

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

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

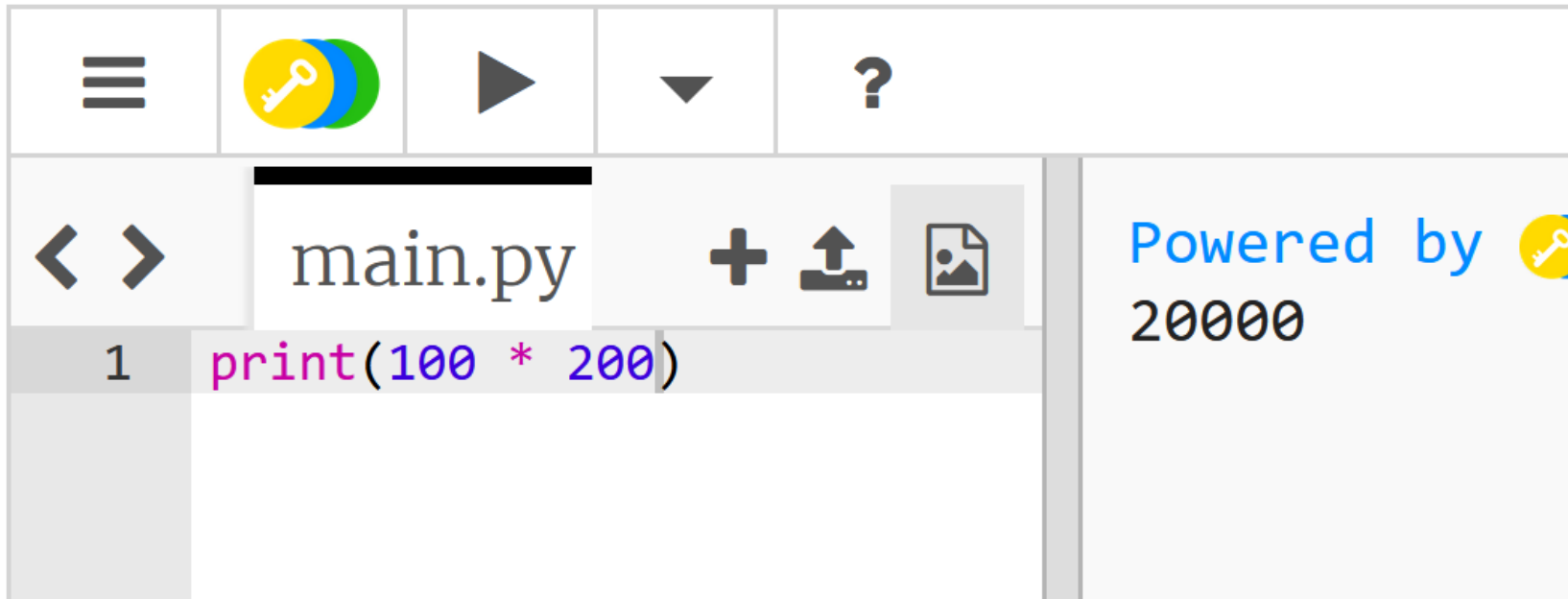
x = 100

プログラム

式の実行結果



式の実行結果として、値が得られる



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 bar contains icons for a menu, a key, a play button, a dropdown arrow, and a question mark. Below this, a tab labeled "main.py" is active. The code editor shows two lines of Python code: "1 x = 100" and "2 print('x = ', x)". To the right of the code editor, a panel labeled "Result" displays the output: "Powered by [key icon] tri" and "('x = ', 100)".

```
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 has a tab labeled "main.py". The code inside the tab is:

```
1 x = 100
2 y = 200
3 print(x + y)
```

The third line is highlighted. To the right of the code editor, there is a panel showing the execution result: "Powerec" in blue text and "300" in black text below it. The interface includes navigation arrows, a plus sign, an upload icon, and a file icon.

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

A screenshot of a Python IDE. The left pane shows a file named 'main.py' with the following code:

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

The right pane shows the output of the program:

```
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

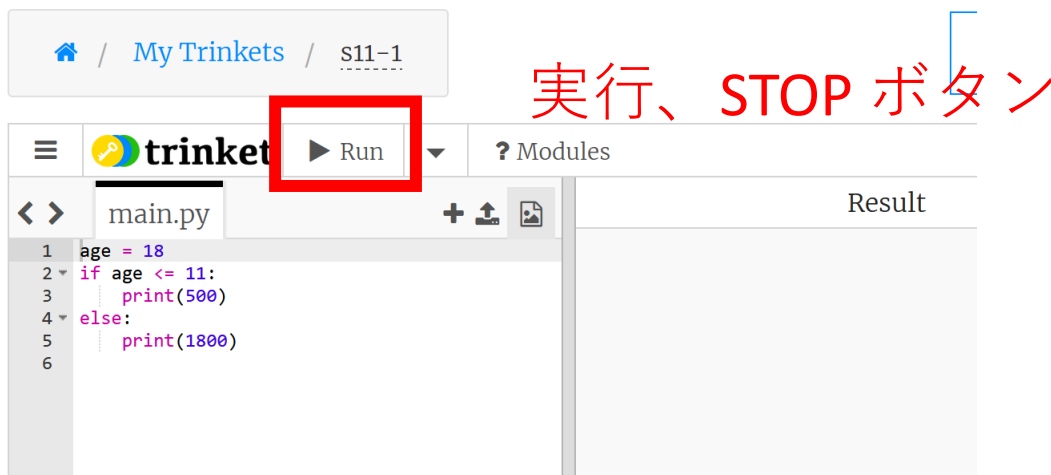
trinket でのプログラム実行



- 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

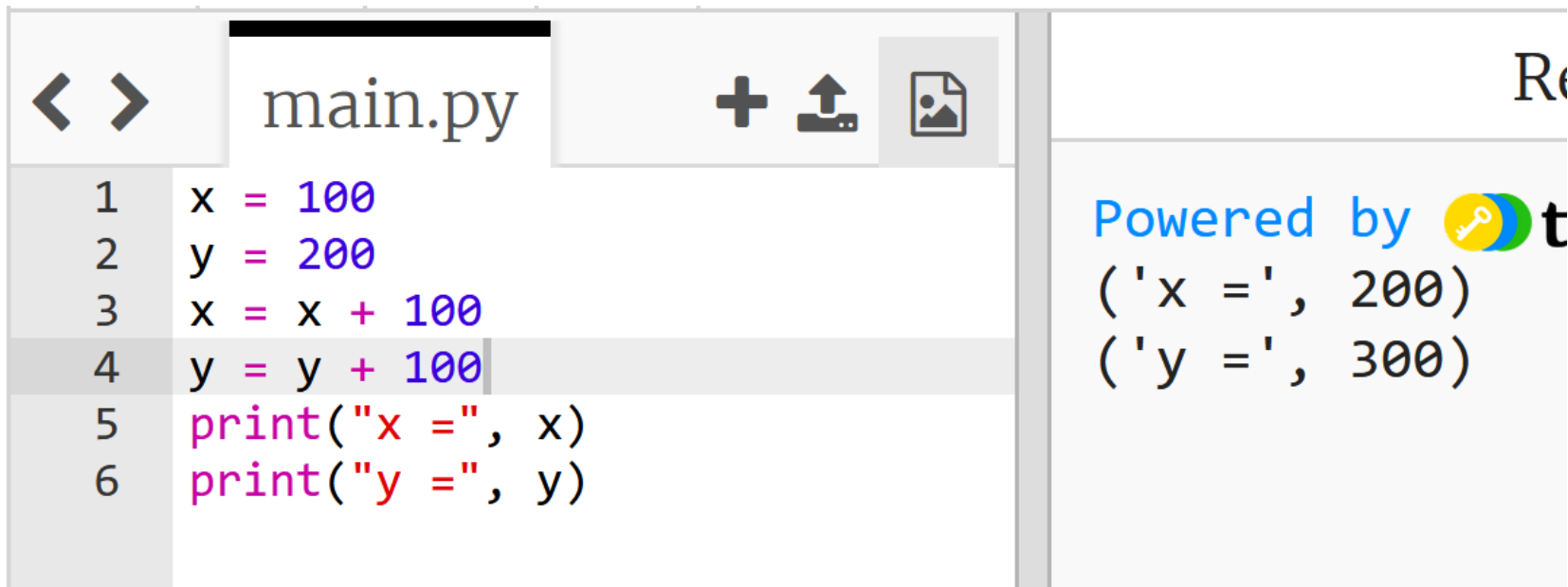
【トピックス】

- 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



The screenshot shows a Trinket.io interface. On the left, a code editor for 'main.py' contains the following Python code:

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

On the right, the output area displays the results of the code execution:

```
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
を確かめてください

```
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，正しいか確かめてください

```
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)

- **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

ページ 18, 19

【トピックス】

- trinket の利用
- input
- print

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)
```




① trinket の次のページを開く


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

② 実行する。

③ 右下の画面で **3 Enter キー**

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

④ 結果の **28.26** を確認

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

⑤ 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。

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

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。入力とは、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。