

# 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情報工学における未来志向

②プログラミングの楽しさと達成感

③プログラミングの基本的な概念

# アウトライン

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

# 10-1. プログラミングの基礎①（オブジェクトとメソッド，引数，代入）

- **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
- **シミュレーション、大量データ処理、AI連携、ITシステム制作**など、さまざまな活動で、**プログラミング**は役立つ
- **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

# プログラミングの基礎



**最初は、基本的な要素を理解していく**

- オブジェクトとメソッド、引数、変数、代入
  - ← **今回の授業**
- クラス、条件分岐、繰り返し、リスト、パッケージ
  - ← **次回以降の授業**

# 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**

**hero.moveDown()**

**hero**      **オブジェクト**  
**moveDown()**   **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

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

# 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()
```

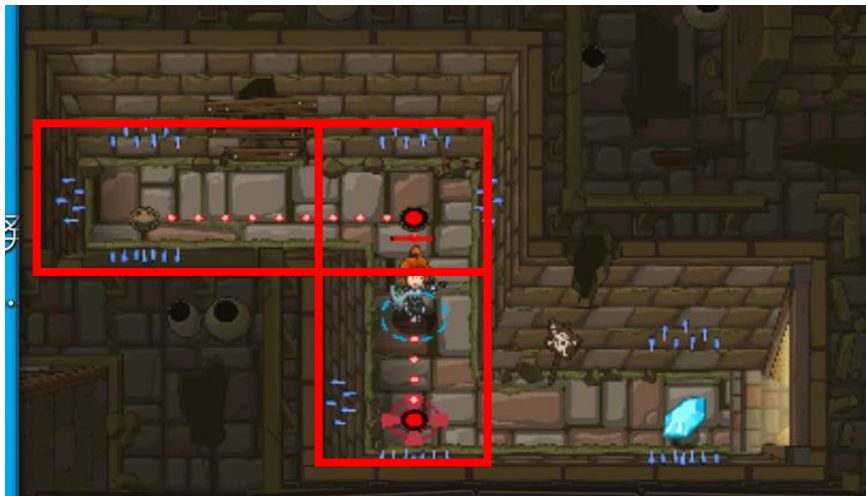
オブジェクトとメソッド  
(Python プログラム)



# オブジェクトとメソッド



オブジェクトが動く



実行画面

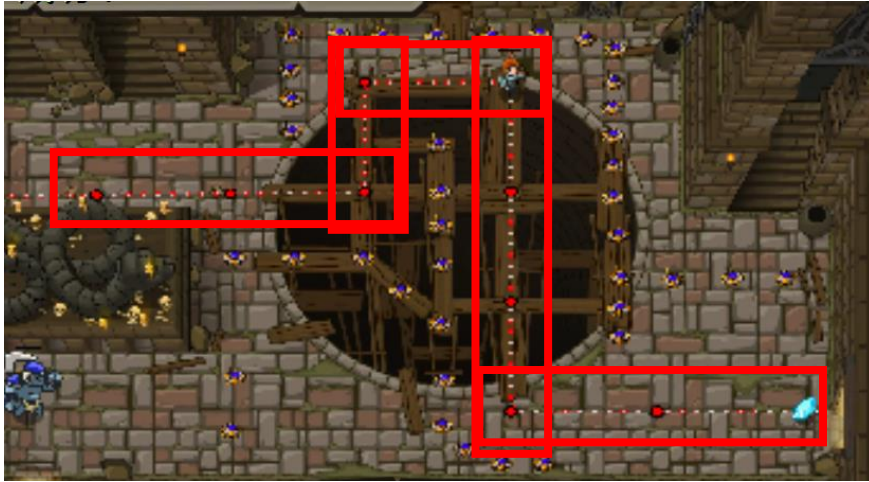
```
1  # 宝石まで移動させよう！  
2  # 壁に当たったらダメだぞ！  
3  # 下にコードを打ち込め！  
4  
✓ 5  hero.moveRight()  
▶ 6  hero.moveDown()
```

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。

# 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 (「変数」は, 数学の変数とは違う意味)
- **代入** : プログラムで, 「**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

**x = 100**

**プログラム**

# メソッドアクセスと代入



```
x = 100
a = x + 200
enemy1 = hero.findNearestEnemy()
hero.attack(enemy1)
```

Python プログラムの例

- **代入** : **オブジェクト名** + 「**=**」  
+ 式または値またはメソッド呼び出し
- **メソッドアクセス** : **オブジェクト名** + 「**.**」  
+ **メソッド名** + 「**()**」 (引数を付けることも)

Python プログラムでは、その他にも、属性アクセス、関数呼び出し、制御、「\*」、「+」などの演算子、コマンド、定義など

## 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



- コードコンバットは、Python プログラム学習などができるゲーム。オンラインで実行.
- ゲーム内で、キャラクタに指示をするためにプログラムを書く



# 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**

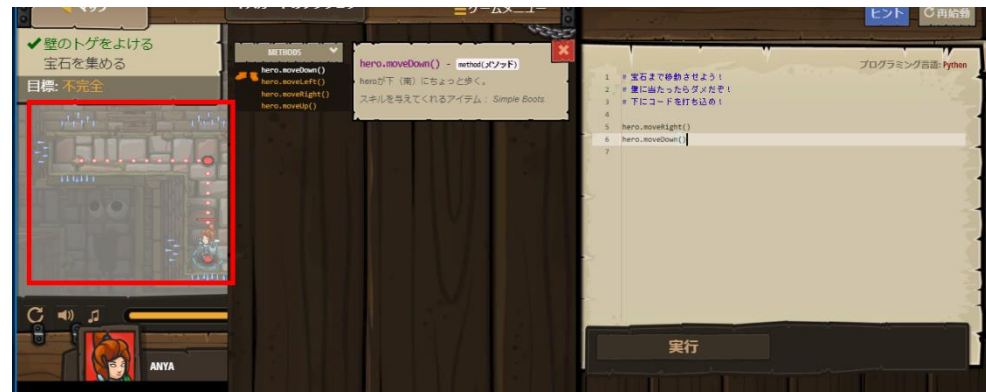
`moveDown()` は**メソッド**

- 文字列**は「"」または「'」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)`





# 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>

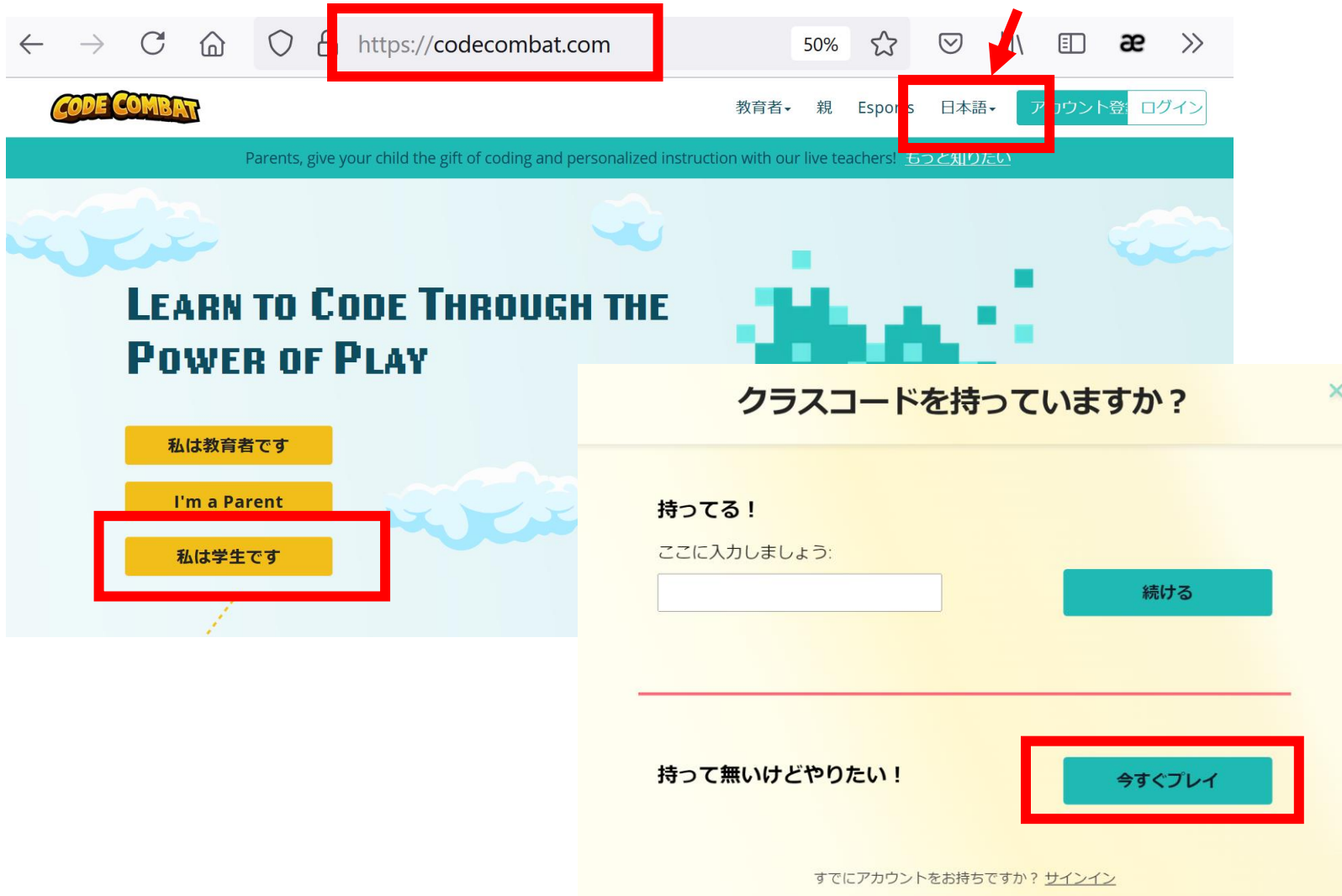
「課金のメッセージ」などで心配なことがあるときは、無理に使い続けないこと



課金のメッセージの例

## ② 「今すぐプレイ」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 CodeCombat website interface. The browser address bar displays `https://codecombat.com`. The navigation bar includes links for "教育者" (Educator), "親" (Parent), "Esports", "日本語" (Japanese), "アカウント登録" (Account Registration), and "ログイン" (Login). The "日本語" link is highlighted with a red box and a red arrow. Below the navigation bar, the main content area features the text "LEARN TO CODE THROUGH THE POWER OF PLAY" and three 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. A modal dialog box is open, asking "クラスコードを持っていますか？" (Do you have a class code?). It has two sections: "持ってる！" (I have it!) with a text input field and a "続ける" (Continue) button, and "持って無いけどやりたい！" (I don't have it but I want to try!) with a "今すぐプレイ" (Play now) button highlighted by a red box. At the bottom of the modal, there is a link: "すでにアカウントをお持ちですか？ サインイン" (Do you already have an account? Sign in).

# 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 0/20  
ロック

自分でレベルを造る事を習おう！

ウェブ開発 0/12  
ロック

HTMLをもっと学ぼう！

辺境の森林 0/121  
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

ゲーム開発2 0/29  
ロック

シンプルAIスクリプティング、ユニットやゴールをカスタマイズ出来る。

キースガードのダンジョン 0/44  
**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

CODECOMBAT! 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](#)

続けて HPML や スクリプトを習って！

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 本語



# 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 (デフォルト)」を選び、「次へ」をクリック

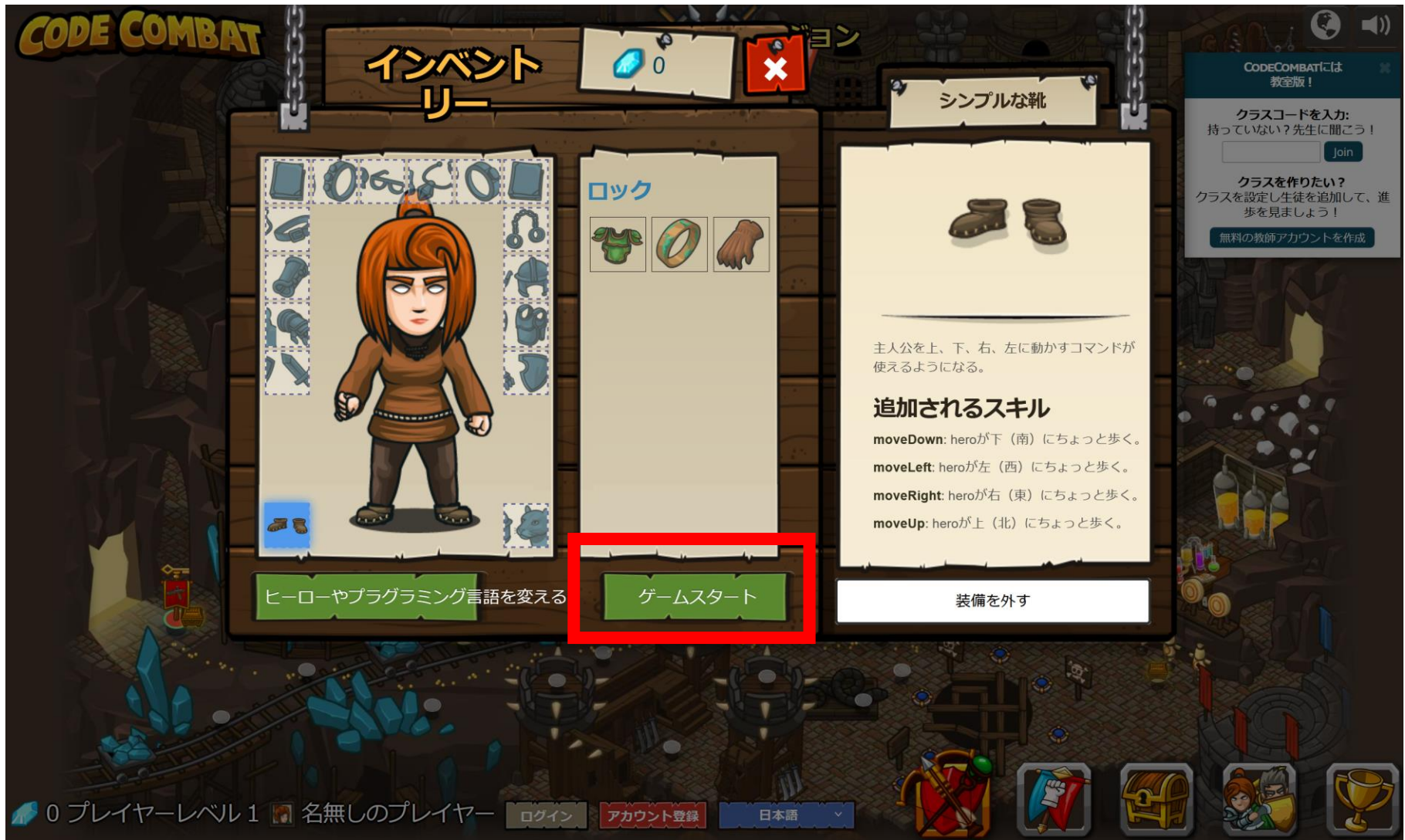


# ⑦ 使用可能なアイテムを選ぶ (ダブルクリック)





## 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



マップ キースガードのダンジョン カウント登録 ゲームメニュー

### 目標

壁のトゲをよける  
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



The screenshot shows the game interface for 'Anyade'. At the top, there are navigation buttons: 'マップ' (Map), 'キースガードのダンジョン' (Keith Guard's Dungeon), and 'ゲームメニュー' (Game Menu). A red box highlights the 'ヒント' (Hint) button in the top right corner. On the left, a status bar shows 'トゲをよける 宝石を集める' (Avoid thorns, collect gems) and '目標: 未完成' (Goal: Incomplete). The main area is a 2D dungeon with a character named Anya. A yellow arrow points from the character towards the right. At the bottom left, there is a character portrait of Anya and a text box that says 'moveDown のコマンドを6行目に打ち込もう' (Enter the moveDown command in line 6) with 'スキップ (esc)' and '続ける' (Continue) buttons. On the right, a 'ヒント' (Hint) panel is open, showing a list of instructions in Japanese and a Python code snippet: 

```
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
```

 Below the hint panel is an '実行' (Execute) button and a list of movement commands: 

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

# 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



The screenshot shows a game interface with a top navigation bar containing 'マップ' (Map), 'キースガードのダンジョン' (Keith Guard's Dungeon), and 'ゲームメニュー' (Game Menu). On the left, a quest log indicates 'トゲをよける 宝石を集める' (Avoid thorns, collect gems) with a '目標: 未完成' (Goal: Incomplete) status. The main area is a 2D dungeon with a character and a blue gem. A yellow arrow points from the gem to the hint panel on the right. The hint panel is titled 'ヒント' (Hint) and shows a Python code snippet: 'hero.moveRight()'. Below the code is a section titled '実行' (Execute) which contains a list of methods: 'hero.moveDown(steps)', 'hero.moveLeft(steps)', 'hero.moveRight(steps)', and 'hero.moveUp(steps)'. A red box highlights this list. At the bottom, a blue instruction box says 'moveDown のコマンドを6行目に打ち込もう' (Enter the moveDown command on line 6) with 'スキップ (esc)' (Skip) and '続ける' (Continue) buttons. The character's name 'ANYA' is visible in the bottom left corner.

マップ キースガードのダンジョン ゲームメニュー ヒント

✓ トゲをよける  
宝石を集める  
目標: 未完成

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
```

実行

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

moveDown のコマンドを6行目に打ち込もう  
スキップ (esc) 続ける

ANYA

# 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**

The screenshot displays a game interface with a top navigation bar containing 'マップ' (Map), 'キースガードのダンジョン' (Keith Guard's Dungeon), and 'ゲームメニュー' (Game Menu). The main area is split into two panels. The left panel shows a character in a dungeon environment, with a red rectangle highlighting a specific area. The right panel is a code editor titled 'プログラミング言語: Python' (Programming Language: Python) with a 'ヒント' (Hint) button. It contains the following code:

```
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 is an '実行' (Execute) button. At the bottom of the screen, there is a status bar showing the character's name 'ANYA', a heart icon, and the number '11'. The bottom right corner of the image has the number '30'.



# 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

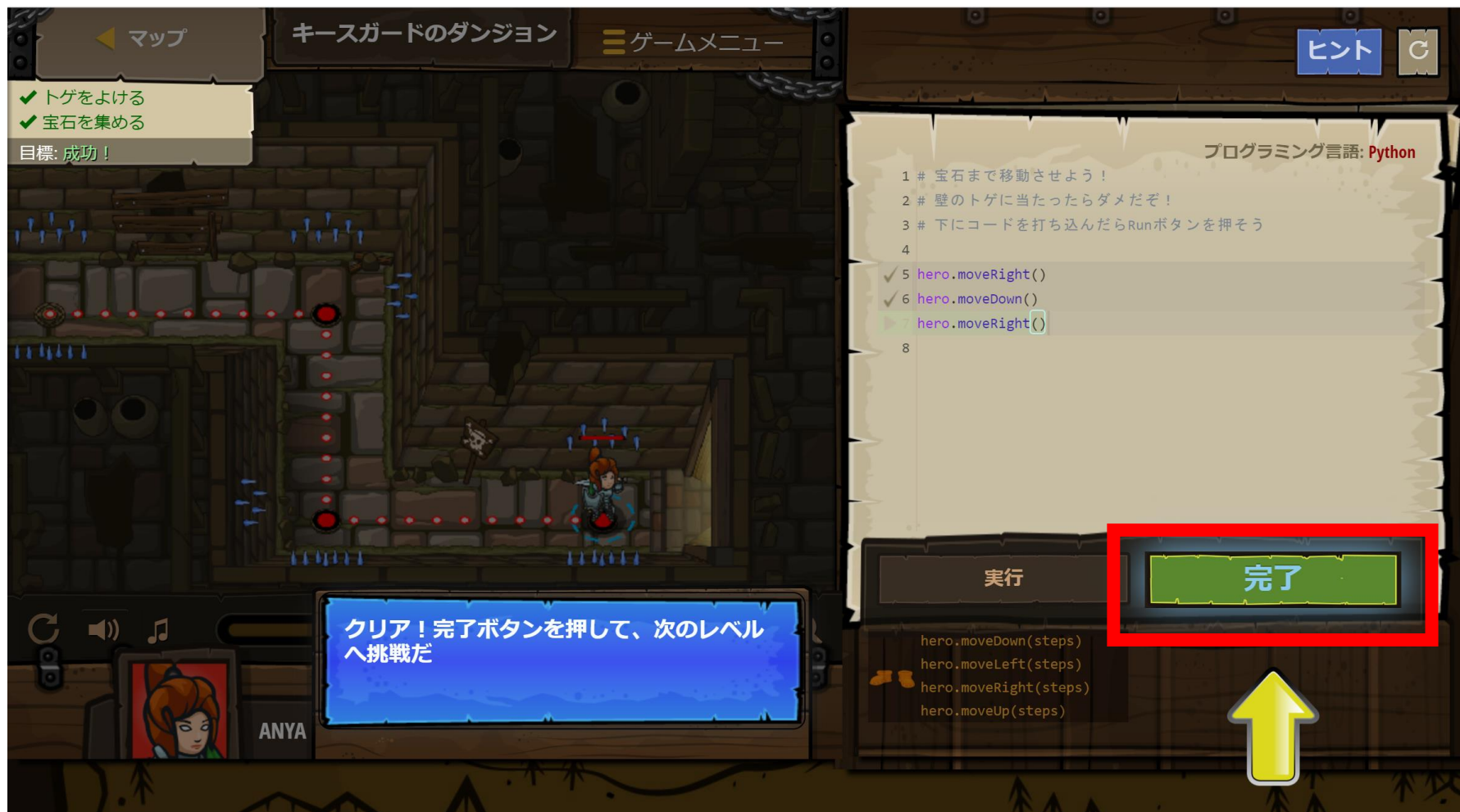
すべての  
目標達成  
を目指す



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。



木法が成功したら、「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

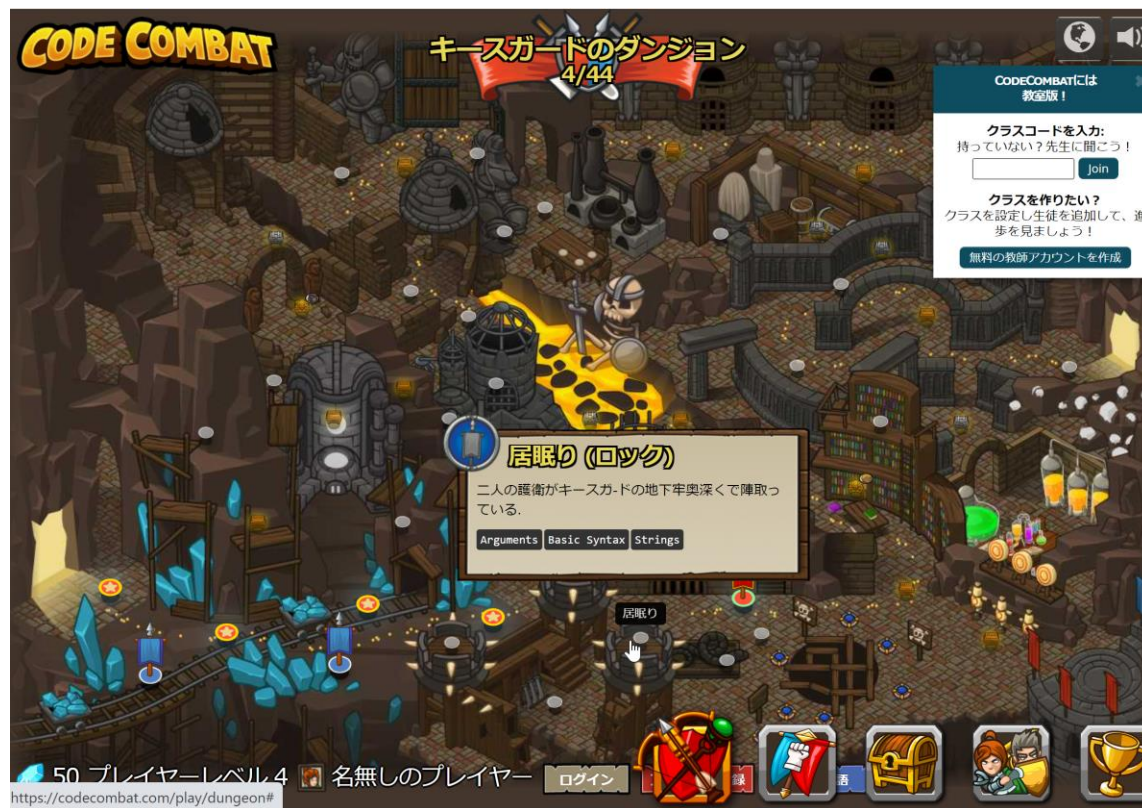
- 赤い旗を選択できる
- 有料のものもある



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

# 続けてみる



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

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。)

①CODECOMBATプレミアムへの登録

・・・ 有料

②クラスコード（大学のセレッソで案内）の利用

・・・ 登録が必要。使い方や、何ができるかは、  
各自で自主的に試し、調べてください

## 10-3. プログラミングの楽しさと達成感

# 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**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

- **自分のアイデアを形にすることで得られる達成感**
- 自分でデザインし、問題が生じたときは自分で解決していく。
- **自分の手でプログラムを完成させるプロセスは、大いに充実感をもたらすもの。**

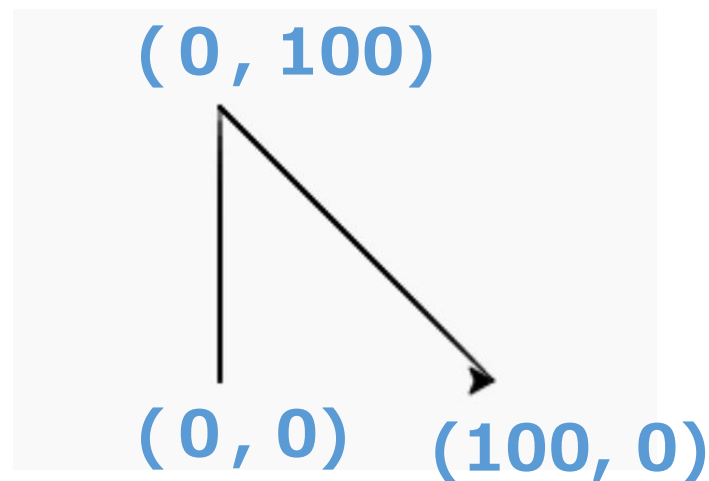
# タートルグラフィックス



カーソルを使って絵を描く

```
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



```
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

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



# 演習 プログラミングはクリエイティブ

## 【構成】

- ①オブジェクト生成、形状の設定
- ②移動
- ③色、円

## 【トピックス】

- ・ モジュールのインポート
- ・ オブジェクトの生成
- ・ メソッド（移動）



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。

- 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

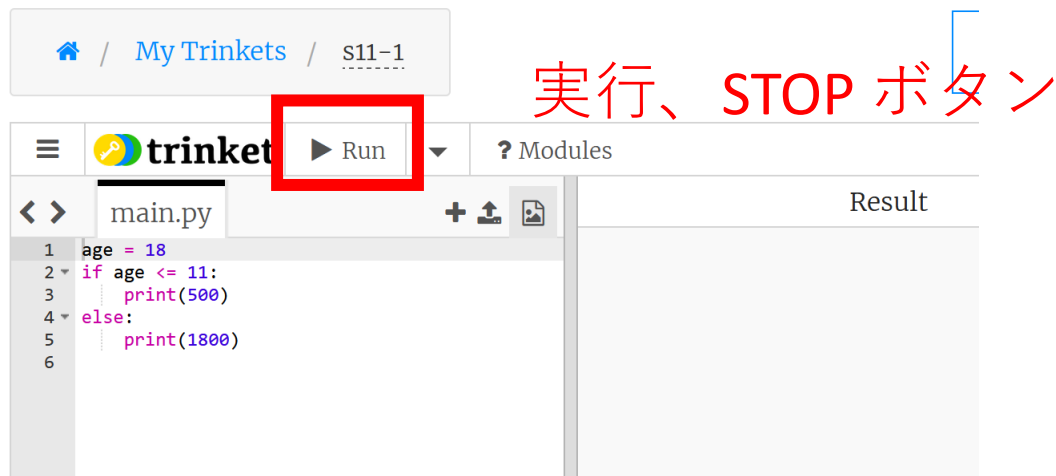
# trinket でのプログラム実行



- 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>

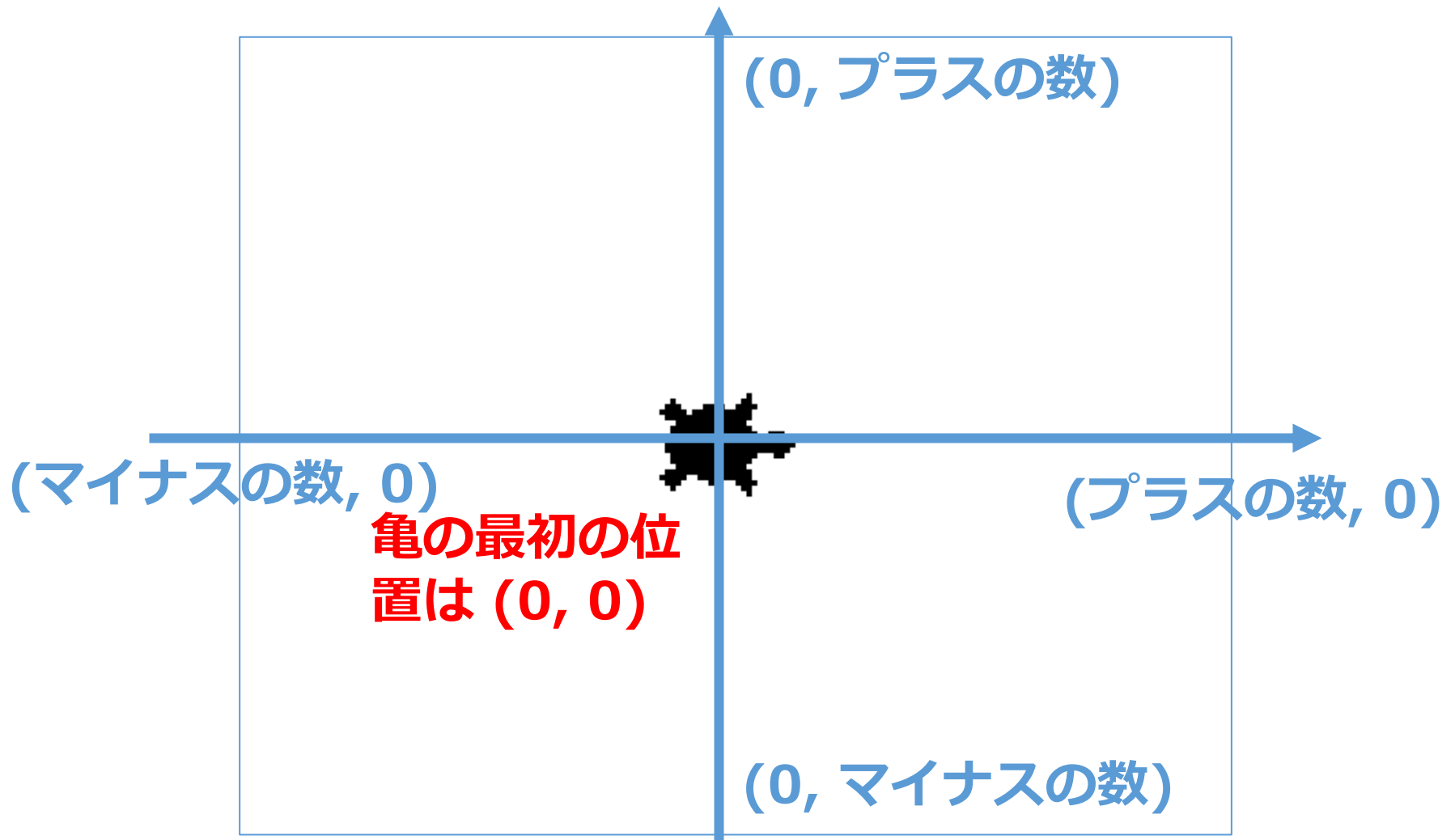
のように、違うプログラムには違う URL が割り当てられる



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



① 1 つめ

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

② 2 つめ

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

③ 3 つめ

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

①

```
import turtle
```

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

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

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

モジュールのインポート

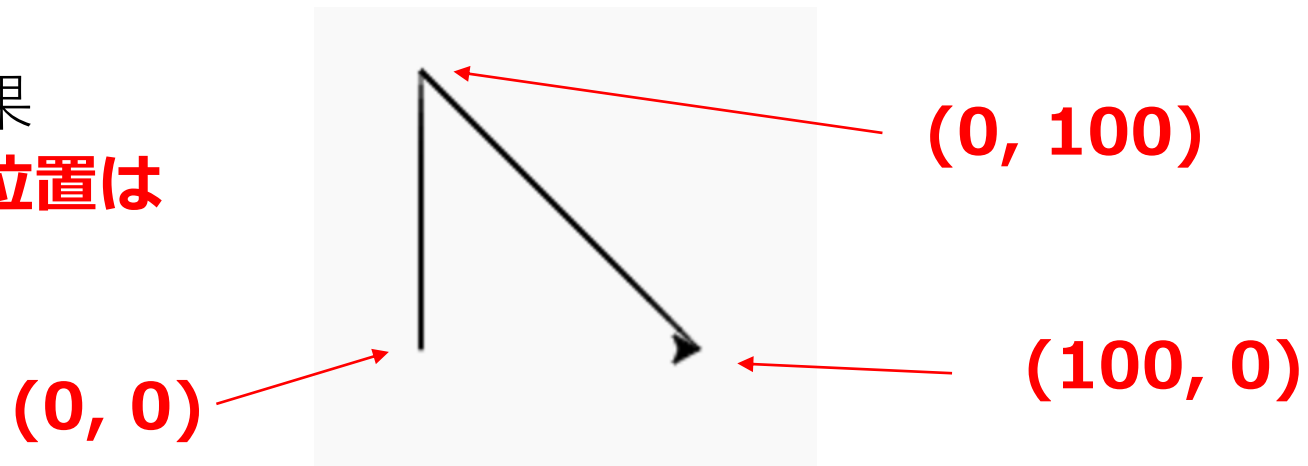
オブジェクト生成。t へのセット。

**(0, 100) への移動**

**(100, 0) への移動**

実行結果

**最初の位置は  
(0, 0)**



②

```
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)
```

モジュールのインポート

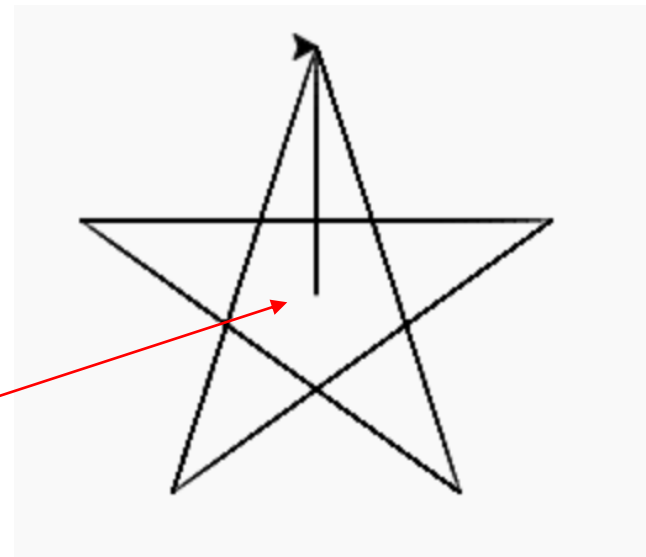
オブジェクト生成。t へのセット。

**移動**

実行結果

**最初の位置は  
(0, 0)**

**(0, 0)**



### ③色、円



```
import turtle
```

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

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

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

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

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

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

モジュールのインポート

オブジェクト生成。t へのセット。

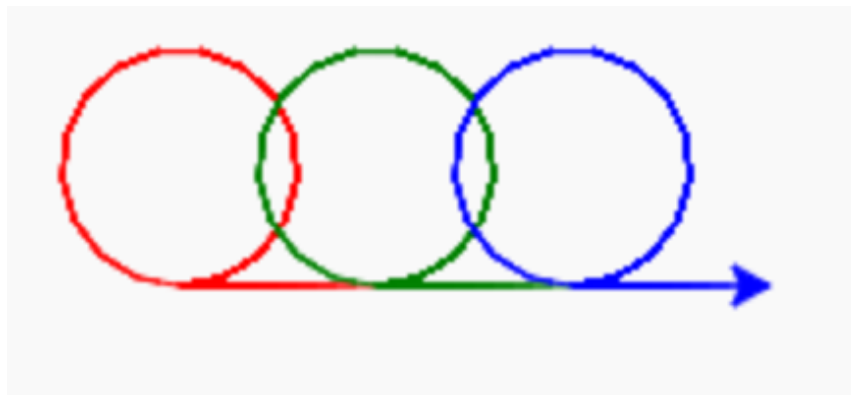
色は、赤、緑、青

色を変える

半径 30 の円

前に50進む

実行結果





## ①情報工学における未来志向

- プログラミングの学習は、未来志向のスキルの取得に役立つ
- プログラミングの経験により自身が深まる
- 将来の多様なキャリアパスへの視野（IT企業、製造業、クリエイティブな職種など）

## ②プログラミングの楽しさと達成感

- 自分のアイデアを形にする楽しさと達成感
- 自分でデザインし、問題が生じたときは自分で解決していく

## ③ プログラミングの基本的な概念

**プログラミングの基本的な概念（オブジェクト、メソッド、引数、代入など）を習得することは、ソフトウェア開発の根幹です。これらは、コンピュータを活用して種々の作業を自動化し効率化することを可能にし、現代社会における重要な能力です。多様なキャリアパスを切り開く道になります。**

# 全体まとめ



- **オブジェクト**：操作や処理の対象となるもの。
- **変数**：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**：プログラムで変数に値を保存する操作。  
「**x = 100**」は**変数x**に**100**を**代入**する。
- **メソッド**：オブジェクトに属する機能や操作。
- **メソッドアクセス**：オブジェクト名 + . + メソッド名 + ()  
(引数を付けることも)

hero.moveDown()ではheroはオブジェクト、moveDown()はメソッド。

- **引数**：メソッドが行う操作の詳細情報。

例えば、Hero.attack("fence", 36, 26)