

pf-4. 変数, 代入, 入力と出力

(Python 入門)

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

金子邦彦





① プログラミングの基本的な概念

② コンピュータにおける入力と出力の重要性

4-1. 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.

式の実行結果



式の実行結果として、値が得られる

A screenshot of a code editor interface. The top bar contains icons for a menu, a key, a play button, a dropdown arrow, and a question mark. Below the top bar, the editor shows a file named "main.py" with a single line of code: "1 print(100 * 200)". To the right of the code editor, the execution result is displayed as "Powered by 20000".

```
main.py  
1 print(100 * 200)
```

Powered by 20000

プログラム

実行結果

変数への代入



プログラムで、「**x = 100**」のように書くと、**x**
の値が 100 に変化する

A screenshot of a code editor interface. The top toolbar contains icons for a menu, a key, a play button, a dropdown arrow, and a question mark. Below the toolbar, the file name "main.py" is displayed. The code editor shows two lines of Python code: "1 x = 100" and "2 print('x = ', x)". To the right of the code editor is a results pane labeled "Result" which displays the output: "Powered by [key icon] tri" and "('x = ', 100)".

```
main.py  
1 x = 100  
2 print('x = ', x)  
Result  
Powered by [key icon] tri  
('x = ', 100)
```

プログラム

実行結果

式の中に変数を含めることができる

A screenshot of a code editor interface. The editor title bar shows "main.py" with navigation arrows on the left and icons for adding, uploading, and inserting files on the right. The code area contains three lines of Python code: "1 x = 100", "2 y = 200", and "3 print(x + y)". To the right of the code editor, a vertical panel displays the execution result: "Powerrec" in blue text and "300" in black text below it.

```
main.py  
1 x = 100  
2 y = 200  
3 print(x + y)  
Powerrec  
300
```

プログラム

実行結果

式と変数の Python プログラム



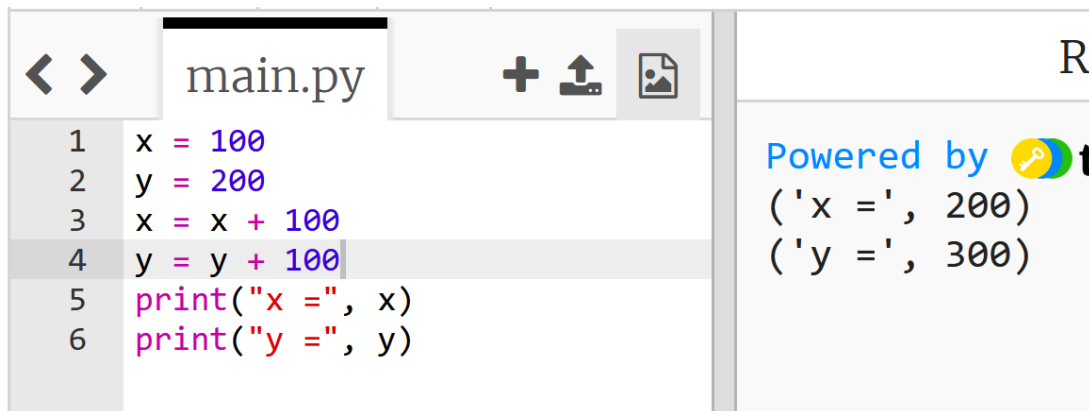
変数 x, y に 100 を足して表示

```
x = 100
y = 200
x = x + 100
y = y + 100
print("x =", x)
print("y =", y)
```

x に 100 を足す

y に 100 を足す

print は、メッセージ（文字列）や変数の値を表示



```
< > main.py + ⬆️ 📄 Re
1 x = 100
2 y = 200
3 x = x + 100
4 y = y + 100
5 print("x =", x)
6 print("y =", y)

Powered by Python
('x =', 200)
('y =', 300)
```

実行結果

- 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

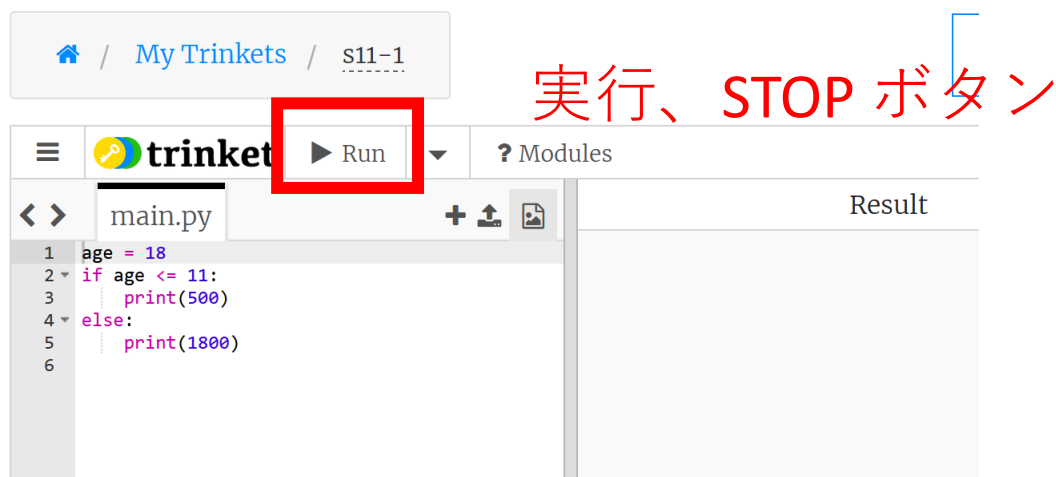


trinket でのプログラム実行

- 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/0fd59392c8>

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

演習 変数 x, y を使って計算

ページ 11 ~ 15

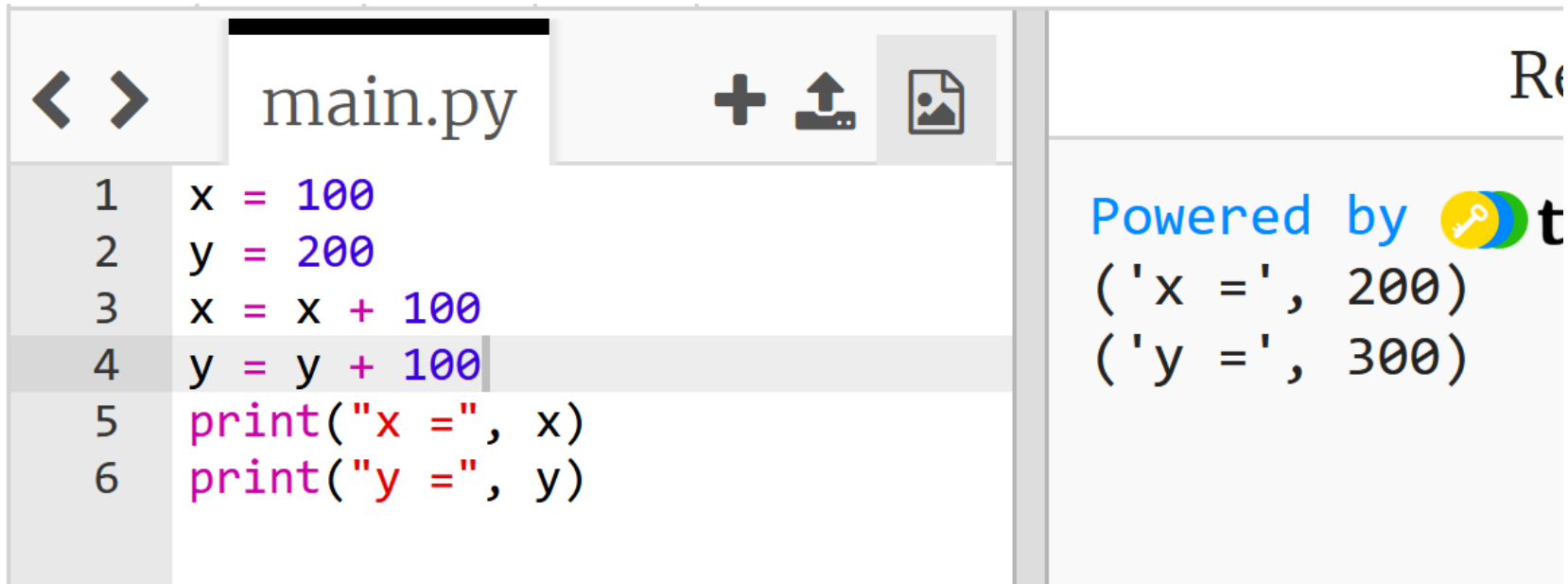
【トピックス】

- trinket の利用
- 式と変数
- 確認クイズに自主的に挑戦

① trinket の次のページを開く

<https://trinket.io/library/trinkets/1e414fec80>

② 実行結果が、次のように表示されることを確認



```
< > main.py + ⬆️ 📄  
1 x = 100  
2 y = 200  
3 x = x + 100  
4 y = y + 100  
5 print("x =", x)  
6 print("y =", y)  
Re  
Powered by 🔑 t  
( 'x =', 200 )  
( 'y =', 300 )
```

③ 確認クイズ

x, y を **2倍する** プログラムはどちらが正しいですか？

なお、掛け算には「*」を使う。

正解だと思う方を、自分でコンピュータに入れて、正しいかを確認してください

```
x = 100
y = 200
x = x * 2
y = y * 2
print("x =", x)
print("y =", y)
```

```
x = 100
y = 200
2 * x = x
2 * y = y
print("x =", x)
print("y =", y)
```

④ 確認クイズ

x, y の値を**半分にする**プログラムはどちらが正しいですか？

なお、割り算には「/」を使う。

正解だと思う方を、自分でコンピュータに入れて、正しいか
を確かめてください

```
x = 100
y = 200
x = 2
y = 2
print("x =", x)
print("y =", y)
```

```
x = 100
y = 200
x = x / 2
y = y / 2
print("x =", x)
print("y =", y)
```

⑤ 確認クイズ

x と y を足した値を新しい x の値にするプログラムはどちらが正しいですか？

正解だと思う方を，自分でコンピュータに入れて，正しいかを確認してください

```
x = 100
y = 200
x = x + y
print("x =", x)
print("y =", y)
```

```
x = 100
y = 200
x + y
print("x =", x)
print("y =", y)
```

答え合わせ



③

```
main.py  
1 x = 100  
2 y = 200  
3 x = x * 2  
4 y = y * 2  
5 print("x =", x)  
6 print("y =", y)
```

Powered by triton
('x =', 200)
('y =', 400)

④

```
main.py  
1 x = 100  
2 y = 200  
3 x = x / 2  
4 y = y / 2  
5 print("x =", x)  
6 print("y =", y)
```

Powered by triton
('x =', 50.0)
('y =', 100.0)

⑤

```
main.py  
1 x = 100  
2 y = 200  
3 x = x + y  
4 print("x =", x)  
5 print("y =", y)
```

Powered by triton
('x =', 300)
('y =', 200)

演習. 変数と代入

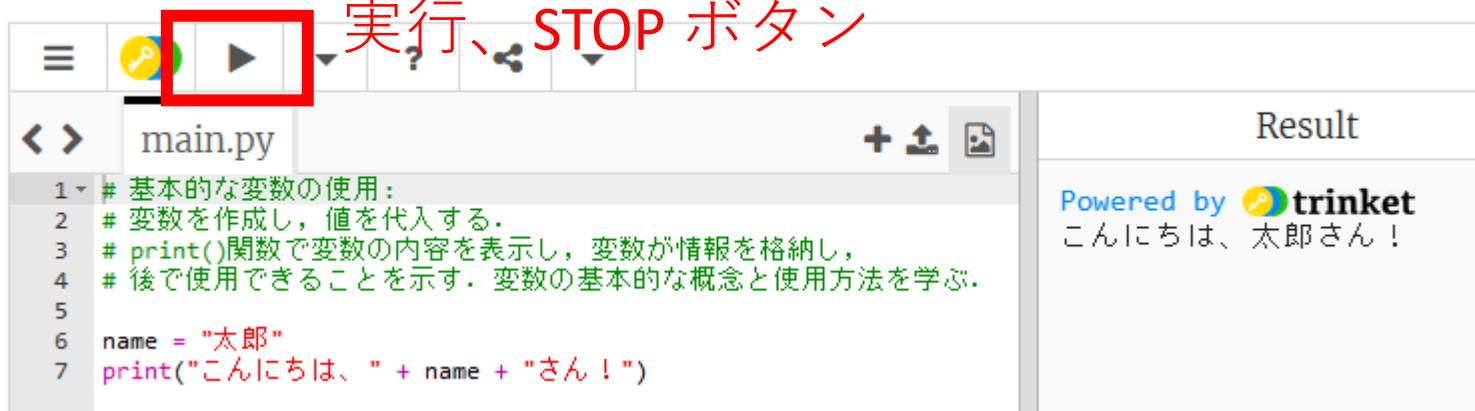
1. 基本的な変数の使用

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、STOP ボタン

```
1 # 基本的な変数の使用:
2 # 変数を作成し、値を代入する.
3 # print()関数で変数の内容を表示し、変数が情報を格納し、
4 # 後で使用できることを示す。変数の基本的な概念と使用方法を学ぶ。
5
6 name = "太郎"
7 print("こんにちは、" + name + "さん!")
```

Result

Powered by trinket
こんにちは、太郎さん！

- 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

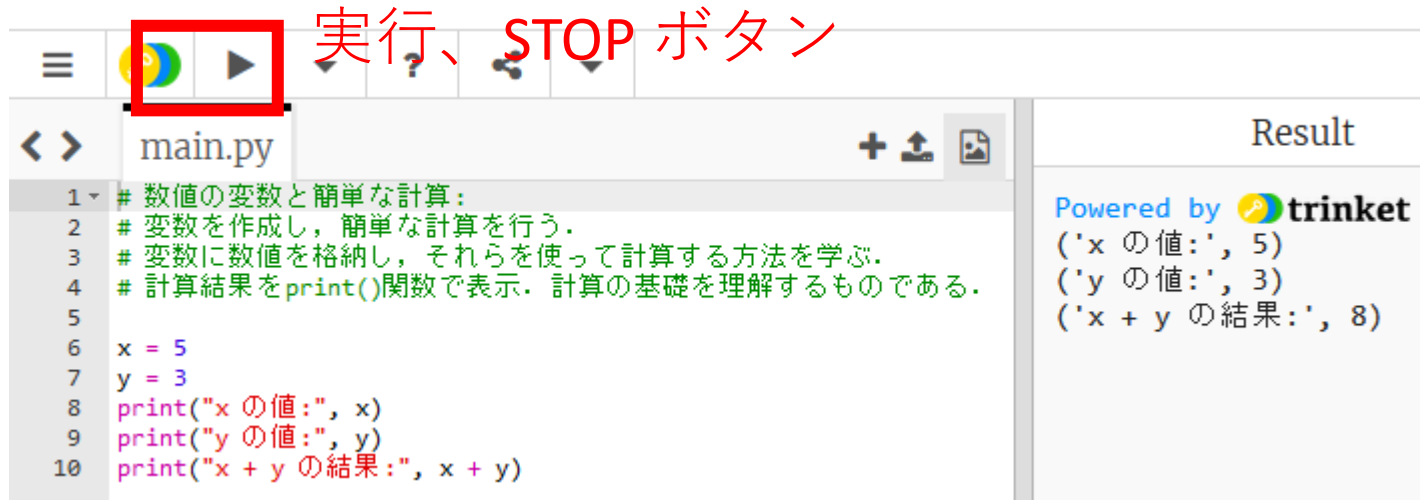
2. 基本的な変数の使用

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、STOP ボタン

```
1 # 数値の変数と簡単な計算:  
2 # 変数を作成し、簡単な計算を行う。  
3 # 変数に数値を格納し、それらを使って計算する方法を学ぶ。  
4 # 計算結果をprint()関数で表示。計算の基礎を理解するものである。  
5  
6 x = 5  
7 y = 3  
8 print("x の値:", x)  
9 print("y の値:", y)  
10 print("x + y の結果:", x + y)
```

Result

Powered by  trinket

('x の値:', 5)
('y の値:', 3)
('x + y の結果:', 8)

- 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

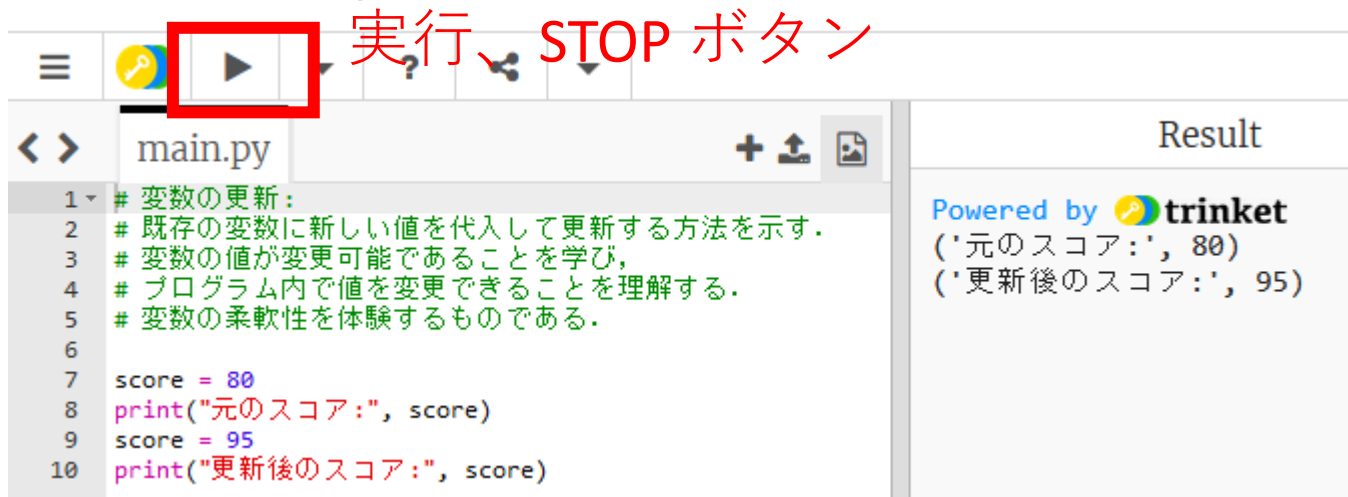
3. 変数の更新

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



The screenshot shows the Trinket IDE interface. At the top, there is a toolbar with a play button (run) and a stop button (stop), both of which are highlighted with a red box. The text "実行、STOP ボタン" (Execute, STOP button) is written in red above the buttons. Below the toolbar, the code editor shows a Python script named "main.py" with the following content:

```
1 # 変数の更新:  
2 # 既存の変数に新しい値を代入して更新する方法を示す。  
3 # 変数の値が変更可能であることを学び、  
4 # プログラム内で値を変更できることを理解する。  
5 # 変数の柔軟性を体験するものである。  
6  
7 score = 80  
8 print("元のスコア:", score)  
9 score = 95  
10 print("更新後のスコア:", score)
```

To the right of the code editor, the "Result" panel displays the output of the script:

```
Powered by trinket  
( '元のスコア:', 80)  
( '更新後のスコア:', 95)
```

- 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

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



4-2. 入力と出力

入力と出力



- **入力**は、他のコンピュータや人間などが、コンピュータにデータを入れる
 - **input** は、**キーボードから与えられたデータ（文字列）**を、**Enter** キーが押されるまで読み込む。
- **出力**は、コンピュータが、他のコンピュータや人間などにデータを出す
 - **print** は、**メッセージ（文字列）** や、**変数の値**の**表示**を行う

演習

input による入力と print による出力



input による入力と print による出力

① trinket の次のページを開く


<https://trinket.io/python/bdca234a3e>

② 実行する。


③ 右下の画面で **3 Enter キー**

```
Powered by  trinket
teiken =
3
```

④ 右下の画面で, 続けて **5 Enter キー**

```
Powered by  trinket
teiken =
3
takasa =
5
```

⑤ 結果の **7.5** を確認

```
Powered by  trinket
teiken =
3
takasa =
5
('teihen * takasa / 2 =', 7.5)
```

⑥ **3角形の面積を求めるプログラム**である. いろいろ試してみよう.

```
1 print("teiken =")
2 teihen = float(input())
3 print("takasa =")
4 takasa = float(input())
5 print("teihen * takasa / 2 =", teihen * takasa / 2)
```




⑥ trinket の次のページを開く

<https://trinket.io/python/3b490869e4>


⑦ 実行する。

⑧ 右下の画面で **3 Enter キー**

```
1 print("r =")
2 r = float(input())
3 print("r * r * 3.14 =", r * r * 3.14)
```

```
Powered by  trinket
r =
3
```

⑨ 結果の **28.26** を確認

```
Powered by  trinket
r =
3
('r * r * 3.14 =', 28.26)
```

⑩ 円周率を 3.14 として、半径から円の面積を求めるプログラムである。いろいろ試してみよう。

文字列の変数



⑪ 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

⑫ 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 # ユーザー入力と変数:  
2 # input()関数を使ってユーザーからの入力を受け取り、  
3 # 変数に保存する。  
4 # 対話的なプログラムの基本を学び、  
5 # ユーザーとのインタラクションを通じて  
6 # 変数にデータを格納する方法を理解する。  
7  
8 user_name = input("あなたの名前を入力してください: ")  
9 print("ようこそ、" + user_name + "さん!")
```

Result

Powered by trinket
あなたの名前を入力してください: kaneko
ようこそ、kanekoさん!

- 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

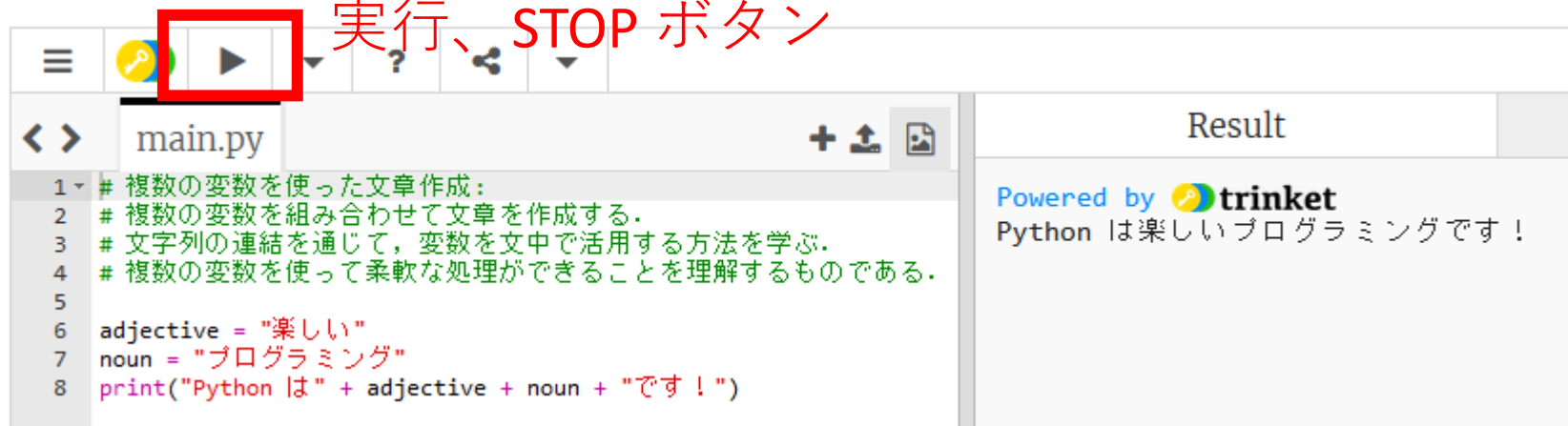
⑬ trinket の次のページを開く

<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


⑭ 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 # 複数の変数を使った文章作成:  
2 # 複数の変数を組み合わせて文章を作成する.  
3 # 文字列の連結を通じて、変数を文中で活用する方法を学ぶ.  
4 # 複数の変数を使って柔軟な処理ができることを理解するものである.  
5  
6 adjective = "楽しい"  
7 noun = "プログラミング"  
8 print("Python は" + adjective + noun + "です!")
```

Result

Powered by  trinket
Python は楽しいプログラミングです!

- 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

① 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（変数、式、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。

② コンピュータにおける入力と出力の重要性

コンピュータの入力と出力は、情報のやり取りの基本です。入力は、ユーザーや他のコンピュータからコンピュータへのデータの伝達であり、キーボード入力などが該当します。出力はその逆で、コンピュータが結果や情報をユーザーや他のシステムへ提供します。これらの機能はコンピュータ間やコンピュータとユーザー間の相互作用の基礎を形成します。

全体まとめ



- **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。

- **式と変数** : 式の中に**変数を含める**ことができる。例えば、「**x = x + 100**」は、**変数xに100を加える操作**である。

- **入力と出力**: コンピュータにデータを入力する操作を「**入力**」、コンピュータがデータを出力する操作を「**出力**」と呼ぶ。

Python の input はキーボードからデータを読み込む。
print はデータを表示する。

- **Trinket**: TrinketはPython等の学習サイトであり、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。また、プログラムの実行や変更も可能。