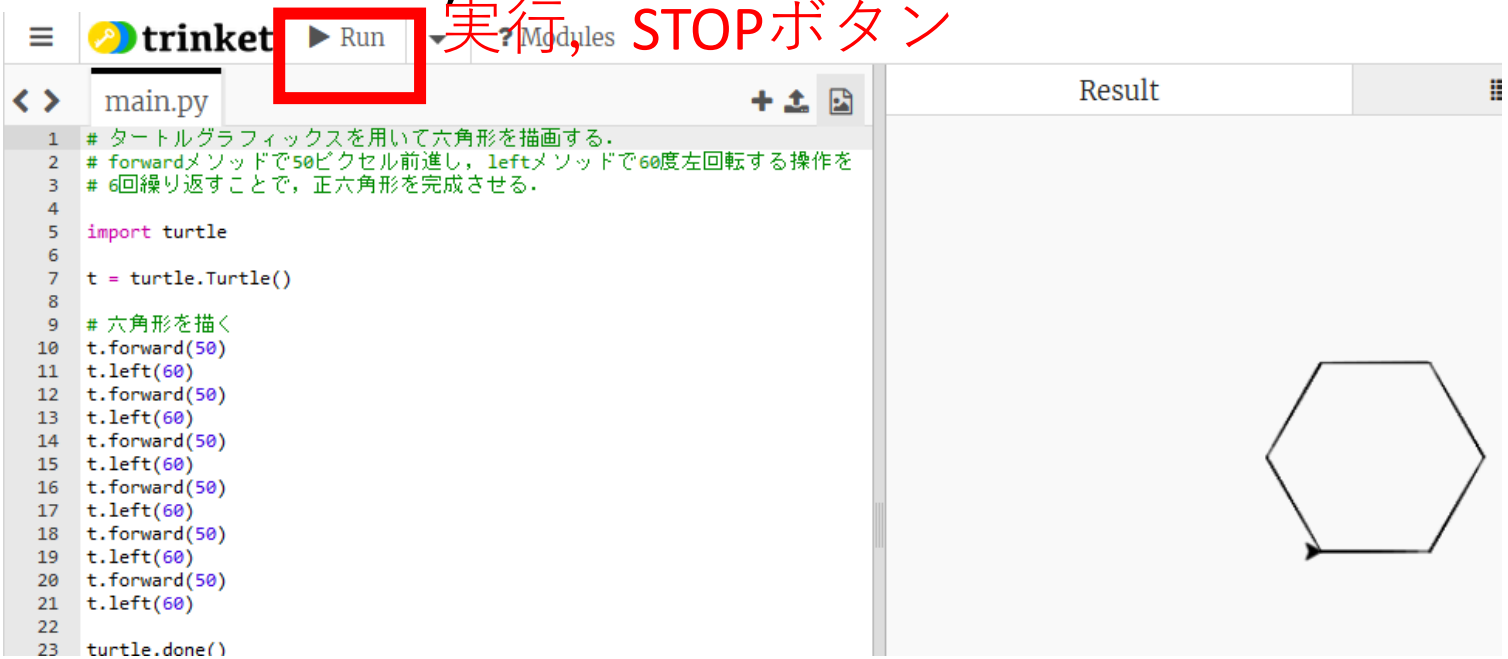


- 実行が開始しないときは，「**実行ボタン**」で**実行**
- ソースコードを**書き替えて再度実行**することも可能

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



- 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

①Pythonでグラフィックスを描く

資料のプログラムを動かし理解を深める

②Python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

①Python でグラフィックスを描く

資料のプログラムを動かし理解を深める

②Python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

①https://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)
```

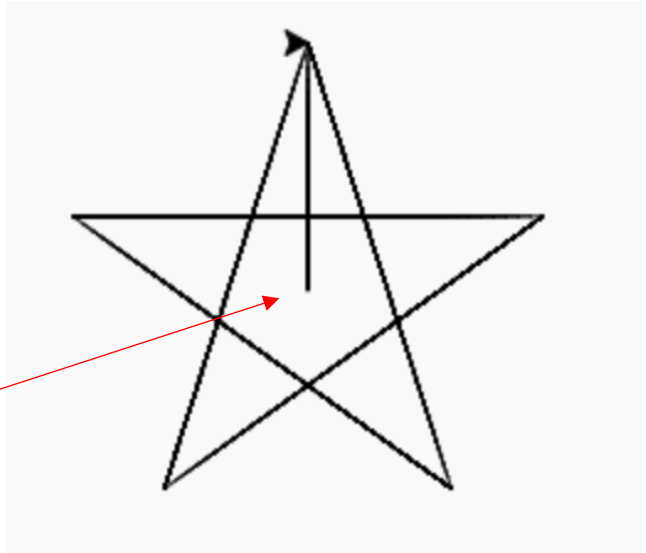
ライブラリのインポート

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)



②https://trinket.io/python/f8cd554693

```
import turtle

t=turtle.Turtle()

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

foriin 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．

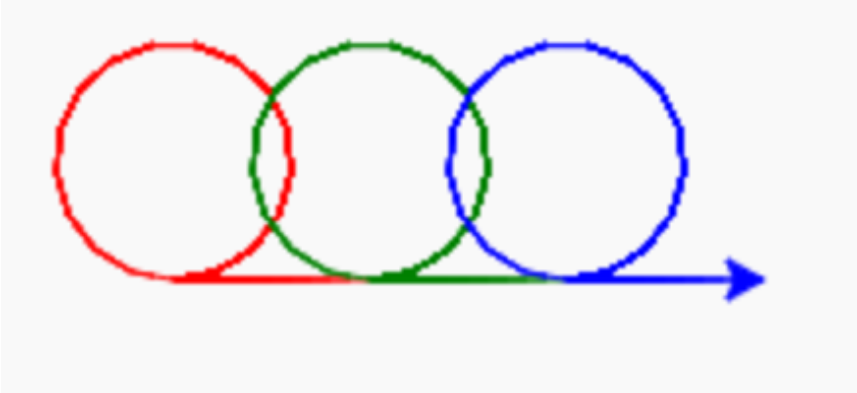
色は，赤，緑，青

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（汎用プログラミング言語）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

クトルグラフィックスで論理的思考力を養成．

再アップロード完了しました

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（人工知能）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