

cs-7. プログラミング思考の基礎

(コンピューターサイエンス)

金子邦彦



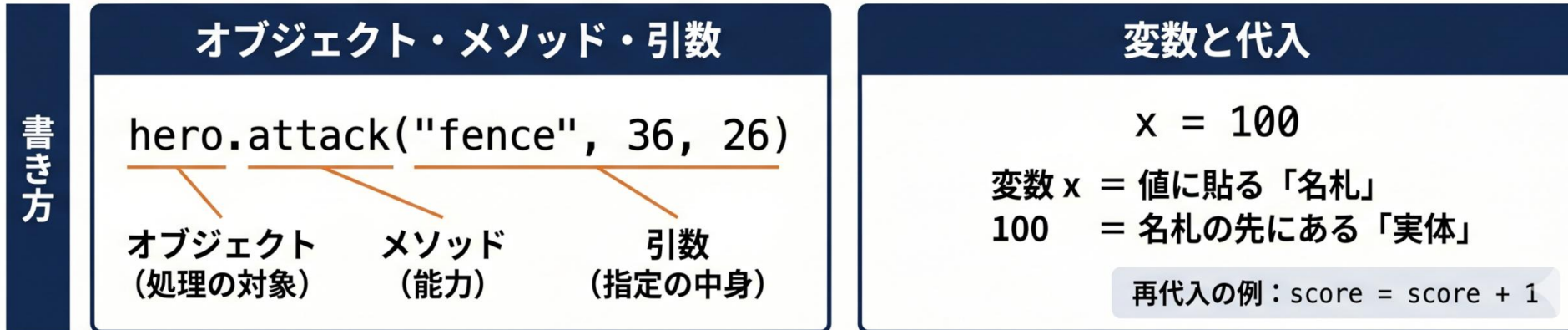
「コンピューターサイエンス」第7回の内容



考え方（プログラミング的思考の四要素）



書き方（コードに現れる基本要素）

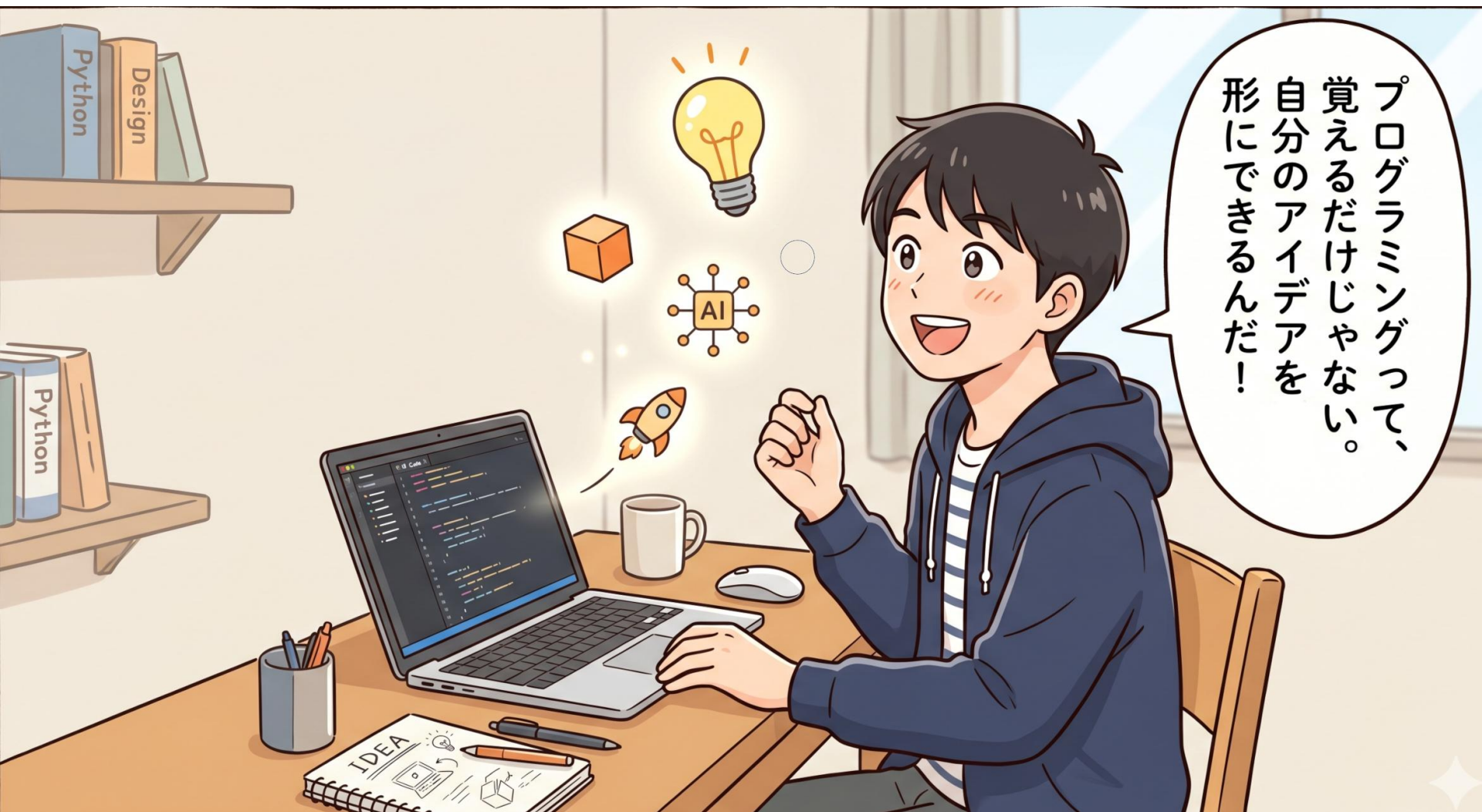


※ オブジェクト・メソッド・引数・変数などの用語は本授業で説明する



7-1. プログラミングとは何か

- 人間の力を増幅する



情報工学科の学び

学ぶこと

コンピュータ



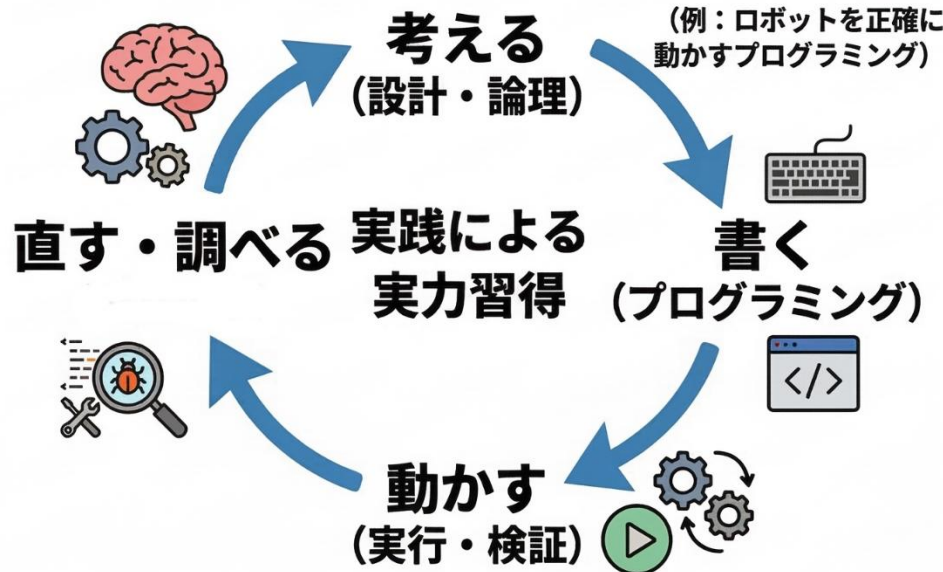
インターネット

デジタル活用



情報工学の
面白さ

実践的な学びのサイクル



学習スタイル



個人ワーク
(一人で取り組む)



グループワーク
(仲間と助け合う)

創造



(新しいアイデア)

設計



(システム全体)

具現化



(形にする・実装)

プログラミングは創造的 — 授業の狙い

×

覚える、
問題を解くだけ

捉え直す

○

創造的な作業

プログラミング = 人間の力を増幅する営み

ゴール: 応用へ進むための土台をつくる

説明パート

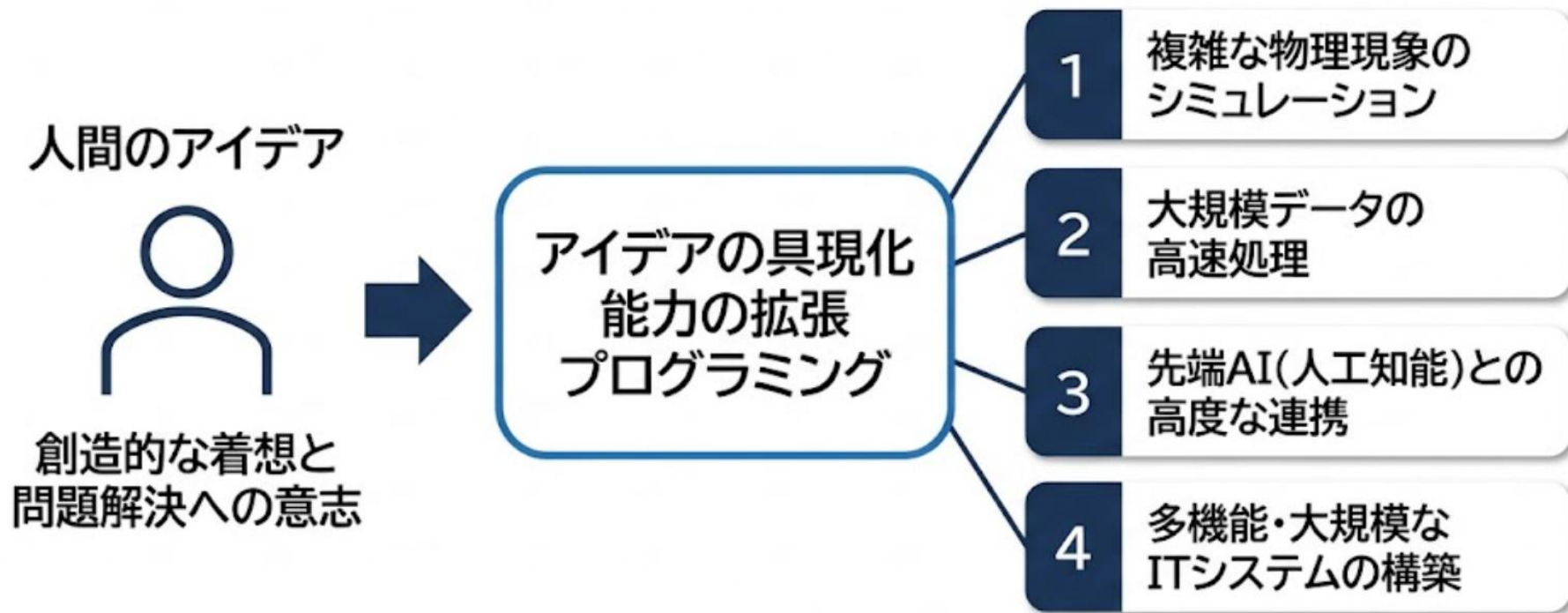
プログラミング的思考の枠組みを学ぶ

+

演習パート

CodeCombat で楽しく体得する

プログラミングで何ができるのか



誤解: 単なる『覚えて、問題を解くだけの勉強』

正解: 『自分のアイデアを形にする』創造的な作業

プログラムとは — 命令を書いた手順の集まり



プログラムは、命令を書いた手順の集まり

指示を与える

自動で処理

結果を得る



合計40歩進んだ!

だから複雑な作業も**自動化**・**効率化**できる

プログラミングを学ぶことで何が身につくか



将来へ

Step1

使いこなす

コンピュータを自分の
思い通りに活用できる

Step2

増幅する

コンピュータの活用
で、自分自身の能力を
増幅できる

Step3

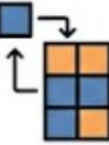
成長する

創造力・発想力・
デザイン力・行動力・
チャレンジ精神・
論理的思考力

Step4

広げる

可能性を広げる。
プログラミングは
『部品』を組み立て
て作品を作ることに
似ている
→ エンジニアの素養が
身につく





これからは"AIに指示してプログラミングする"側に立つ — LLMは使いこなす道具

プログラミング学習における LLM の活用

LLM とは

- 大規模言語モデル (Large Language Model)
- 自然言語でやり取りし、コードの生成・説明・修正を支援する AI

代表例：ChatGPT / Claude / Google Colab の AI アシスタント

特性を
理解

学習パートナー としての二面性

強み

強力な学習パートナーになる (コード生成・説明・修正を即座に支援)

注意点

誤ったコードを自信ありげに提示することがある

正しく
使う

望ましい使い方

1. 生成されたコードは必ず自分で実行して確かめる
2. なぜそう書くのかを LLM に説明させる
3. まずは LLM に頼らず基本概念を体得する

LLM活用の具体例

大前提：生成されたコードはそのまま信用せず、自分で実行して結果を確かめること

活用例1：コードの生成を頼む

依頼の例

Pythonで [1, 2, 3] と [4, 5, 5] の折れ線グラフを描くコードを書いて



LLMの応答

plt.plot(...) を使ったコードが返ってくる



学習者の行動

Colabなどで実行し、
思いどおりのグラフが出るかを確認する

活用例2：コードの意味を説明させる

依頼の例

この1行のどこがオブジェクトで、どこがメソッドで、どこが引数か説明して



LLMの応答

各部分の役割を解説してくれる



学習者の行動

『なぜそう書くのか』を
説明させることで、自分の理解を固める

LLM活用の具体例 – やるべき使い方とやってはいけない使い方

エラーの原因を尋ねる

推奨される使い方

実行してエラーが出る

↓
エラーメッセージをそのまま貼り付ける

↓
「このエラーの意味と直し方を教えて」と質問

↓
示された直し方を理解し、自分で修正して再実行

書き換え（リファクタリング）を頼む

推奨される使い方

• 「このコードを変数を使って書き直して」

• 「score = score + 1 のような再代入の例に書き換えて」

学んだ概念に沿った書き換えを依頼することで、概念と実コードの対応が見えやすくなる

やってはいけない使い方

避けるべき使い方

- 生成されたコードをコピーして提出するだけ
- 自分では一度も実行しない
- 意味を確認しない

強力な学習パートナーではなく、誤りを増幅する道具になってしまう

LLMは「自分で実行し、意味を確認する」使い方で初めて学習パートナーになる



7-2. プログラミング的思考と実践



ねこが進むほどスコアが増える — この小さな世界から、プログラミングが始まる

ねこの行動と得点（スコア）

ねこが10歩進むたびにスコアが1増え、10回くりかえして10になる



10歩を10回くりかえすと、スコアは10になる

対象・構造・条件の3点で世界を捉える

対象 (世界にあるモノ)

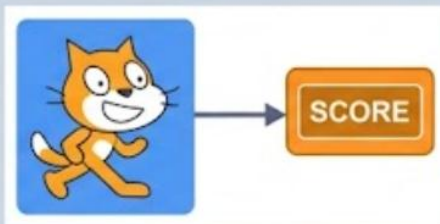


→ ねこ：
この世界で
動く 主体



→ スコア：
ねこの成果
果を測る量

構造 (モノどうしの関係)



→ ねこが前に進むこと
と、スコアが増えるこ
とが結びついている

前進と得点が連動しているのが、
この世界のしくみ

条件 (世界の決まり)

ねこは1回に
10歩ずつ進む

ねこが進むと、
スコアが1増える

ねこの行動とスコアを
支配するルール

プログラムと『3つの道具』

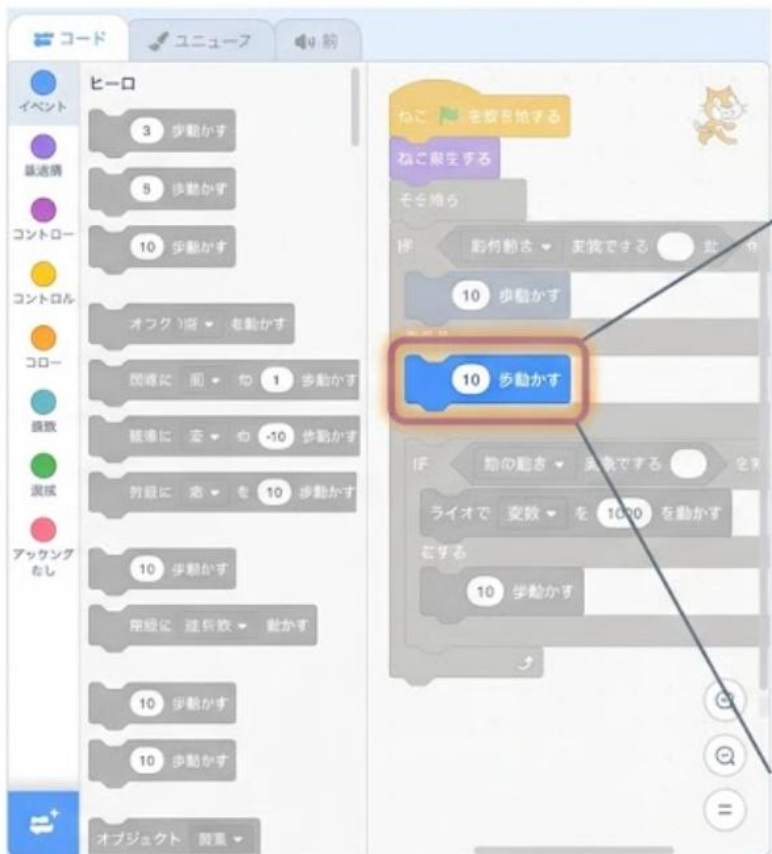


このプログラムが『3つの道具』でできている



- ① オブジェクト・メソッド・引数
- ② 変数と代入
- ③ アルゴリズムの感覚

① オブジェクト・メソッド・引数



オブジェクト＝動かす対象（ねこ）



メソッド＝対象ができる動作

引数＝動作に渡す数値

② 変数と代入



Scratch script showing variable initialization and incrementation:

- Flag clicked: Set score to 11
- New block: Set score to 0
- Set score to 10
- Increment score by 1
- Repeat loop: Set score to 0, then increment score by 1

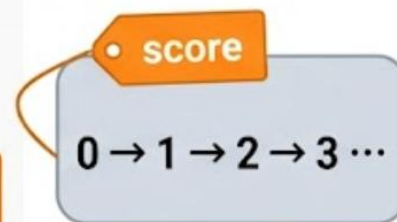
「score を 0 にする」
→ 代入：名札 score に値0を入れる（はじめの得点）



score を 0 にする



score を 1 ずつ変える



「score を 1 ずつ変える」
→ 再代入：今の score に1を足して入れ直す

③ アルゴリズムの感覚

手順1 →  旗が押されたとき (始まり1の合図)

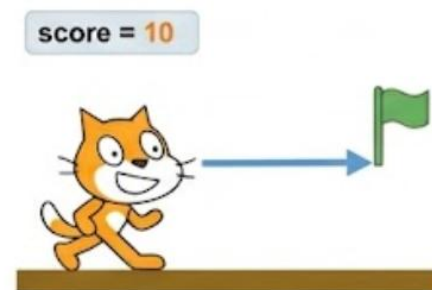
手順2 →  score を 0 にする (準備)

手順3 → 10回繰り返す  10 回繰り返す (くりかえし)

手順4 → 10歩動かす+
scoreを1つ変える
(くりかえす中身)

 10 歩動かす

 score を 1 ずつ変える



どんな順番でブロックを並べればゴールに着くかを考える力

プログラムで使用される『3つの道具』

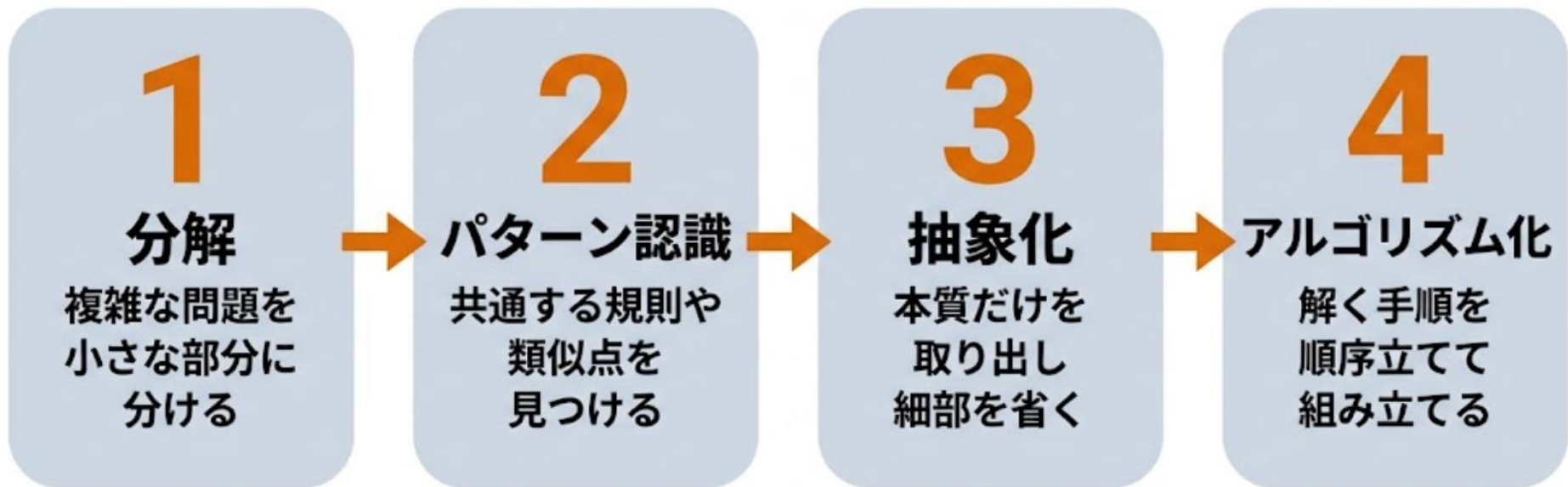


① オブジェクト・メソッド・引数
ねこ・動かす・10のまとまり

② 変数と代入
score に値を入れ、書き換える

③ アルゴリズムの感覚
ブロックを正しい順に積み重ねる

複雑な問題を解くための四つの考え方



現代の知的作業を支える基礎能力

『ねこが進むとスコアが増える』世界を、 分解・パターン認識・抽象化・アルゴリズム化

1 分解

10歩動かす



世界のできごとを最小の動作に分ける：
『10歩動かす』と
『scoreを1ずつ変える』の2つ

2 パターン認識

10歩動かす

→ scoreを1ずつ変える

くりかえし現れる
同じ形を見つける：
『10歩動かす →
scoreを1ずつ変える』
の組が何度も現れる

3 抽象化

N歩は N点

例
5歩なら？

M歩は M点

例
3点なら？

変わりうる数値を見
抜く：歩数や得点はル
ールが変われば変わる。
『N歩進むとM点増え
る』と一般化し、10や
1はその一例と捉える

4 アルゴリズム化

scoreを0にする（準備）

決めた回数くりかえす

N歩動かす+

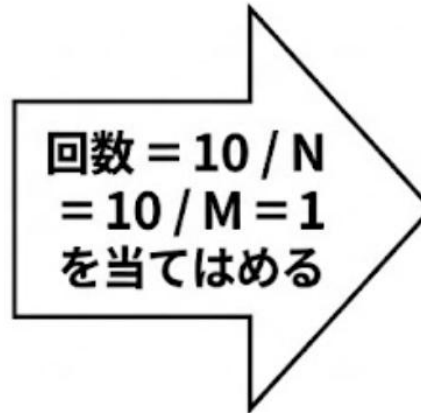
scoreをM点ずつ変える

N=10, M=1

一般化した手順を順
序立てる：準備して
てから、決めた回数
くりかえす形に並べ、
今回はN=10・M=1を
当てはめる

アルゴリズム化の結果に、10, 10, 1の具体的な値を設定

一般の形 (N・M・回数は変わらうる)



Scratchのブロック (数値を当てはめた形)



7-3. 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), which is currently "未完成" (incomplete). The game menu shows "マップ" (Map), "ゲームメニュー" (Game Menu), and "ヒント" (Hint). The code editor on the right is set to "プログラミング言語: Python" and 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 the "実行" (Run) button and a list of available methods: `hero.moveDown(steps)`, `hero.moveLeft(steps)`, `hero.moveRight(steps)`, and `hero.moveUp(steps)`. The character's health bar at the bottom shows 11.

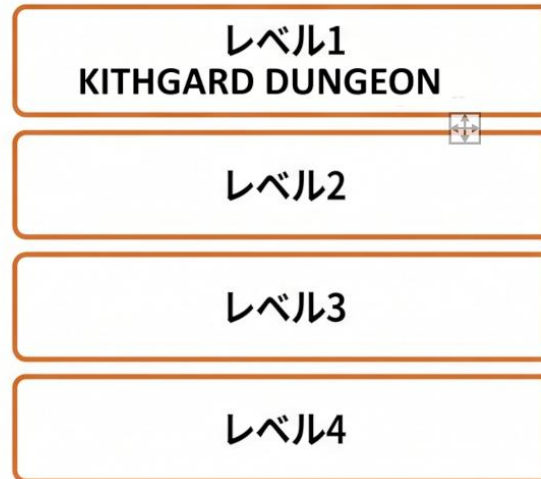
CodeCombat 初心者向けまとめ (登録なし・無料の範囲)

開始までのステップ



アカウント登録不要
クラスコード不要

無料で遊べる範囲



無料・登録なしで遊べるのは最初の4レベル。5つ目からは有料(プラチナ購入)のため、本授業では先に進まない

学べるコードの形

```
hero.moveDown()  
→ オブジェクトとメソッド  
  
hero.attack("fence", 36, 26)  
→ 引数付き呼び出しと文字列のクォート
```

演習の基本サイクル



無料範囲だけで本授業の学習目標は達成できるため、課金は不要

CodeCombat で学ぶ Python基礎の3トピック

① オブジェクトとメソッド

`hero.moveDown()`

↑
オブジェクト

↑
メソッド

ドット (.) でオブジェクトとメソッドをつなぐ

② 文字列

ダブルクォーテーション

“Hello”

‘Hello’

↑ ↑
シングルクォーテーション

文字列はクォーテーションで囲む

③ 引数付きメソッド呼び出し

引数 (複数)

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

`hero.attack(enemy1)`

↑
引数 (1つ)

引数はある場合もない場合もある

CodeCombat を通して、オブジェクト・メソッド・文字列・引数を理解

処理を表す三点セット

hero.moveRight(3)

オブジェクト

処理の対象を表すもの

メソッド

オブジェクトの能力を表すもの

引数

能力を細かく指定するもの

オブジェクト・メソッド



オブジェクトが動く



実行画面

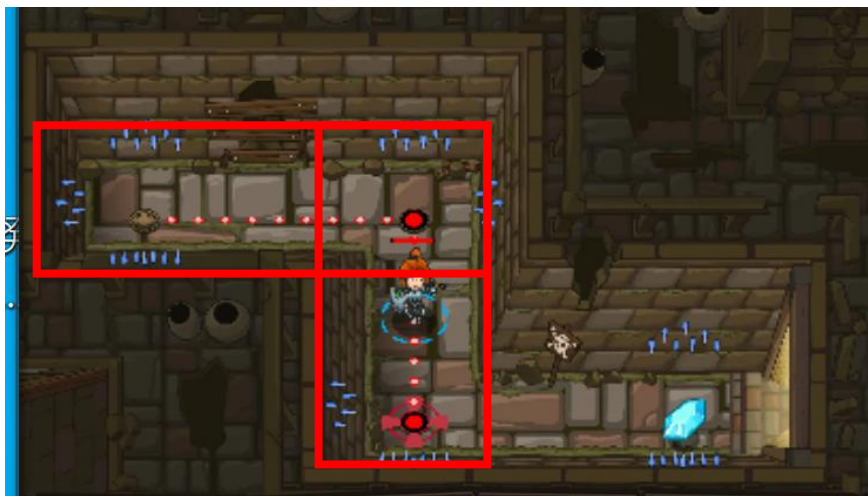
```
1 # 宝石まで移動させよう！
2 # 壁に当たったらダメだぞ！
3 # 下にコードを打ち込め！
4
5 ▶ hero.moveRight()
```

オブジェクトとメソッド
(Pythonプログラム)

オブジェクト・メソッド



オブジェクトが動く



実行画面

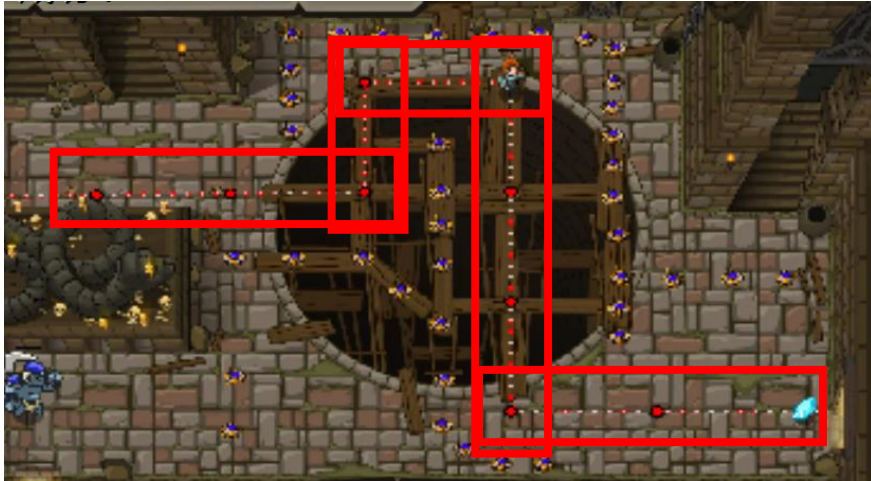
```
1 # 宝石まで移動させよう！  
2 # 壁に当たったらダメだぞ！  
3 # 下にコードを打ち込め！  
4  
✓ 5 hero.moveRight()  
▶ 6 hero.moveDown()
```

オブジェクトとメソッド
(Pythonプログラム)

オブジェクト・メソッド・引数



オブジェクトが動く



実行画面

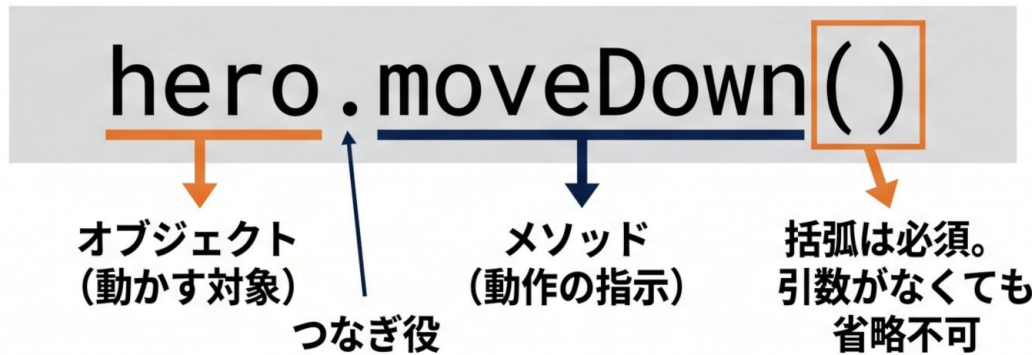
```
1 # Use arguments with  
   farther.  
▶ 2 hero.moveRight(3)  
3  hero.moveUp()  
4  hero.moveRight()  
5  hero.moveDown(3)  
6  hero.moveRight(3)
```

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。

プログラミング基礎 メソッド呼び出しの読み方

記法 オブジェクト.メソッド()



引数の書き方ルール

引数が複数あるときはカンマで区切る

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

文字列はクォート (" ") で囲む

`"fence"` ← クォートあり

数値はそのまま書く (クォート不要)

`36, 26` ← クォートなし

同じメソッドでも引数で挙動が変わる

`hero.attack("fence")` → 柵を攻撃
`hero.attack("ogre")` → オーガを攻撃

引数が動作の対象や量を決める

メソッド名から動作を推測

`moveDown` → 下へ移動
`moveRight` → 右へ移動
`attack` → 攻撃する

英単語の意味が動作のヒント

プログラムは上から順に実行

1 `hero.moveDown()`
↓
2 `hero.moveRight()`
↓
3 `hero.attack("ogre")`

書いた順番どおりに動作する

CodeCombat手順

プログラミングの練習だけでなく、ゲーム要素（パズル）も楽しんで解こう

ステージA 開始準備

- Webブラウザで <https://codecombat.com> を開く
- 表示が日本語でないときは、右上のメニューで「English US」を「日本語」に
- 「Start Playing」をクリック • 「今すぐプレイ」 登録・ログインは行わない。クラスコードはない

ステージB ゲーム起動

- KITHGARD DUNGEON を選び「ゲームスタート」
- 最初のダンジョンを選択（音量を調整） • Python（デフォルト）を選び「次へ」
- アイテムをダブルクリックで装備 → 「レベルスタート」

ステージC コード編集と実行

- 編集画面でコードを書く
- 「実行」をクリックしてキャラクターの動きを確認
- 困ったら「ヒント」「メソッド」「ミッション」を活用

```
hero.moveRight()  右へ移動  
hero.moveDown()  下へ移動
```

英語表示があっても翻訳未完のため設定の不備ではない

ステージD クリアして次へ

- 「目標：成功！」になるまで編集と実行を繰り返す
- すべての目標達成を目指す
- 成功したら「完了」→「続ける」で次のレベルへ

課金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

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 「Start Playing」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



教育者 ▾ 親 Play ▾

ログイン

アカウント登録

日本語

楽しみながら、コーディングと
AIの使い方を学ぼう！

今すぐサインアップして、無料でゲーム
やリソースを体験しよう。

私は教育者です

I'm a Parent

Start Playing

```
1 while true do
2     pet:moveTo("slime")
3     pet:spin()
4
5
6
```

SAVE

クラスコードなし、今すぐプレイ



「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:

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](#)

クラスコードは使わない

③ KITHGARD DUNGEON (キースガードのダンジョン) を選ぶ. 「ゲームスタート」をクリック



音が出るので, このとき, 各自で「音量」を調整

CODECOMBAT

教育者 ▾ 親 Play ▾ ログイン アカウン

KITHGARD DUNGEON
0/44

ゲームスタート

Escape the dungeon and level up your coding skills!

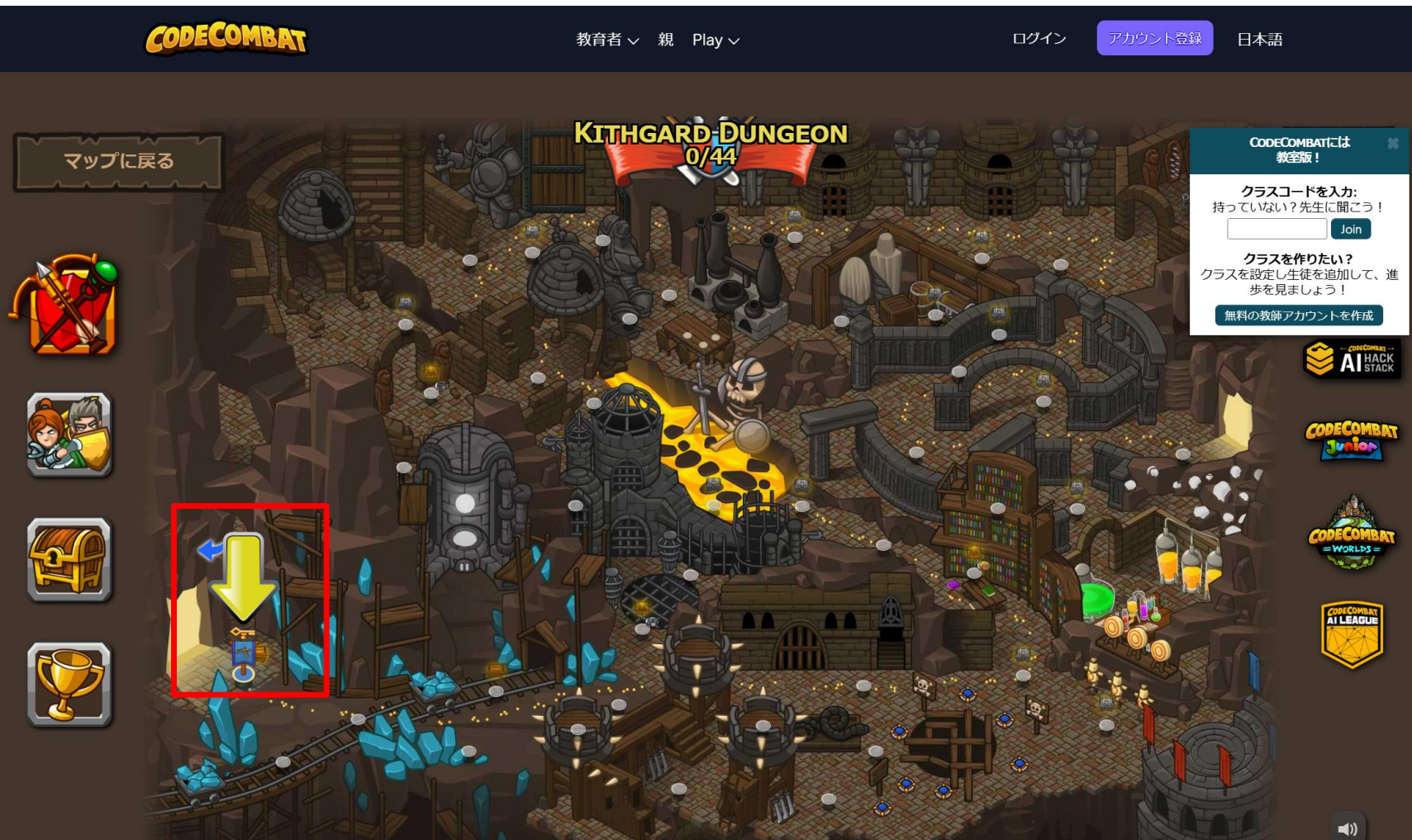
ゲーム開発 0/22
ロック
自分でレベルを造る事を習おう!

ウェブ開発 0/12
ロック
HTMLをもっと学ぼう!

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



音が出るので、このとき、各自で「音量」を調整



最初のダンジョン

⑤ 「ゲームスタート」をクリック



Database Lab.

CODECOMBAT

教育者 ▾ 親 Play ▾

ログイン

アカウント登録

日本語

マップに戻る

KITHGARD DUNGEON
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](#)



地下牢のダンジョン

ダイヤを取ってダンジョンを脱出しよう—他のものにはぶつからないように。このレベルでは、ヒーローの基本的な動きを学ぼう。

Basic Syntax

ゲームスタート



⑥ 「Python (デフォルト)」を選び、「次へ」をクリック



CODECOMBAT

マップに戻る

ヒーロー選択

0 ★ 1

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くて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	

Code Format: TEXT CODE

プログラミング言語: PYTHON (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



CODE COMBAT

インベントリー

0

シンプルなお靴

ロック

追加されるスキル

moveDown: heroが下(南)にちょっと歩く。
moveLeft: heroが左(西)にちょっと歩く。
moveRight: heroが右(東)にちょっと歩く。
moveUp: heroが上(北)にちょっと歩く。

ゲームスタート

ヒーローやプログラミング言語を変える

装備を外す

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

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

⑨ 「レベルスタート」をクリック



目標

トゲを避けよう。
ダイヤを集めよう。



コードでプログラムを書いてヒーローを導こう！

右のエディターにコードを書いて、準備ができれば「実行」をクリックしよう。ヒーローがそれを読み取り、指示に従うよ。

壁のトゲに触れないように、ヒーローを廊下の先まで進めよう。

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 a game interface for a dungeon. At the top, there are navigation buttons: "マップ" (Map), "ASK II", "ヒント" (Hint), "アカウント登録" (Account Registration), and "ゲームメニュー" (Game Menu). The "ヒント" button is highlighted with a red box. On the right side, a hint panel is open, displaying a list of instructions in Japanese and a code snippet: `hero.moveRight()`. At the bottom, a blue instruction box says "6行目に `moveDown` コマンドを入力しよう。" (Enter the `moveDown` command on the 6th line.) with "スキップ (esc)" and "続ける" (Continue) options. The bottom right corner shows a list of movement commands: `hero.moveDown(steps)`, `hero.moveLeft(steps)`, `hero.moveRight(steps)`, and `hero.moveUp(steps)`.

但し、英語で表示される場合がありなる。
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。47

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 地下牢のダンジョン ASK AI ヒント アカウント登録 ゲームメニュー

トゲを避けよう。
ダイヤを集めよう。
目標: 未完成

- 1 # ダイヤに向かって進もう。
- 2 # トゲに触れないように！
- 3 # 下にコードを入力して、終わったら「実行」ボタンをクリックしよう。
- 4
- 5 `hero.moveRight()`
- 6

実行

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

ANYA 11

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 with a character in a dungeon. The character is positioned at the bottom of a red-bordered area. The code editor on the right shows the following code:

```
1 # ダイヤに向かって進もう。  
2 # トゲに触れないように！  
3 # 下にコードを入力して、終わったら「実行」  
   ボタンをクリックしよう。  
4  
5 hero.moveRight()  
6 hero.moveDown()  
7  
8
```

The code editor has a button labeled "実行" (Execute) at the bottom. Below the code editor, there is a list of movement methods:

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

The game interface includes a map button, a quest log, and a character status bar at the bottom.

迷ったら、「ミッション（目標）」や「ヒント」を確認する

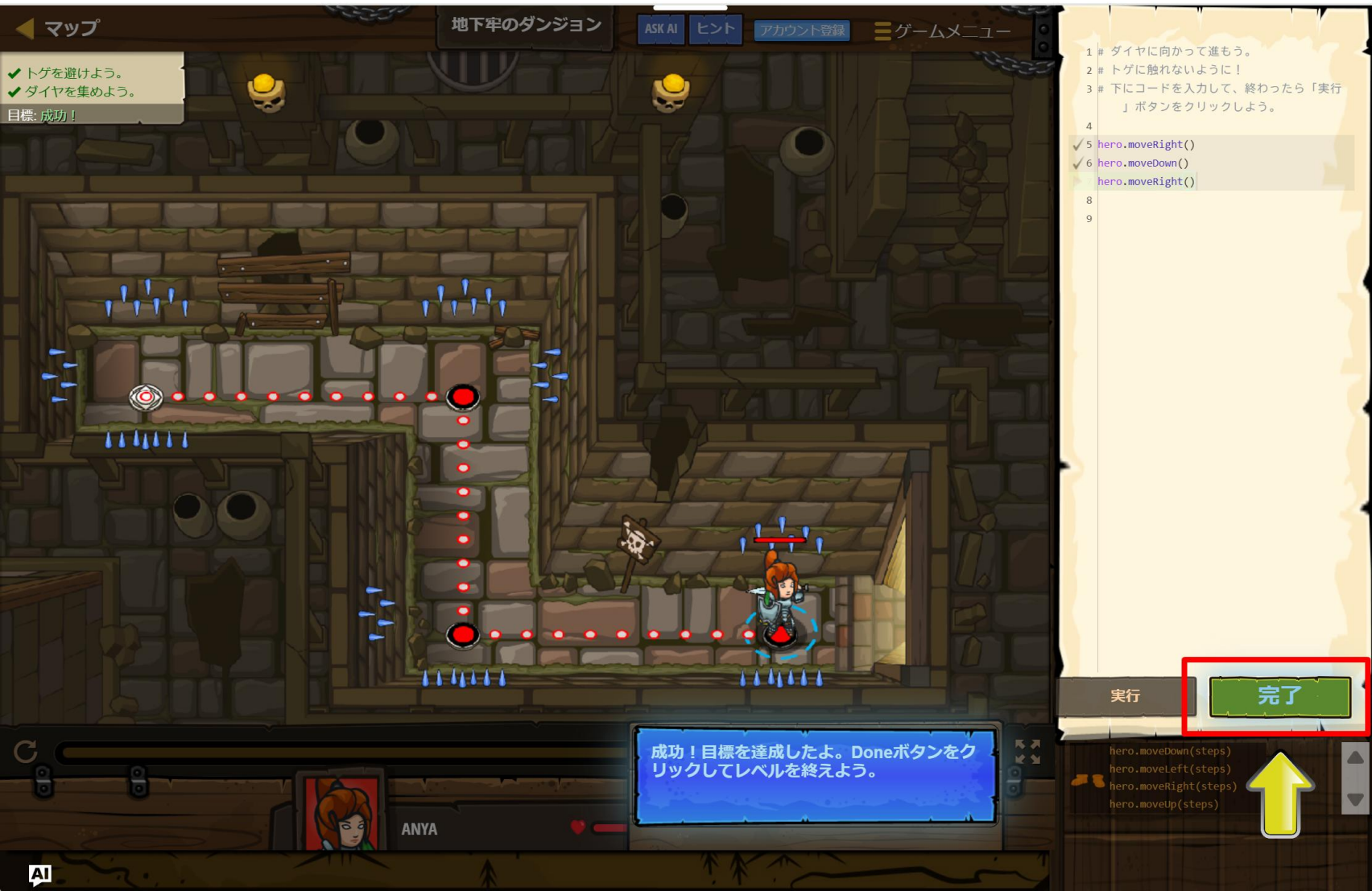
- プログラミングの練習だけでなく、**ゲーム要素（パズル）**もある。楽しんで解く



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



The screenshot shows a game editor interface. On the left, a character named ANYA is in a dungeon environment. A red dotted line indicates a path from the character to a target (a red circle). The top of the screen has a menu with options like 'マップ', '地下牢のダンジョン', 'ASK AI', 'ヒント', 'アカウント登録', and 'ゲームメニュー'. A notification box in the top left says '目標：成功！'. On the right, a code editor shows a list of commands: 1 # ダイヤに向かって進もう。 2 # トゲに触れないように！ 3 # 下にコードを入力して、終わったら「実行」ボタンをクリックしよう。 4 5 hero.moveRight() 6 hero.moveDown() 7 hero.moveRight() 8 9. A '実行' (Execute) button is visible, and a '完了' (Done) button is highlighted with a red box. A yellow arrow points to the '完了' button. At the bottom, a blue message box says '成功！目標を達成したよ。Doneボタンをクリックしてレベルを終えよう。'. The bottom right corner has a yellow arrow pointing up and a '52' page number.

```
1 # ダイヤに向かって進もう。
2 # トゲに触れないように！
3 # 下にコードを入力して、終わったら「実行」ボタンをクリックしよう。
4
5 hero.moveRight()
6 hero.moveDown()
7 hero.moveRight()
8
9
```

実行 完了

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

成功！目標を達成したよ。Doneボタンをクリックしてレベルを終えよう。

ANYA

52

⑬ 完了の確認. 「続ける」をクリック

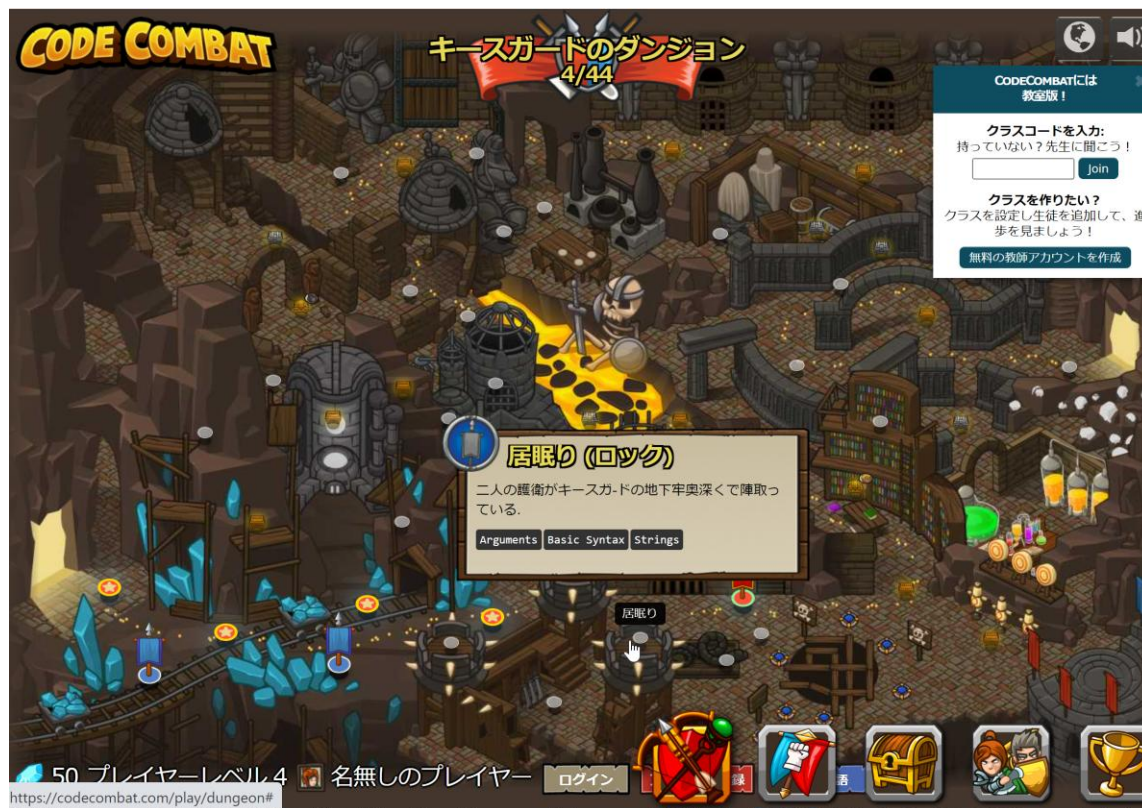


続けてみる

- 青い旗をたどる。4つは無料
- 旗の中には有料のものもある（ロックされている）



青い旗をたどる



青い旗でも「プラチナを購入」は
避けて別のものへ。無料範囲で学
習目標は達成できる

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



CodeCombatの注意点：3つの利用パターン

本授業の範囲

無料・クラスコード無し

Pythonの最初の4レベル

- 登録不要、すぐ遊べる
- 本授業はこの範囲

無料・クラスコードあり

- 演習内容が違うものになる
- 名前・メールアドレス等の登録が必要

(本授業では使わない)

有料 (会員登録)

- 500以上のレベルが解放
- 興味のある人のみ
- 各自の判断で

(本授業では使わない)

本授業では、会員登録しない、クラスコードも使わない