

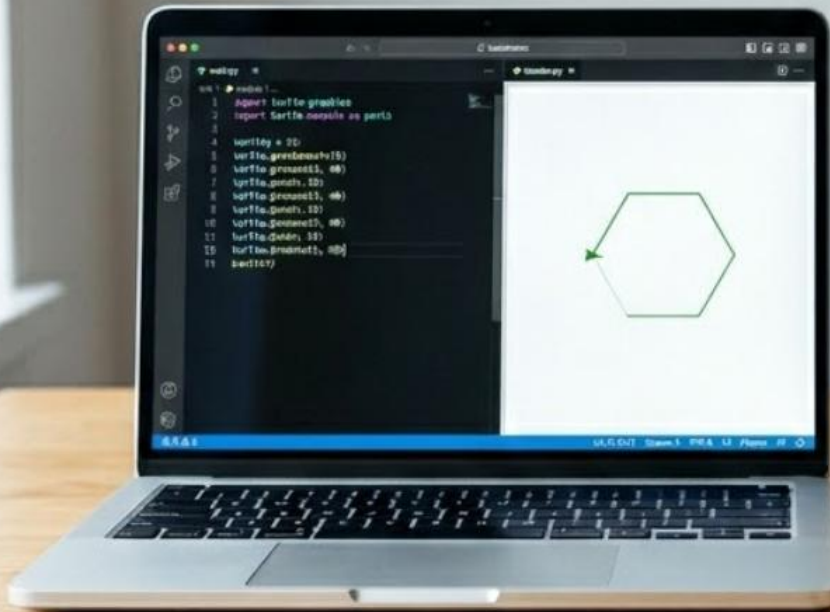
Python の基本：オブジェクト、メソッド、変数

(プログラミング基礎・用語集・体験)

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

金子邦彦





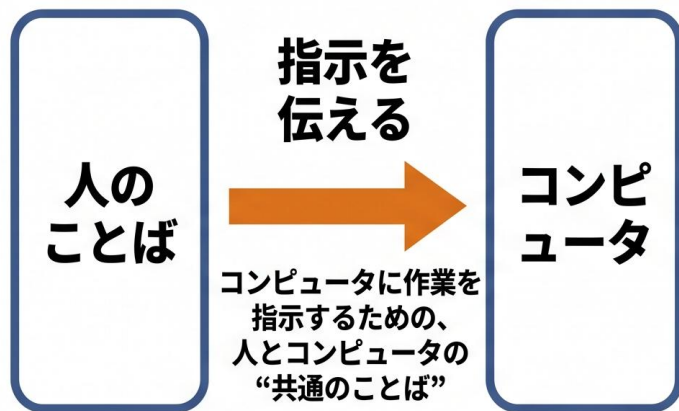
プログラミング思考を支える3つの道具：①オブジェクト・メソッド・引数、
②変数と代入、③アルゴリズムの感覚

本授業の図は生成AIである Google Nano Banana 2 を用いて作成

Python (パイソン) 言語



プログラミング言語とは



その言語の一つが Python

プログラムとは

指示を並べた“手順書”
がプログラム



Python の特徴

読みやすい

— 英語の文章に近く、
見て意味が
分かりやすい

書きやすい

— 少ない記述で
動かせる

応用が広い

— 簡単な処理から
高度な開発まで対応

シンプルな用途から高度なプログラムまで、はば広く使える

プログラミング思考を支える3つの道具

複雑な問題を解くための土台となる基本概念

①

**オブジェクト・
メソッド・引数**

操作する対象と、
その動作、そして動
作に渡す追加情報

②

変数と代入

データに名前を付
けて保管し、必要に
応じて書き換える

③

アルゴリズムの感覚

問題を解く手順を
順序立てて組み立て
る直感

処理を表す三点セット

turtle.goto(0, 100)

オブジェクト

処理の対象を
表すもの

メソッド

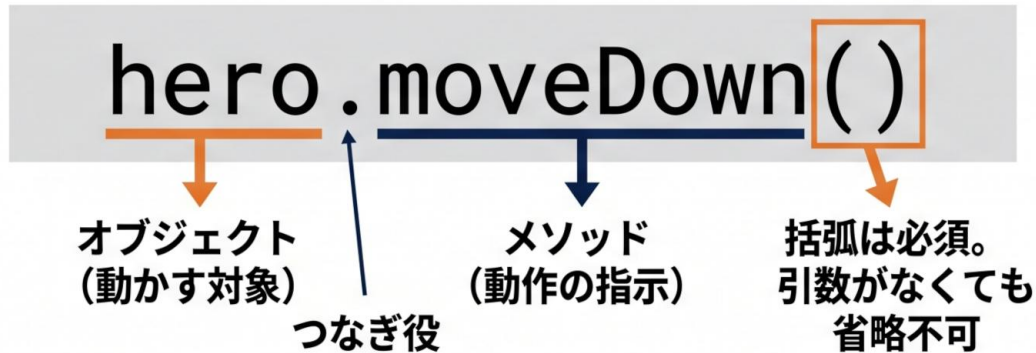
オブジェクトの
能力を表すもの

引数

能力を細かく
指定するもの

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

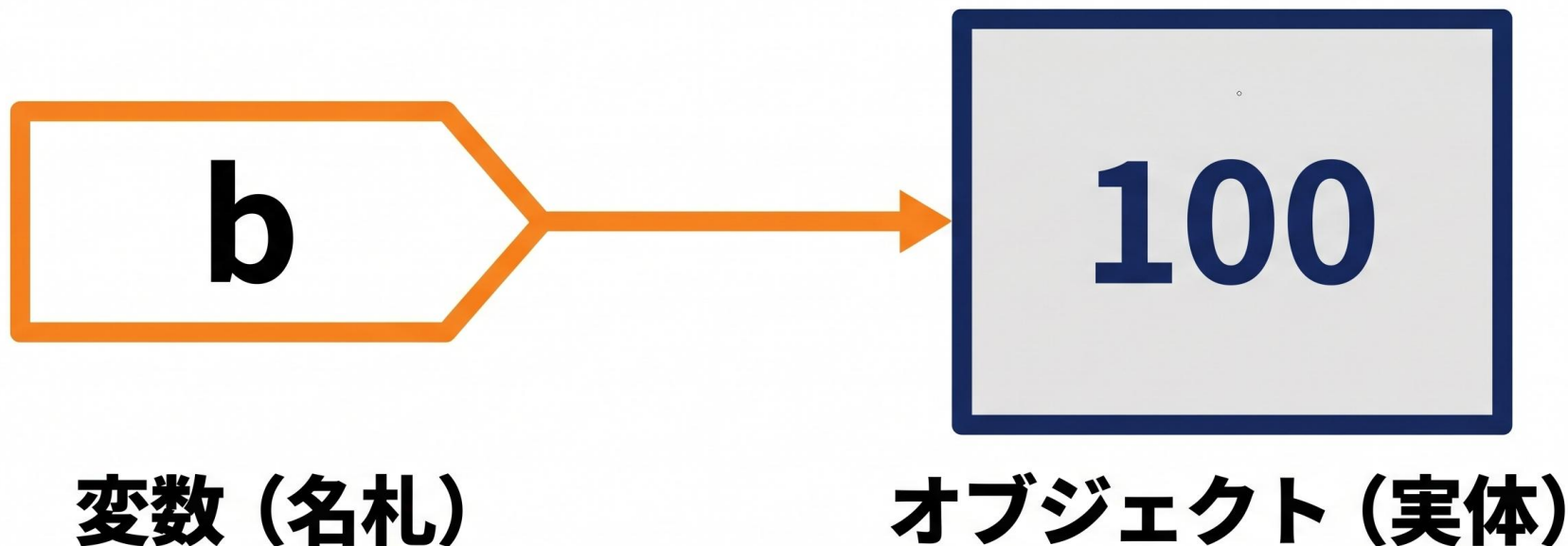
プログラムは上から順に実行

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

書いた順番どおりに動作する



変数は『名札』、オブジェクトは『実体』。名札が実体を指し示す。



① データに名前を付けて保管する

② 変数 (名札) は実体を指し示すだけ

③ 必要に応じて書き換えられる

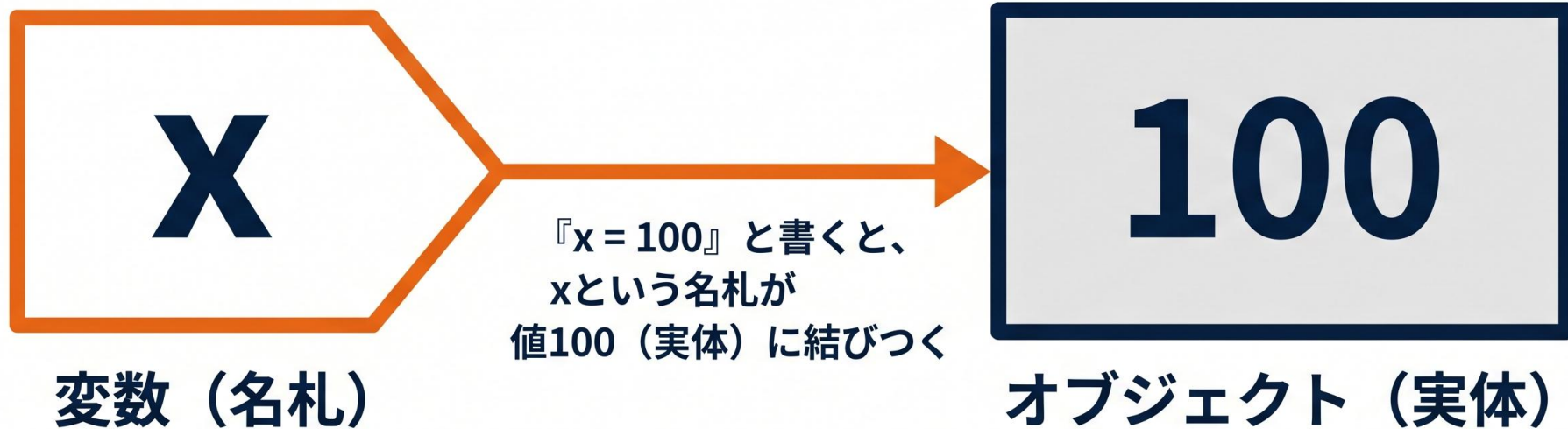


変数と代入



代入とは、名札（変数）に実体（値）を結びつけること

$$x = 100$$



再代入



再代入：今の値をもとに計算した結果を、同じ変数にもう一度入れ直す



今の値

$$x + 100$$

(入れ直す)



入れ直した値

今のxの値に100を足す



今の値

$$\text{score} + 1$$

(入れ直す)



入れ直した値

今のscoreに1を足す

代入文の構文ルール



代入文は『左辺=変数名、右辺=式』の順で書く

正しい (○)

x = x * 2

左辺：変数名 代入 右辺：式

誤り (×)

2 * x = x

左辺が式になっている → 不可

左辺は必ず変数名でなければならない。右辺の計算結果を左辺の変数に入れる。

アルゴリズムの感覚：手順を順序立てて組み立てる



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

```
t.forward(50)  
t.left(60)  
t.forward(50)  
t.left(60)  
t.forward(50)  
t.left(60)  
t.forward(50)  
t.left(60)  
t.forward(50)  
t.left(60)  
t.forward(50)  
t.left(60)
```

① やりたいこと（ゴール）



正六角形を描きたい

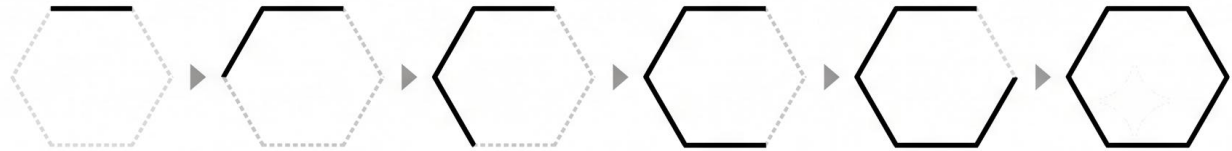
② 小さな手順に分解

前進する（50すすむ）

左に曲がる（60度）

この2つを6回くりかえす

③ 順番に実行すると形になる



六角形の辺が1本ずつ増えて完成する