

## 2. プログラミング入門：タートルグラフィックスとコードコンバットによる演習

Pythonプログラミング講座：基礎から応用まで  
(全15回)

URL: <https://www.kkaneko.jp/pro/pf/index.html>

金子邦彦



# 2-1. 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)**

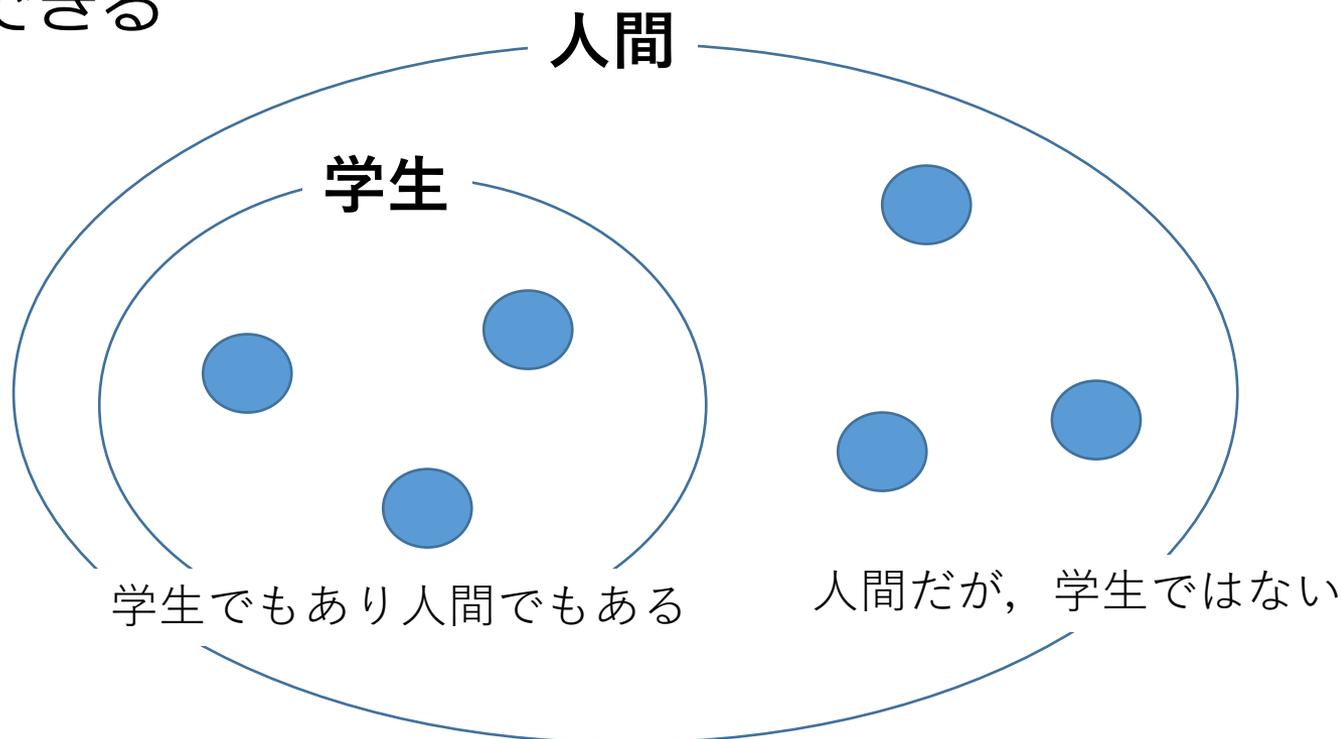


引数

# クラスとオブジェクト



- **クラス**は、同じ種類のオブジェクトの集まりと考えることができる



- **オブジェクト生成**では、**クラス名**を指定

例 `t = turtle.Turtle()`

**クラス名**

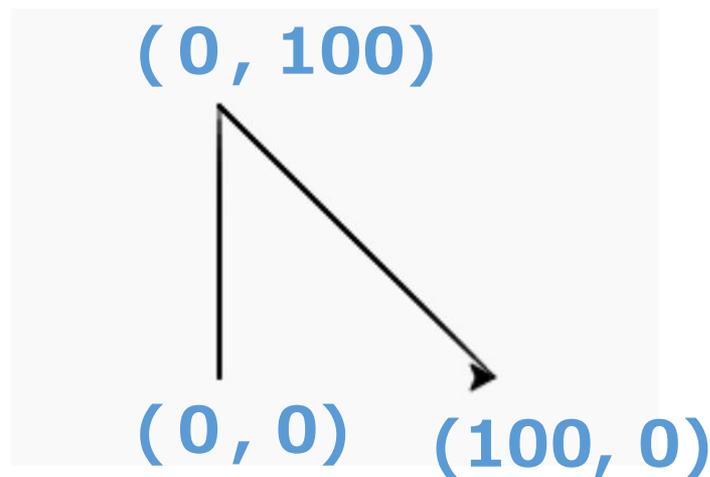
# プログラミング学習のためのタートルグラフィックス



- **カーソル**（画面上を移動する亀の形をしたポインタ）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。
- プログラムの**動作を視覚的に理解**できる
- **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の基本機能



```
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**指定座標への移動**

## 主なメソッド

- **goto** ( <横方向の値> , <縦方向の値> ) **指定座標への移動**
- **forward**( <移動量> ) **前進**
- **backward**( <移動量> ) **後退**
- **right**( <角度> ) **右回り回転**
- **left**( <角度> ) **左回り回転**

# オブジェクトとメソッドの実践例①



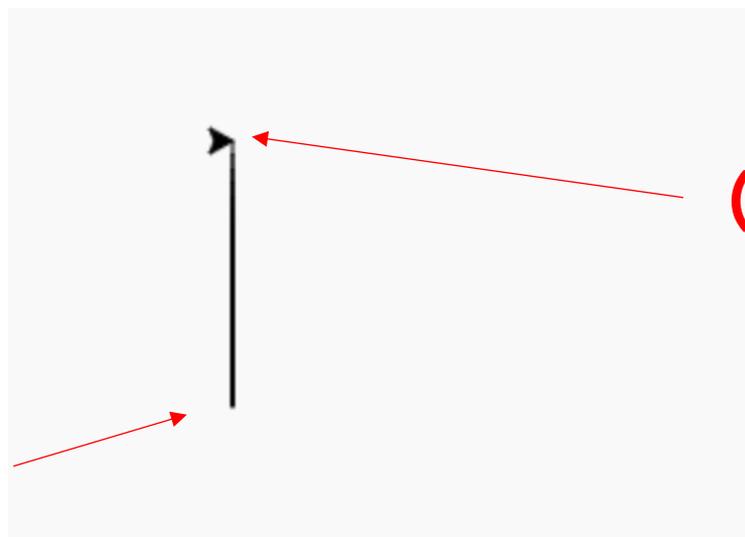
```
import turtle  
t = turtle.Turtle()  
t.goto(0,100)
```

タートルグラフィックスの機能をインポート  
オブジェクト生成. 変数 t にセット.

**座標 (0, 100) に移動**

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の実践例②



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

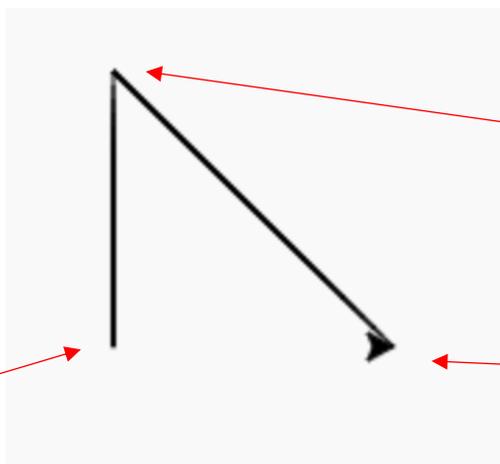
タートルグラフィックスの機能をインポート  
オブジェクト生成. 変数  $t$  にセット.

**座標  $(0, 100)$  に移動**

**座標  $(100, 0)$  に移動**

実行結果  
**最初の位置は  
 $(0, 0)$**

**$(0, 0)$**



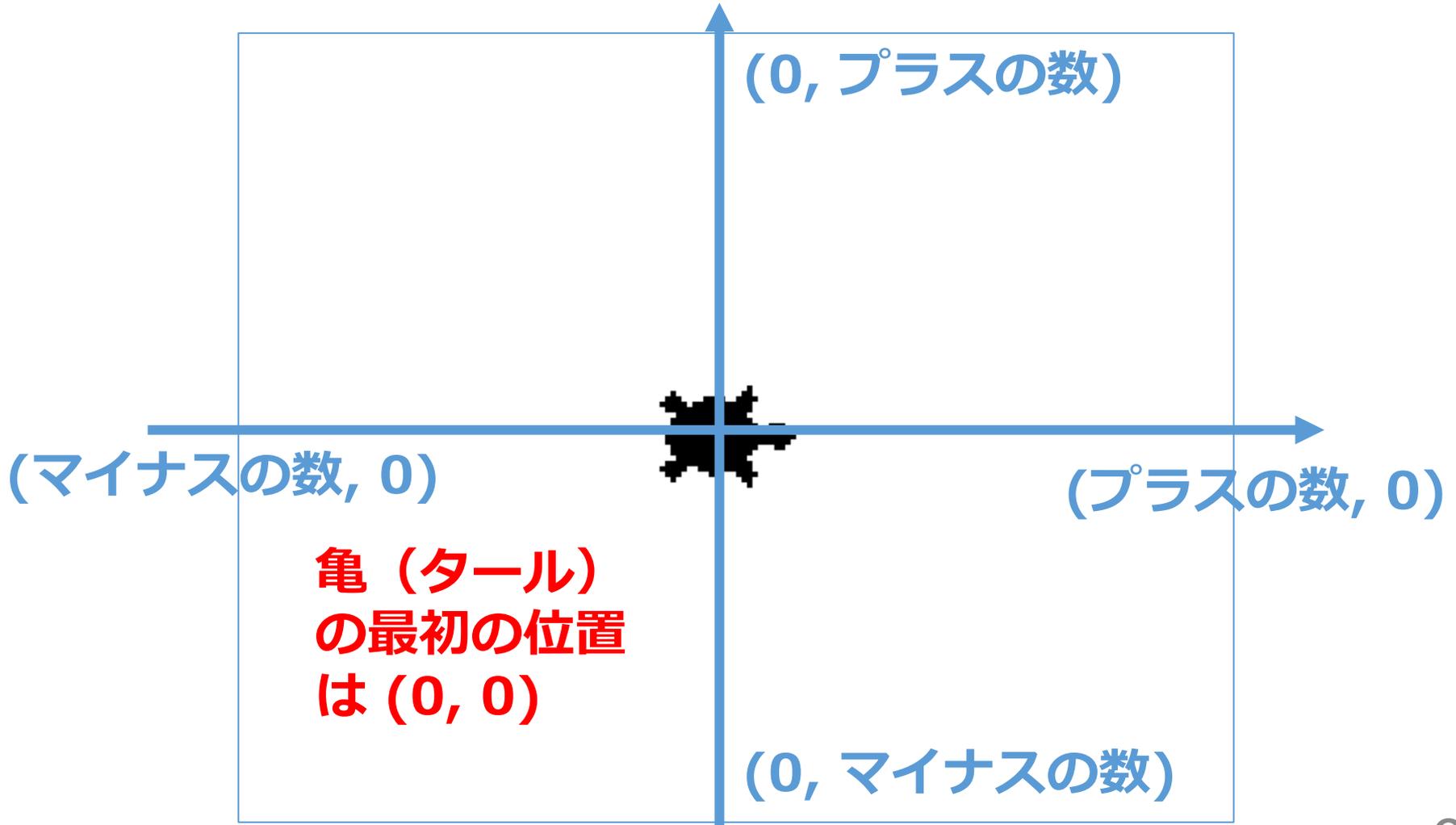
**$(0, 100)$**

**$(100, 0)$**

オブジェクトが動く

# 座標系と goto メソッドの引数

縦方向の値と、横方向の値を、メソッド goto の引数として指定



# Trinket の概要



- Trinket は**オンライン**の Python、HTML 等の**学習サイト**
- ブラウザで動作
- 有料の機能と無料の機能を提供
- **自作プログラムの公開と共有が可能**
- Python の**標準機能**に加え，外部ライブラリ matplotlib.pyplot, numpy, processing, pygal が利用可能

A screenshot of the Trinket web interface. At the top, there is a breadcrumb navigation: a home icon, followed by "/ My Trinkets / s11-1". Below this is a toolbar with a hamburger menu icon, the Trinket logo (a key icon), the text "trinket", a "Run" button with a play icon, a dropdown arrow, and a "? Modules" link. The main area is split into two panes. The left pane is a code editor for a file named "main.py", showing the following Python code:

```
1 age = 18
2 if age <= 11:
3     print(500)
4 else:
5     print(1800)
6
```

The right pane is labeled "Result" and is currently empty.

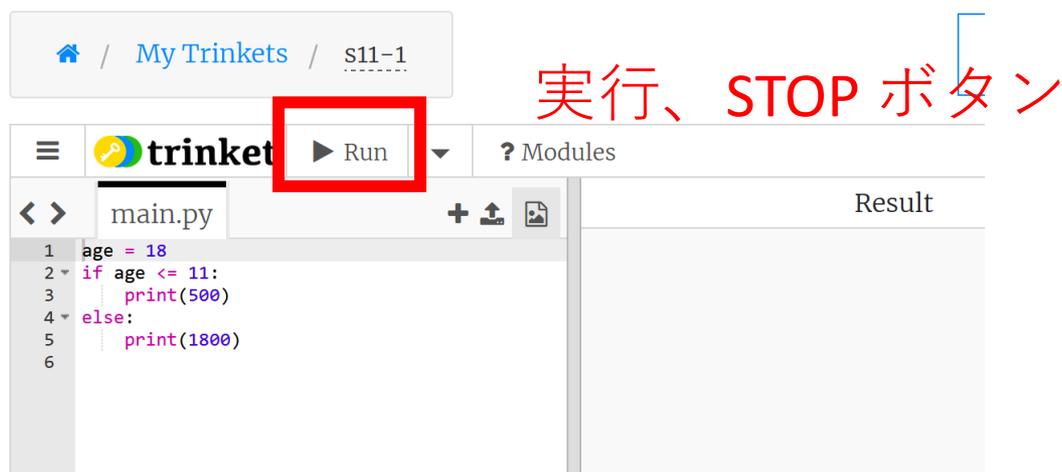


# Trinket の基本操作

- 公開プログラムごとに固有の URL が割り当てられる

例 <https://trinket.io/python/0fd59392c8>

- 「Run」ボタンによるプログラムの開始, 「STOP」ボタンによる終了



確認編集用の  
メイン画面

実行結果

- **メイン画面でのプログラム編集と再実行が可能**
- 左側で実行結果を確認

# 演習. 基本的なプログラミング



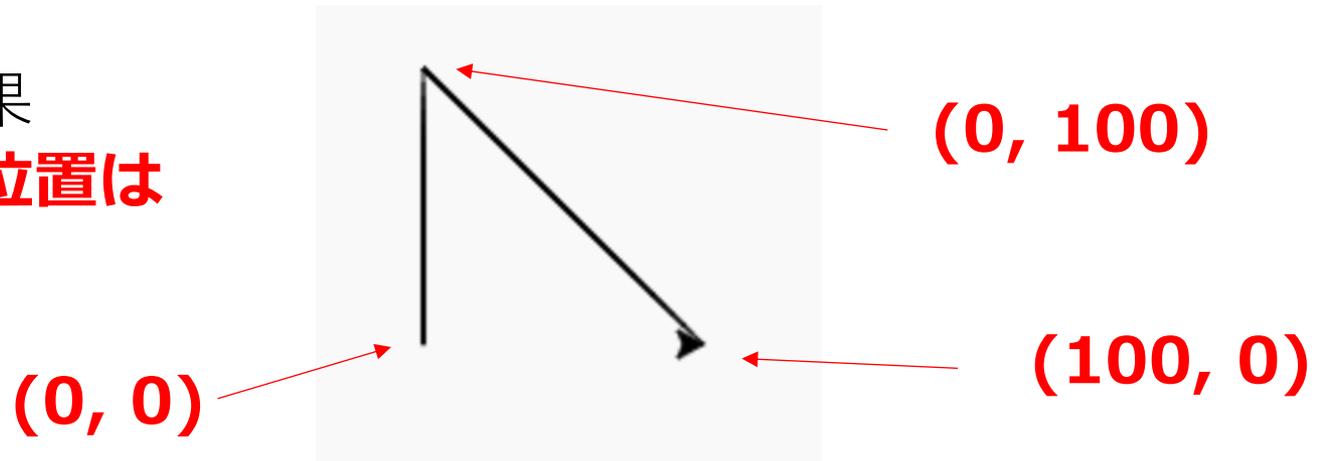
```
import turtle  
t = turtle.Turtle()  
t.goto(0,100)  
t.goto(100,0)
```

タートルグラフィックスの機能をインポート  
オブジェクト生成. 変数 t にセット.

**座標 (0, 100) に移動**

**座標 (100, 0) に移動**

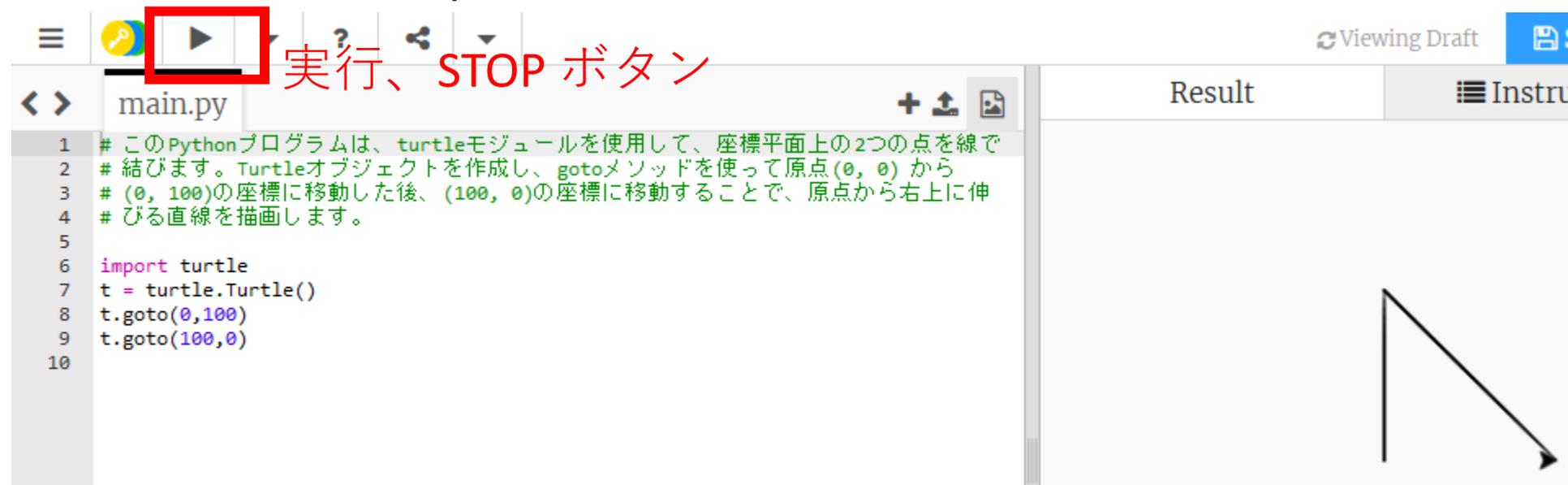
実行結果  
**最初の位置は  
(0, 0)**



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



- 実行が開始しないときは、「**実行ボタン**」で**実行**
- プログラムを書き替えて再度実行することも可能

### ③ プログラムの変更と再実行

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)
```

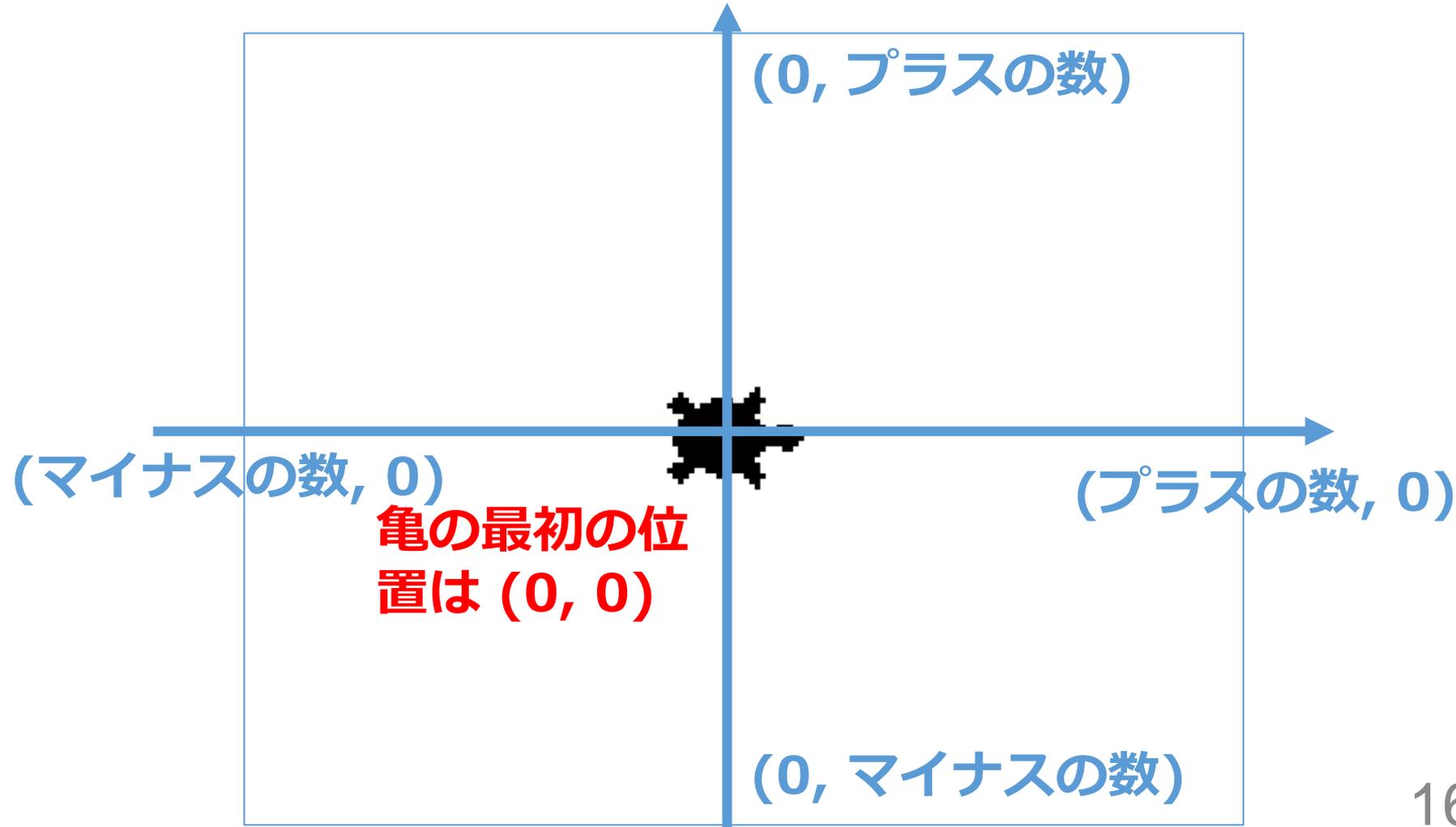
← 書き換え





#### ④ プログラムの改良と発展

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。



## まとめ

- **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`」
- **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。

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

# Pythonプログラミング学習：コードコンバット



- コードコンバット（CodeCombat）では、プログラミングの基礎をゲーム形式で学ぶ
- キャラクターを動かすためのプログラムを作成しながら、段階的にプログラミングスキルを向上させることができる



# 授業における学習範囲と利用形態



本授業では以下の範囲で学習を進める

- **無料, クラスコード不要, Python の基本 5 レベル** **授業範囲**
- **無料, クラスコード使用, より本格的な利用**  
クラスコードの使用には, メールアドレスの登録などが必要
- **有料会員登録 (500以上のレベル) は任意**  
各自の判断で行う

# Python基礎トピックス

- オブジェクトとメソッド

`hero.moveDown()`

`hero` は**オブジェクト**

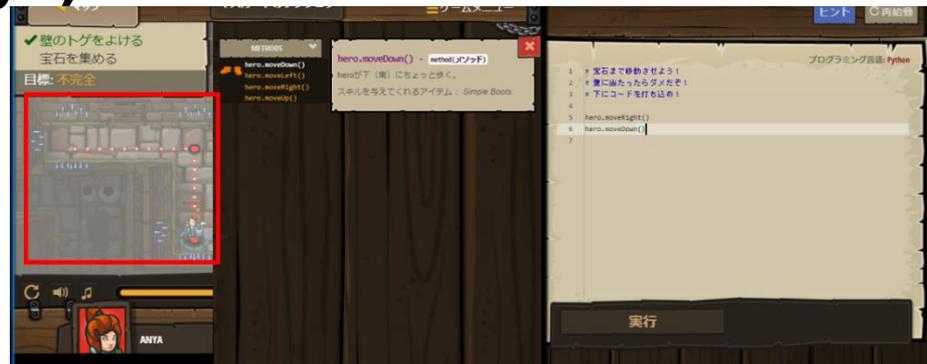
`moveDown()` は**メソッド**

- 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**の, **引数指定**

`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**：プログラム上の操作対象となるもの
- **メソッド**：オブジェクトが実行できる動作

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の活用

- 画面上の**キャラクター（hero）**に対して、**moveDown()**や**moveRight()**などの**メソッド**を適用し、意図した動作を実現。

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)

# 引数の使用方法



- **引数**を使用することで、**攻撃対象の指定** (hero.attack("fence")) や**移動距離の設定**など、より複雑な動作を実現。  
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)

# 演習 基本プログラミング

- オブジェクトの操作
- メソッドの呼び出し
- 引数の使用
- 文字列の扱い

# 学習準備 (1) : Webブラウザでのアクセス方法



<https://codecombat.com>

課金に関するメッセージが表示された場合は、無視を推奨。心配なことがあるときは、無理に使用しないこと。



課金のメッセージの例

# 学習準備 (2) : 初期設定と言語選択



- 「私は学生です」を選択
- 必要に応じて言語設定を日本語に変更
- アカウント登録やログインは行わない



The screenshot shows the CODE COMBAT website interface. At the top, there is a navigation bar with the CODE COMBAT logo on the left, and links for '教育者' (Educator), '親' (Parent), 'Play', 'ログイン' (Login), 'アカウント登録' (Account Registration), and '日本語' (Japanese). The '日本語' link is highlighted with a red box. Below the navigation bar, a teal banner reads '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 content area features a large heading '遊びの力を通じてコーディングを学ぶ' (Learn coding through the power of play). Below the heading, 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). The '私は学生です' button is highlighted with a red box. In the bottom right corner, there is an image of three students looking at a computer screen displaying a coding interface.

# 学習準備 (3) : アクセス権限の確認



- 「今すぐプレイ」を選択
- クラスコードの有無の確認では、**本授業では、クラスコードを使用しない形式**で学習を進める。

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:

続ける

---

持って無いけどやりたい！

**今すぐプレイ**

すでにアカウントをお持ちですか？ [サインイン](#)

# ゲーム開始 (1) : 基本ステージの選択

- 「キースガードのダンジョン」を選択し、「ゲームスタート」をクリック。基本的なプログラミング学習の出発点となる。



# ゲーム開始 (2) : 初期ダンジョンへの挑戦

- 最初のダンジョンを選択する。
- 画面上にはゴールが表示される。
- 音声が出るので、各自で適切な音量に調整すること。



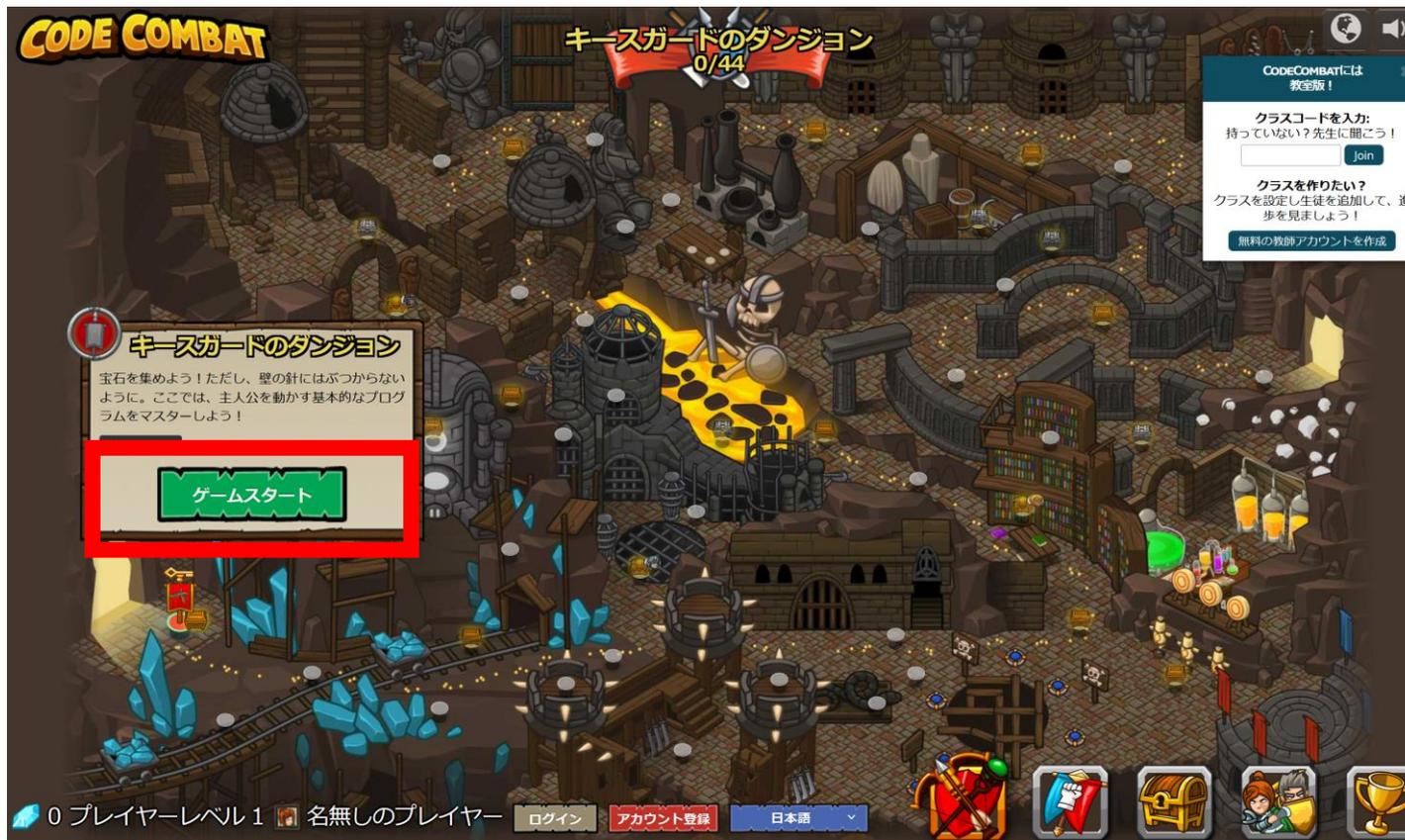
ゴール

最初のダンジョン

# ゲーム開始 (3) : ゲーム環境の起動



- 「ゲームスタート」をクリックして，起動する。
- 次以降の画面で，プログラミングについての画面が表示される。



# プログラミング環境 (1) : 言語設定の確認



- 「Python (デフォルト)」を選択し、「次へ」をクリック。
- この設定により、Python言語が選択される。



# プログラミング環境 (2) : 使用アイテムの選択



- 使用可能なアイテムをダブルクリックで選択
- ゲーム後半ではアイテムが増える (それに応じてメソッドが増える)



# プログラミング環境 (3) : ゲーム開始準備



- 「ゲームスタート」をクリックし、準備を完了する。



# プログラミング実行 (1) : レベルスタート手順



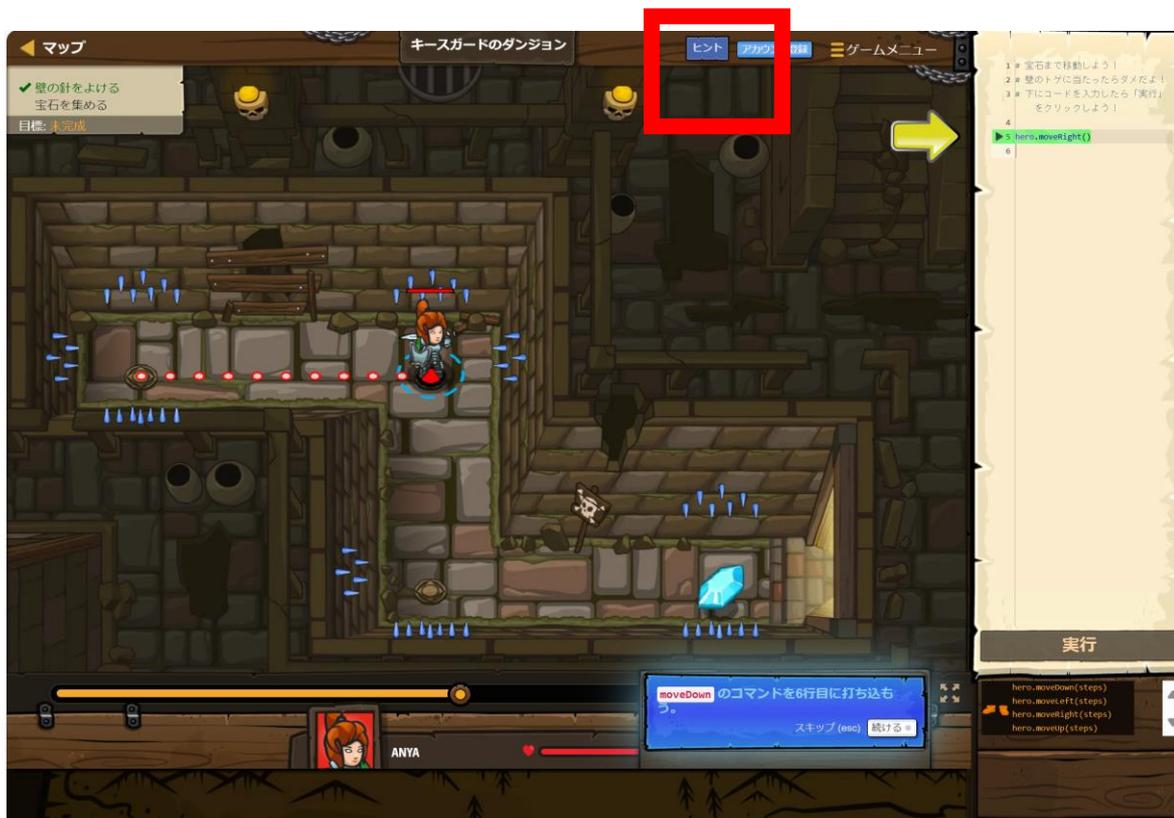
- 目標をよく読む
- 「ゲームスタート」をクリックし、開始する。



# プログラミング支援 (1) : ヒント機能の活用



- 「ヒント」ボタンをクリックすることで、課題解決のためのヒントを参照できる。
- ヒントが英語で表示される場合があるが、これは翻訳の不完全さによるものである。（動作不具合ではない）



# プログラミング支援 (2) : メソッド説明の参照



- 「メソッド」リストから**必要なメソッドを選択**すると、その**使用方法に関する説明**を確認できる。
- 説明が英語で表示される場合があるが、これは翻訳の不完全さによるものである。(動作不具合ではない)

The screenshot shows a game interface with a character in a dungeon. A tooltip for the `hero.moveDown(steps)` method is displayed, providing details such as the method name, description, example, required parameters, and default values. A red box highlights a list of methods in the bottom right corner, including `hero.moveDown(steps)`, `hero.moveLeft(steps)`, `hero.moveRight(steps)`, and `hero.moveUp(steps)`.



# プログラミング実践 (1) : 基本命令の入力

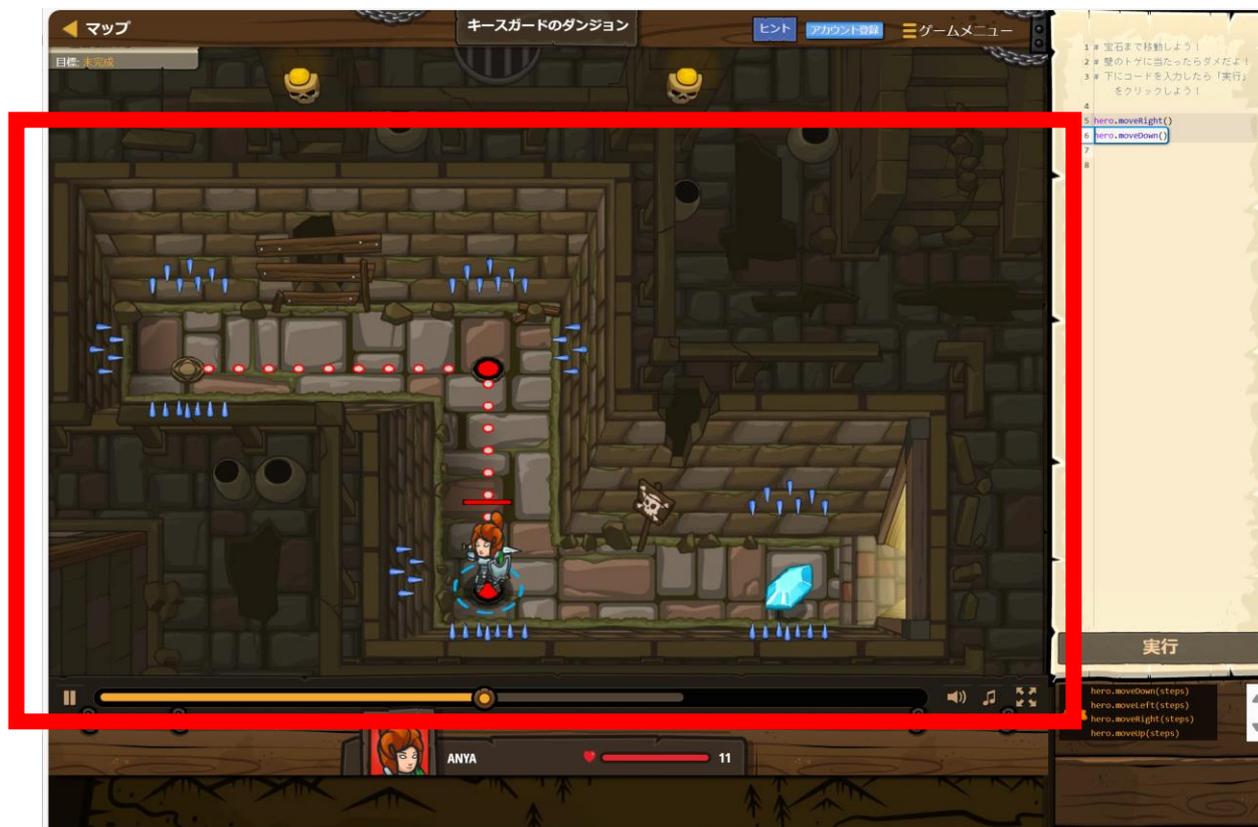
- 編集画面で「`hero.moveDown()`」などのプログラムを入力し、「実行」ボタンでその動作を確認する。



# コーディング実践 (2) : プログラム動作の確認



- 「実行」ボタンをクリックすると、入力したプログラムに従ってキャラクターが動作する。
- `hero.moveRight()`で右に移動, `hero.moveDown()`で下に移動する。



# 問題解決方法：学習支援機能の活用

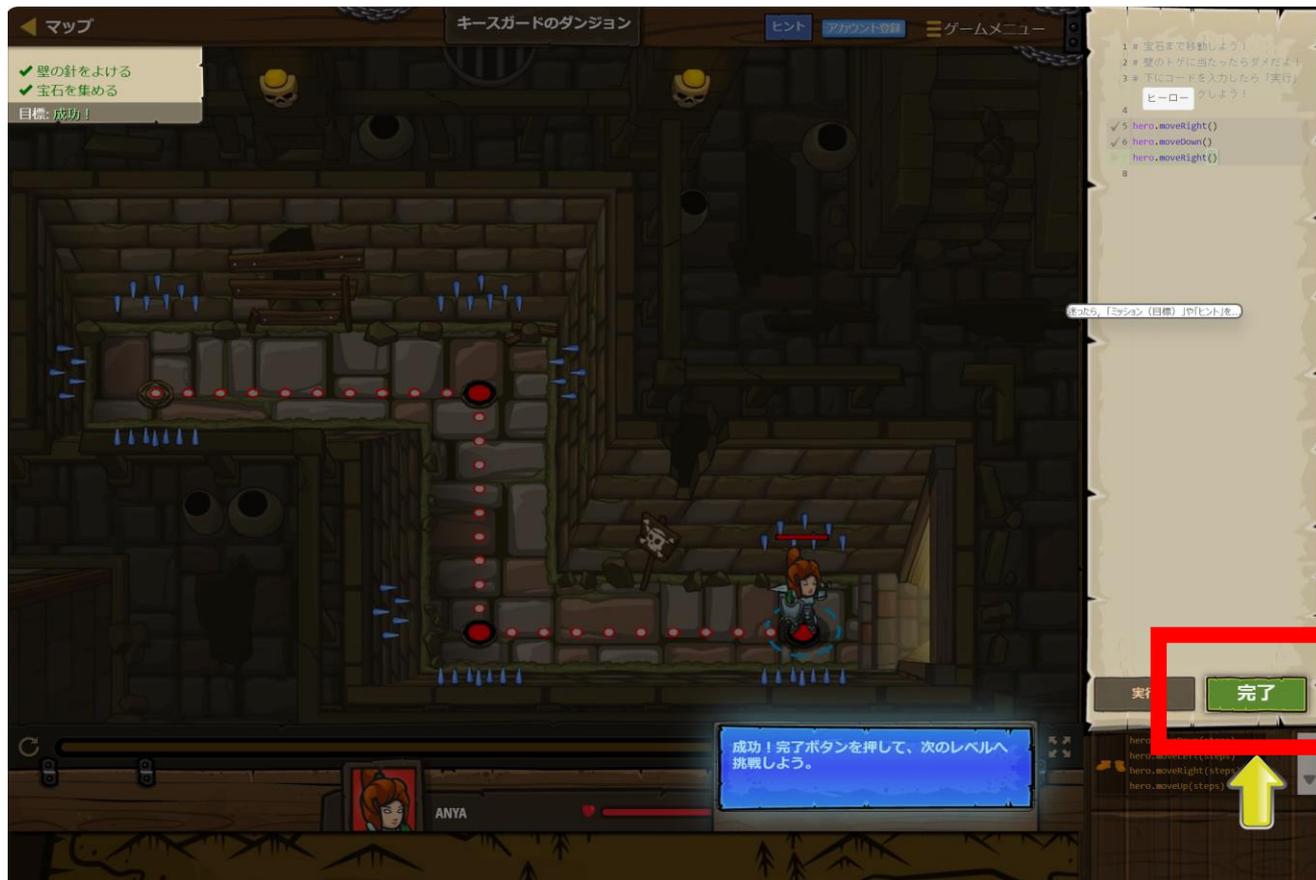
- プログラミングで迷った場合は、「**ミッション（目標）**」や「**ヒント**」を**確認**する。これらの機能を活用することで、効果的な学習が可能となる。

すべての  
目標達成  
を目指す



# 課題達成 (1) : プログラム完成までの手順

- 「**目標：成功！**」と表示されるまで，**編集画面でプログラムを書き換えて「実行」を繰り返す。**
- **目標達成後は「完了」をクリックする。**



# 課題達成 (2) : 次のステージへの進み方



- 完了確認後, 「続ける」をクリックして次に進む.
- これにより, 段階的な学習を進めることができる.



# 発展学習 (1) : 追加ステージの選択方法



- 赤い旗で示される次のレベルを選択できる。
- 青い旗はクリア済みのレベルを示し、「(ロック)」と表示される項目は有料コンテンツとなる。



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

# 発展学習 (2) : 新規装備の獲得と活用



- 新しいレベルでは**追加のアイテムが利用可能となる場合がある。**
- 装備は**ダブルクリック**で選択して使用する。



## 発展学習（3）：無料コンテンツの範囲確認



- 無料かつクラスコードを使用しない形式で、**Pythonの5つの基本レベル**までを学習できる。
- それ以降のレベルは有料コンテンツとなる。



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

# 発展学習（4）：追加機能とアカウント登録



以下の選択は各自の判断に委ねられる（授業の成績には影響しない）

- CODECOMBATプレミアム（有料）への登録
- クラスコードを使用した本格的な学習（無料，要ユーザ登録）



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

