3. Python入門：プログラミングの基礎と創造的学習への展開

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

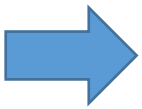
3-1. Python プログラミングの基礎

Python プログラミングの基礎の全体像

- **プログラミング**は，**プログラム**を作成する**創造的な活動**
- **プログラム**は，**コンピュータ**に指示を出し，所定の作業を遂行させる
- **Python** は，**読みやすさ**，**書きやすさ**，**幅広い応用範囲**が特徴

```
a = [200, 400, 300]
for i in a:
    print (i * 1.08)
```

Python 言語のプログラム



```
216.0
432.0
324.0
```

プログラムの実行結果

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**である．**値を保存**し，後から**参照**できる仕組み．

- **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**される操作

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.
- **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

例　　**t.goto(0,100)**

引数は 0 と 100

Python プログラムの書き方（コーディングスタイル）

- **インデント（字下げ）によるブロックの区切り**
  （通常「タブ」または「４つの半角スペース」）
- **コメント**：行の先頭に「#」．プログラムの説明を記述
- **空白行**：プログラムを読みやすくするために挿入可能

例

```
import turtle

for i in range(4):
    turtle.forward(100)
    turtle.right(90)

turtle.done()
```

空白行

インデント（字下げ）

空白行

3-2. 変数，代入

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**である．**値を保存**し，後から**参照**できる仕組み．

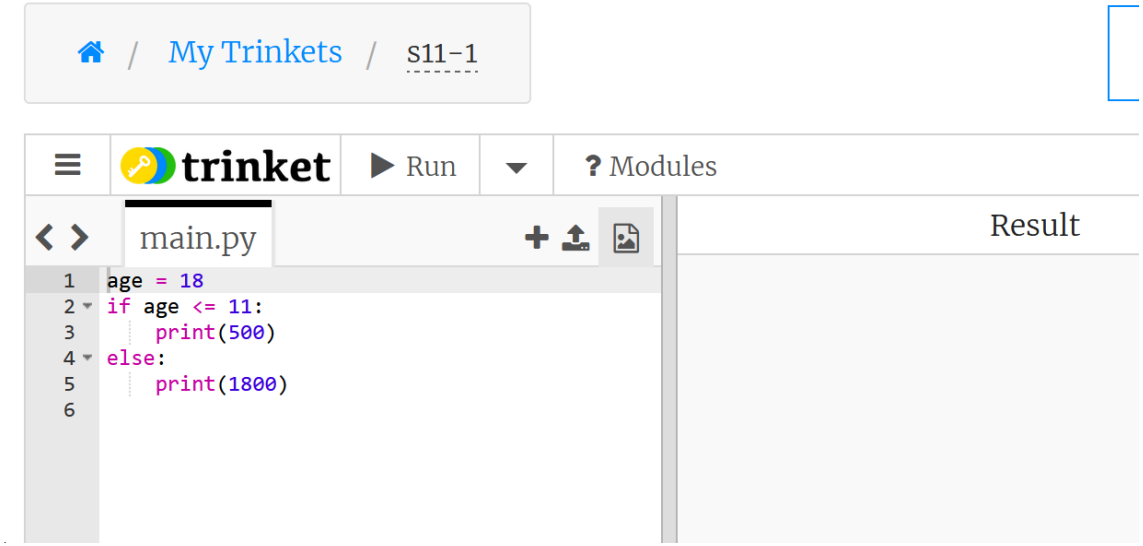
- **代入**：「**x = 100**」のように書くことで，**x という名前の変数に、値 100 が保存**される操作

例　**x = 100**

  - 保存された値は，プログラムの中で何度でも参照することが可能
  - 別の値で上書きすることも可能．

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



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

- **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．変数と代入

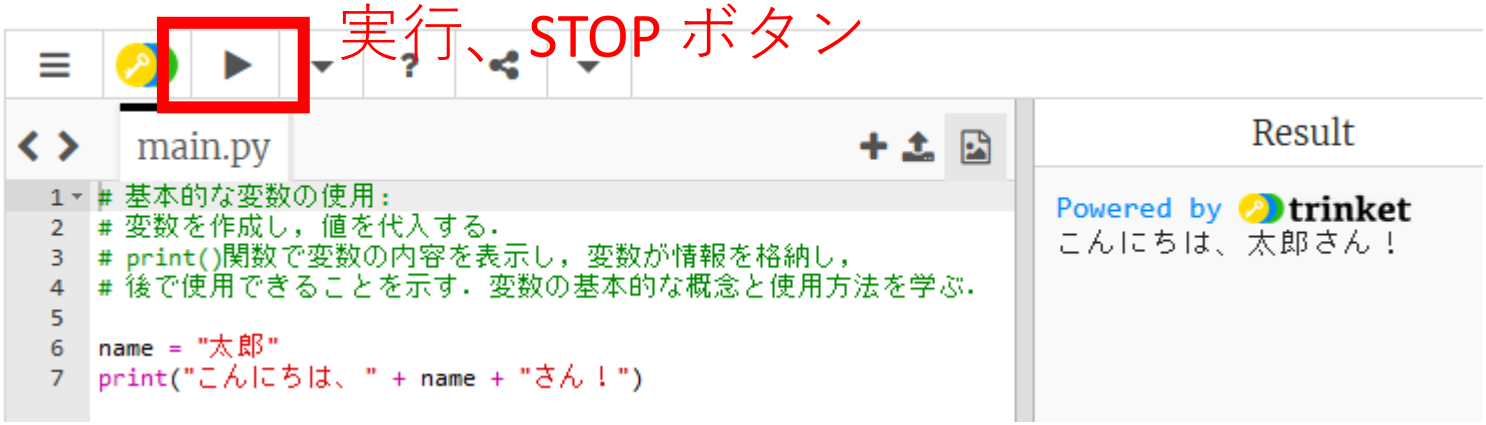
１．基本的な変数の使用

① trinket の次のページを開く

https://trinket.io/python/abafd851480a

- 変数
- 値を代入方法
- print() 関数を使って変数の内容を表示する方法

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**
- プログラムを**書き替えて再度実行**することも可能

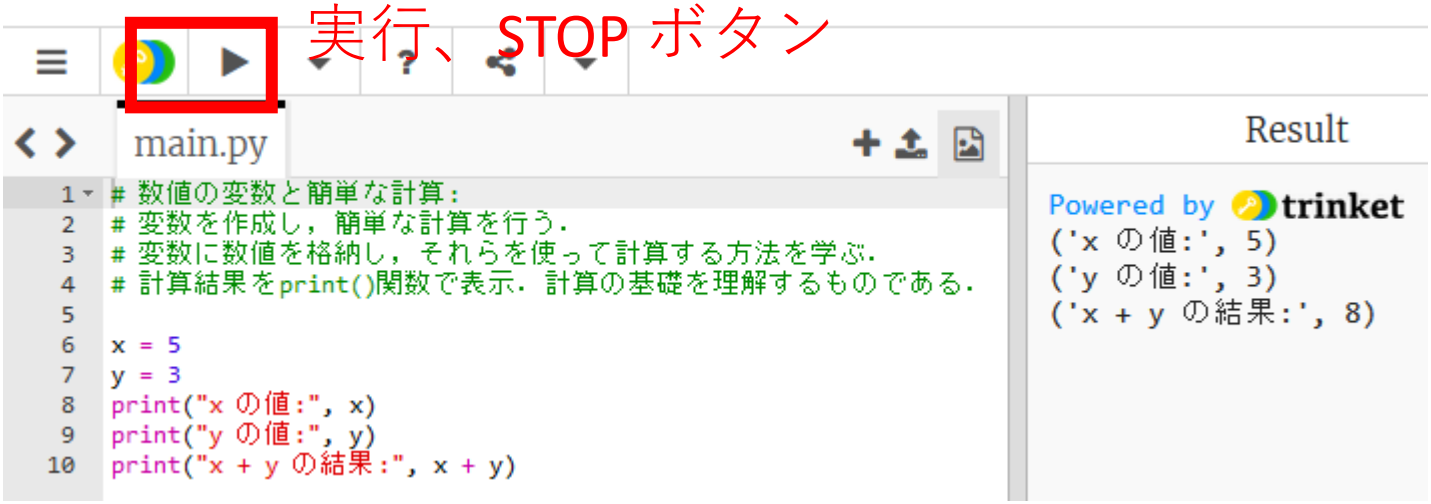
2．基本的な変数の使用

① trinket の次のページを開く

https://trinket.io/python/9870e86d63b9

- 複数の変数を使って簡単な計算を行う
- print() 関数を使って変数の内容を表示する

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

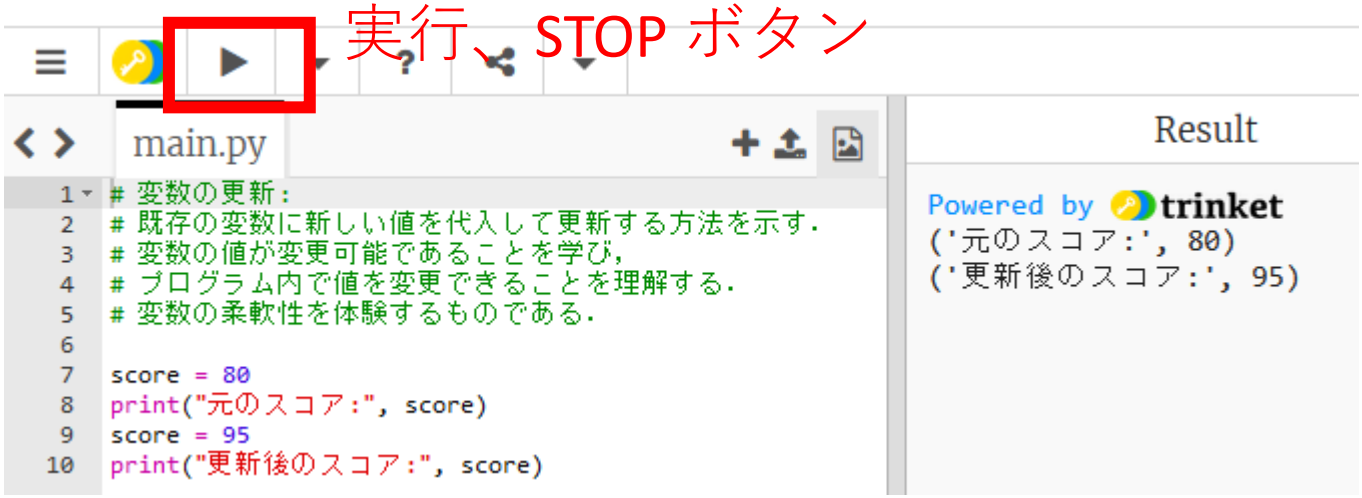
３．変数の更新

① trinket の次のページを開く

https://trinket.io/python/b869619b0874

- 既存の変数に新しい値を代入して更新
- プログラムの中で変数の値が変化

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**
- を**書き替えて再度実行**することも可能

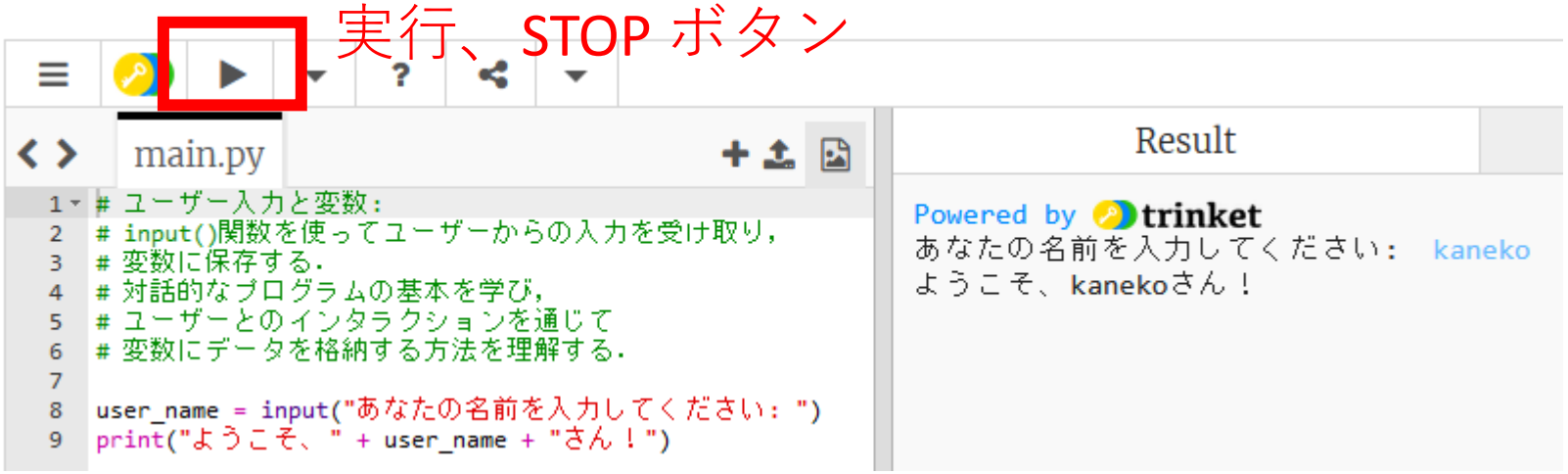
４．ユーザー入力と変数

① trinket の次のページを開く

https://trinket.io/python/45f0bed92360

- input() 関数を使ってユーザーからの入力を受け取る
- 入力された値を変数に保存し，プログラム内で利用
- 対話的なプログラムの基本

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**
- プログラムを**書き替えて再度実行**することも可能

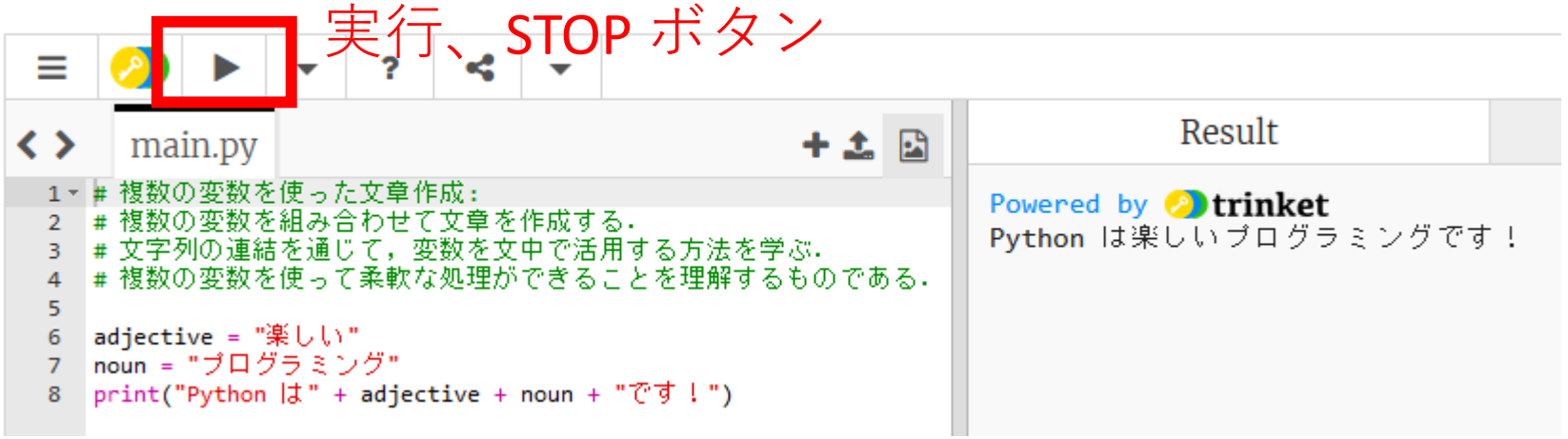
５．複数の変数

① trinket の次のページを開く

https://trinket.io/python/abafd851480a

- 異なる値を別々の変数に保存
- 複数の変数を利用して計算や表示を行う

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**
- プログラムを**書き替えて再度実行**することも可能

まとめ

- 変数と代入は，プログラミングにおける基本的なデータの保存と参照の仕組み
- 変数の使用方法を理解することで，プログラミング学習の基礎を確立できる

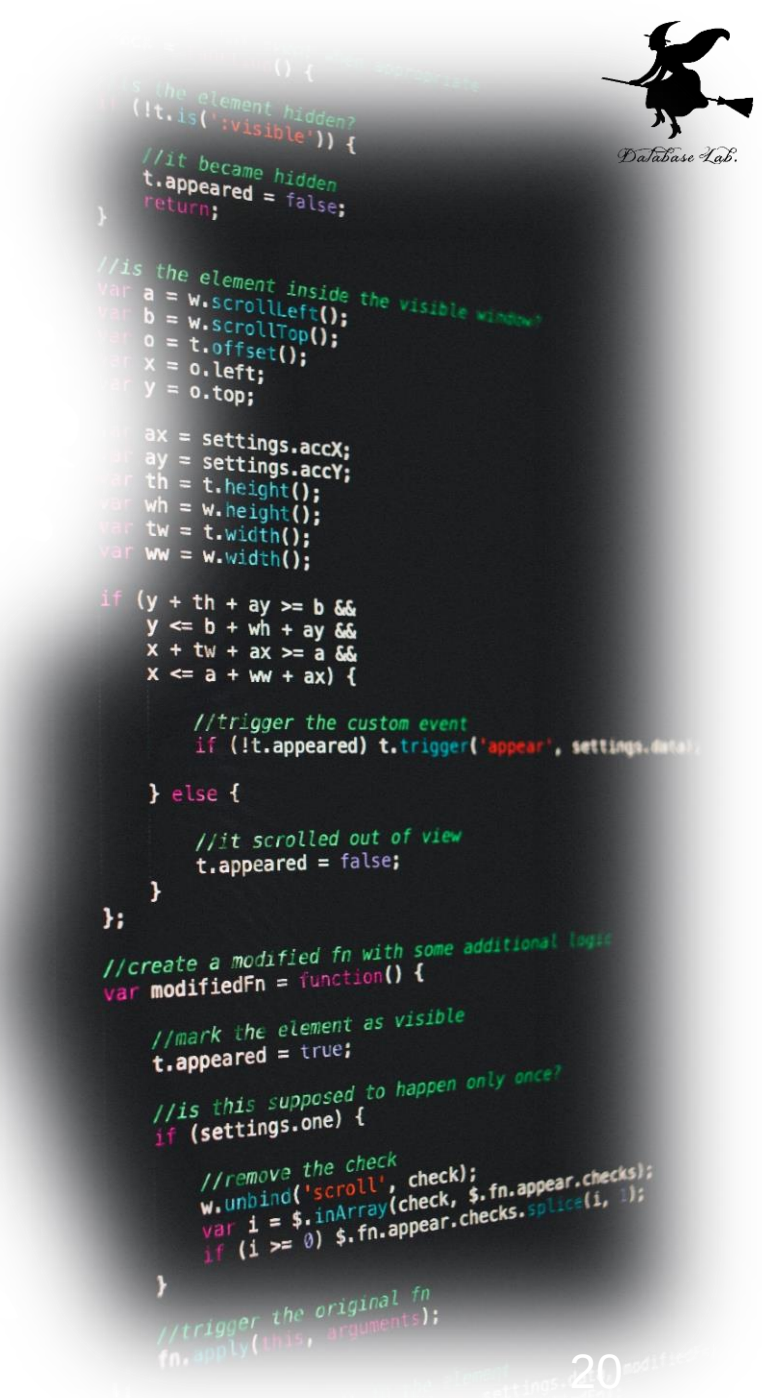
3-3. プログラミングの楽しさと達成感

プログラミング学習の意義と目標

- **論理的思考力や課題解決力の向上**
- **プログラム**の**作成と実行**を通じて，**コンピュータ**の動作原理を学び，**問題解決能力**を高める
- **自分の作ったプログラム**が動作する**喜び**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**
- 人間が**読み書き**，**編集**できる
- **プログラムの動作を理解**し，必要に応じて**改変**するための基礎になる



演習でプログラミングを学ぶ意義，成果

① プログラミングの基本概念の理解

② 創造的なプログラム作成の態度．習慣

③ 問題解決能力

④ 自主的な学習態度と探求心

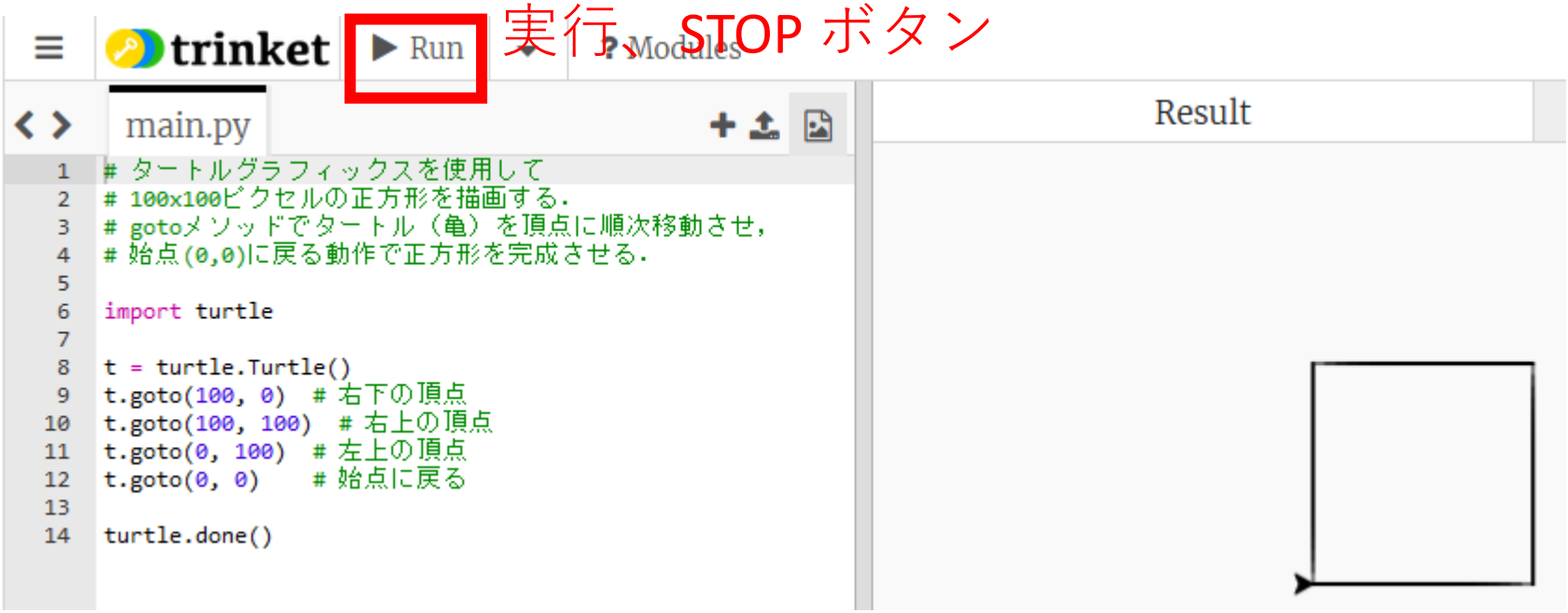
演習
図形のバリエーション

① プログラムを試す．trinket の次のページを開く

https://trinket.io/python/035810ce8d49

・goto の組み合わせによって，独自の図形を描く

② 実行結果が，次のように表示されることを確認



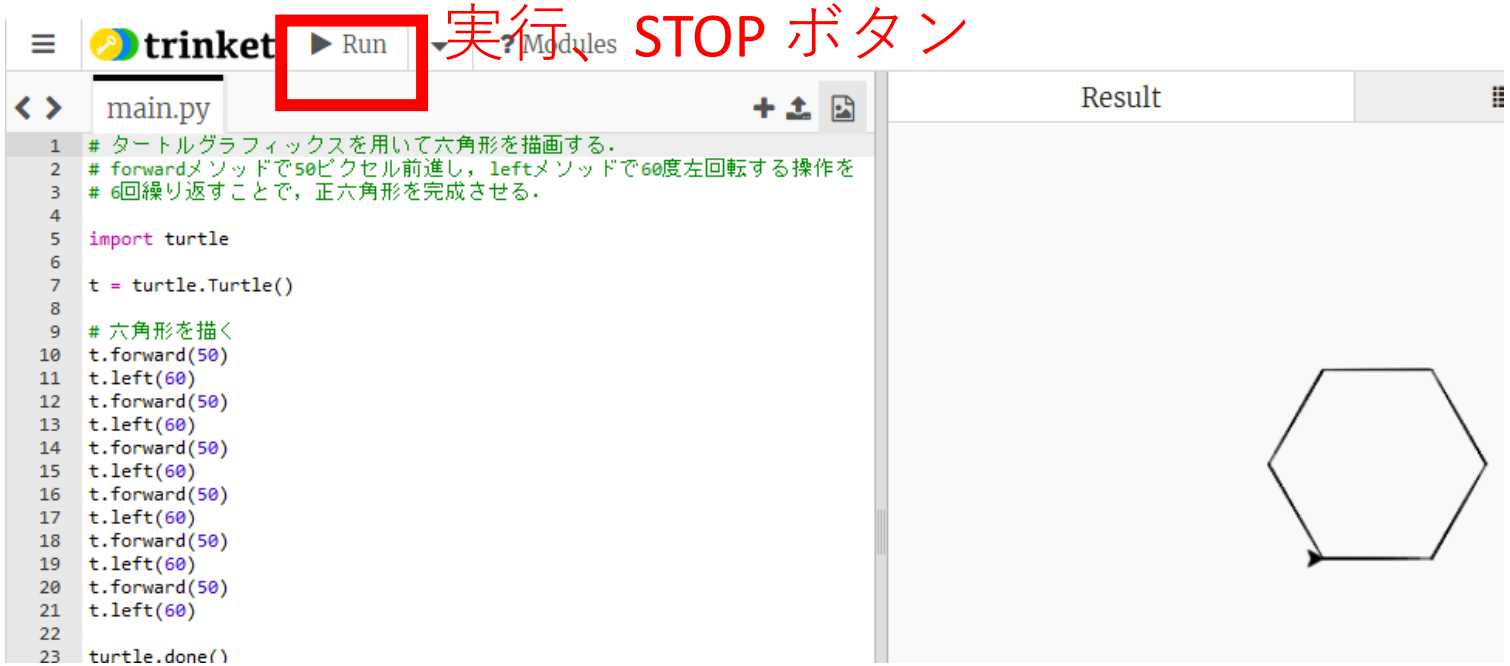
- 実行が開始しないときは、「**実行ボタン**」で**実行**
- プログラムを**書き替えて再度実行**することも可能

③ プログラムを試す．trinket の次のページを開く

https://trinket.io/python/ddb861147133

・forward, left のような新しいメソッドを探求

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**
- プログラムを**書き替えて再度実行**することも可能

自己主導型学習の４つの観点

① 具体的トピックスでの学習

Pythonでグラフィックスを描く．基本的な図形描画の技術を習得

② Pythonの基本を押さえる

オブジェクト，メソッド，引数の概念を理解

③ 発想力，創造力を育む

独自のデザインし実装する

④ 自主性，自己研鑽力を高める

新しい機能を自ら調べ，理解し，試す

演習
プログラミングの楽しさと達成感

オブジェクトの生成，座標指定による移動などの基本操作を学ぶ

① https://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)
```

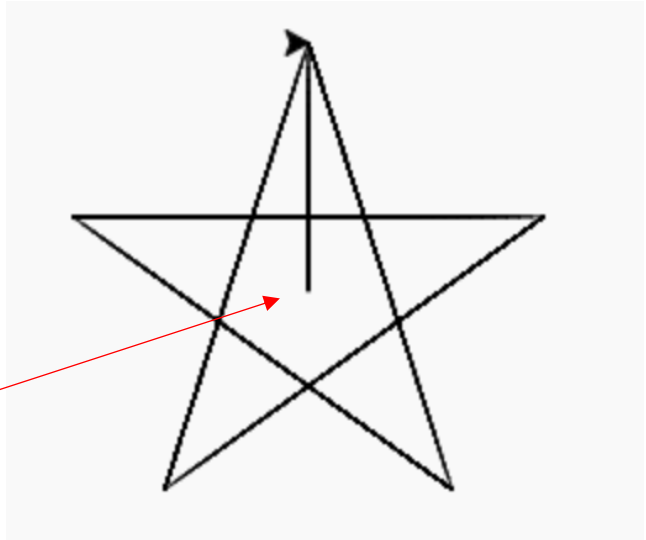
ライブラリのインポート

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)



色の指定方法，繰り返し処理，円の描画方法など，
より発展的なプログラミング

②　https://trinket.io/python/f8cd554693

```
import turtle

t = turtle.Turtle()

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

for i in range(3):

    t.color(colors[i])

    t.circle(30)

    t.forward(50)
```

ライブラリのインポート

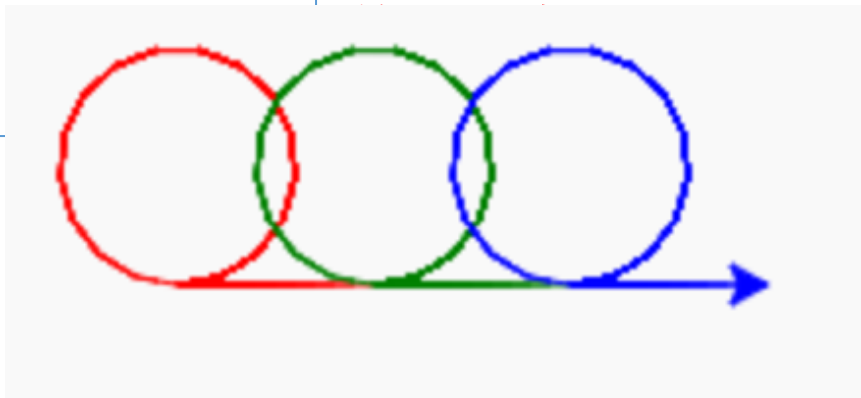
オブジェクト生成。t へのセット。

色は、赤、緑、青

色を変える

半径３０の円

実行結果



プログラミングを学ぶ意義と達成感

①卒業後は，ITエンジニアとして活躍しよう

②**基本的なプログラミングの知識**：変数と式，オブジェクトの生成，メソッドの使用をおさえておくこと

③プログラミングの世界においても，**失敗やエラーは成長のチャンス**になる.

④楽しみながらプログラミングに取り組もう．プログラミングは創造的な作業であり，自分のアイデアを形にすることができるもの.

⑤将来は，**自分が興味を持つテーマやプロジェクト**を見つけ，プログラミングによる作品作りに**挑戦**してみよう．行動や体験が自信と成長につながる.