

# pf-7. 繰り返し, リスト

(Python 入門)

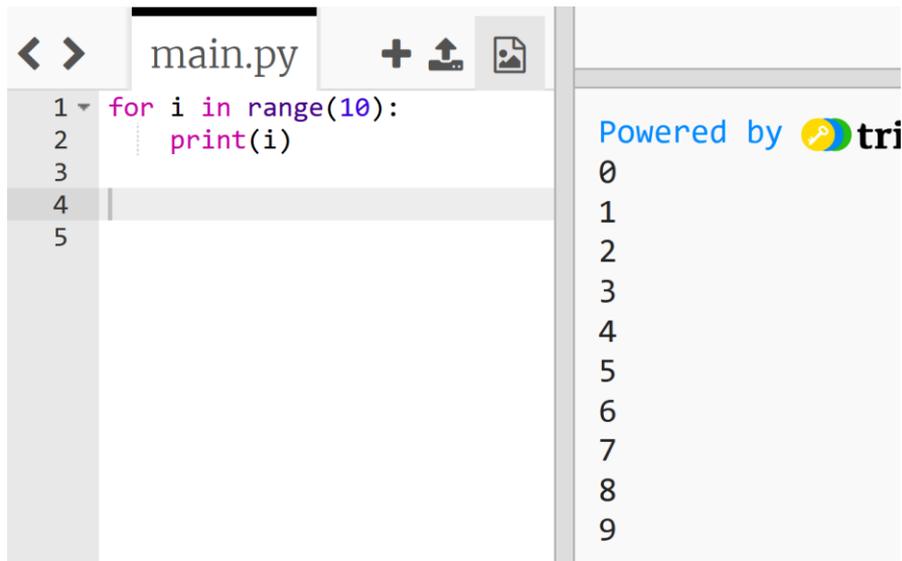
URL: <https://www.kkaneko.jp/pro/pf/index.html>

金子邦彦



# for による繰り返し

for を用いて**特定の処理を繰り返す**ことができる



```
main.py
1 for i in range(10):
2     print(i)
3
4
5
```

Powered by tri

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

プログラム

実行結果

- Python では, 「for i in range(10)」により, **i の値は, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 と変化**する.
- for で繰り返し処理したい部分は**字下げ**する

- 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

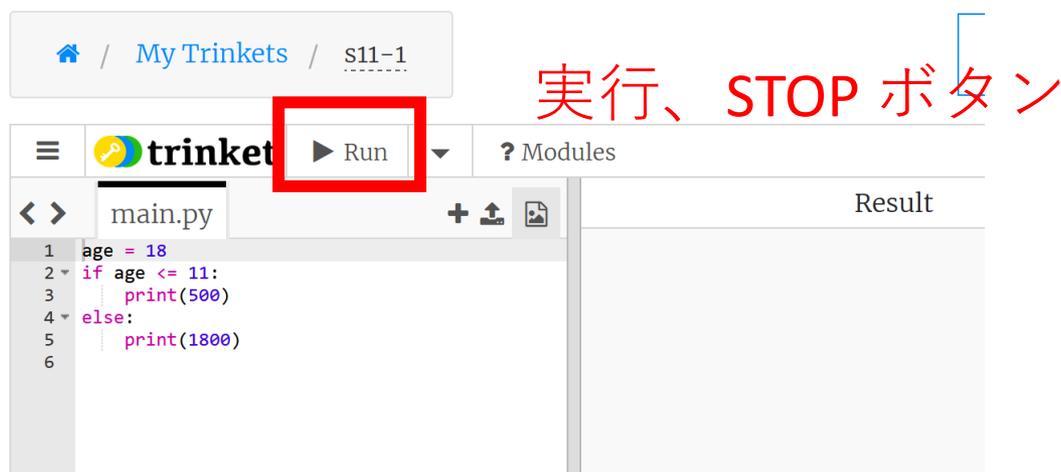
math, matplotlib.pyplot, numpy, operator, processing, pygal, random, re, string, time, turtle, urllib.request



# 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

## 繰り返しでの変数値の変化

ページ6～10

### 【トピックス】

- trinket の利用
- for による繰り返し
- for i in range(5) での i の値の変化
- 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/python/27f6ebe1da>

② 実行結果が、次のように表示されることを確認

```
1 for i in range(5):  
2     print(i * '&')  
3  
4  
5
```

Powered by 

&  
&&  
&&&  
&&&&

Python で「`i * '&'`」は、**& を  $i$  個**の意味になる

### ③ 確認クイズ

**& を 9個まで表示**したい（下図のように）ときは、  
どのようにプログラムを書き換えますか？

**自分でチャレンジ**してください

```
1 for i in range(5):  
2     print(i * '&')
```

**このプログラムを書き換える**

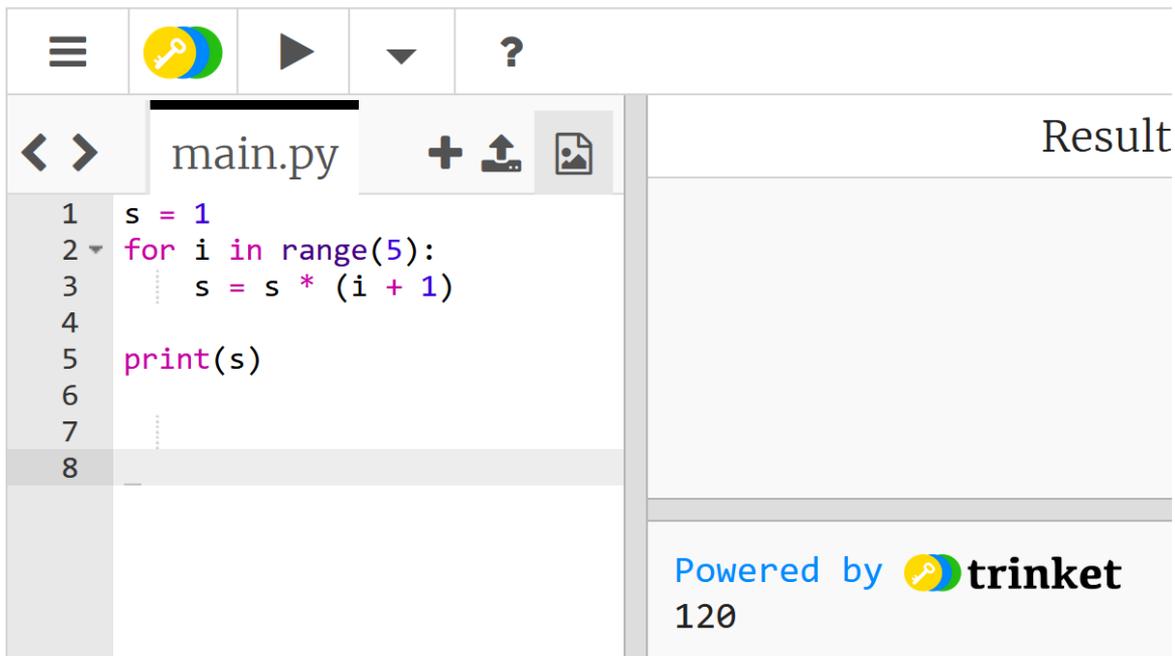
```
&  
&&  
&&&  
&&&&  
&&&&&  
&&&&&&  
&&&&&&&  
&&&&&&&&  
&&&&&&&&&  
&&&&&&&&&&
```

**このような結果を得る**

④ trinket の次のページを開く

<https://trinket.io/python/b7eb532453>

⑤ 実行する。  $1 \times 2 \times 3 \times 4 \times 5$  の計算が行われる。  
120 が表示される



```
1 s = 1
2 for i in range(5):
3     s = s * (i + 1)
4
5 print(s)
6
7
8
```

Result

Powered by  trinket  
120

## ⑥ 確認クイズ

**1 × 2 × 3 × 4 × 5 × 6 × 7 × 8**を表示したい（下図のように）ときは、どのようにプログラムを書き換えますか？

**自分でチャレンジしてください**

```
1 s = 1
2 for i in range(5):
3     s = s * (i + 1)
4
5 print(s)
```

**このプログラムを書き換える**

Powered by  **trinket**  
40320

**このような結果を得る**

# 答え合わせ



③

```
main.py
1 for i in range(10):
2     print(i * '&')
3
```

Powered by t

&  
&&  
&&&  
&&&&  
&&&&&  
&&&&&&  
&&&&&&&  
&&&&&&&&  
&&&&&&&&&

⑥

```
1 s = 1
2 for i in range(8):
3     s = s * (i + 1)
4
5 print(s)
6
7
8
```

Powered by t

40320

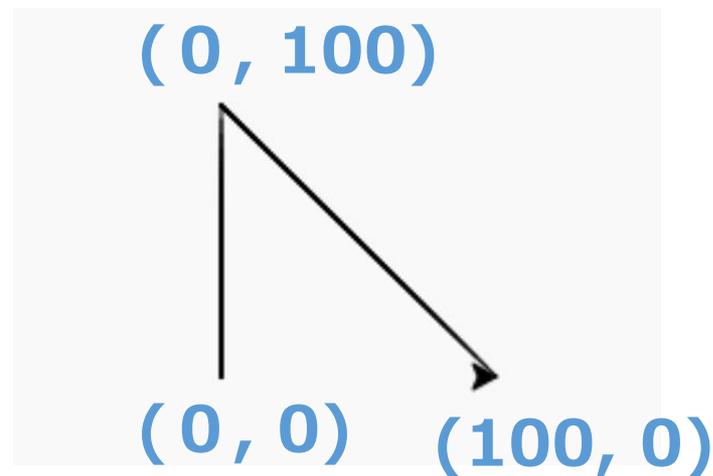
# タートルグラフィックス



カーソルを使って絵を描く

```
1 import turtle
2 t = turtle.Turtle()
3 t.goto(0,100)
4 t.goto(100,0)
```

タートルグラフィックスの機能をインポートする「import turtle」が必要



# タートルグラフィックス



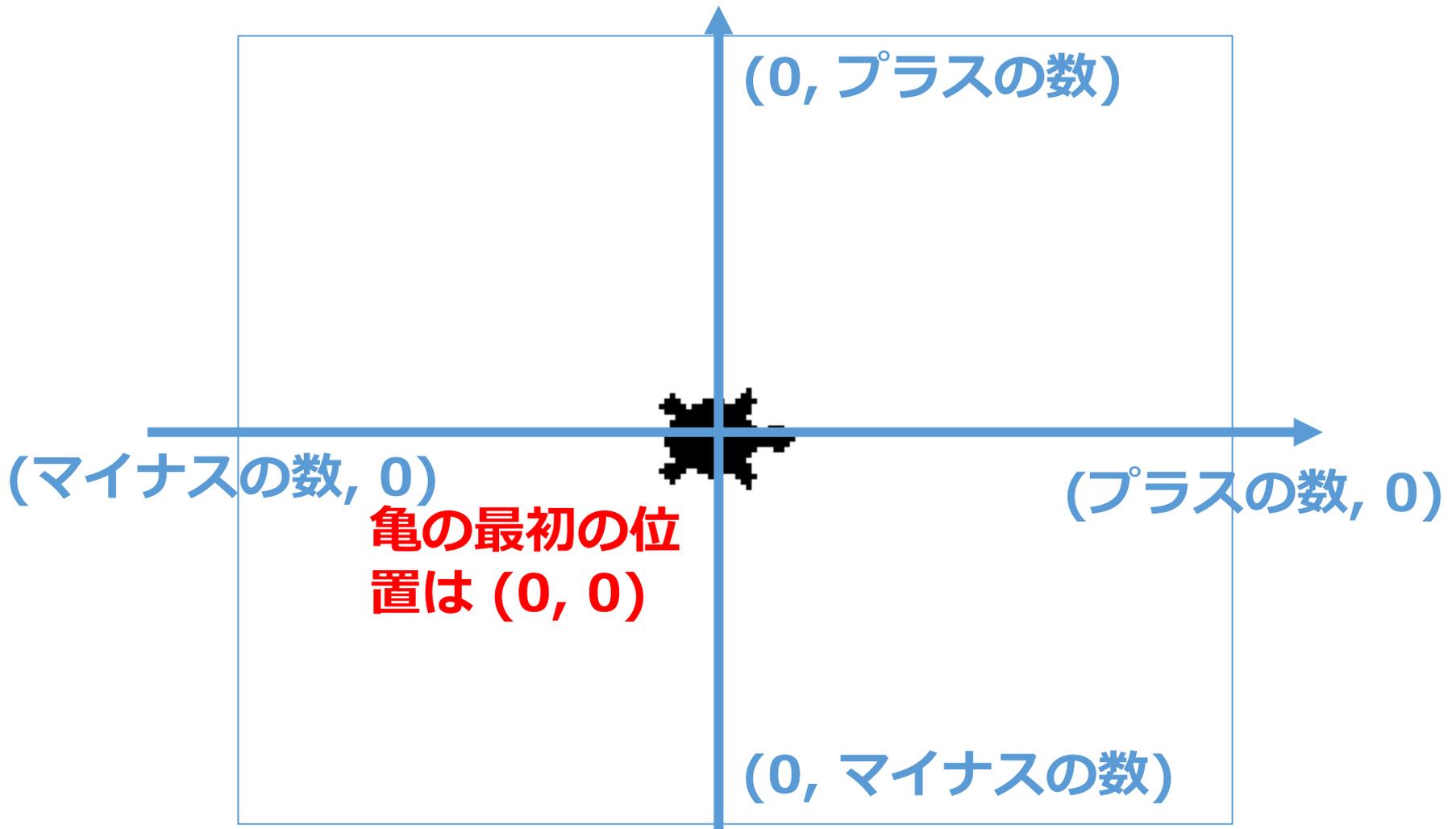
```
1 import turtle
2 t = turtle.Turtle()
3 t.goto(0,100)
4 t.goto(100,0)
```

オブジェクト    **メソッド**

- **メソッド**は、オブジェクトが持つ機能を呼び出すためのもの
- 「**goto**」は**指定した座標への移動**

## 主なメソッド

- **goto** ( <横方向の値> , <縦方向の値> )      **移動**
- **forward**( <移動量> )      **前進**
- **backward**( <移動量> )      **後退**
- **right**( <角度> )      **右回りに回転**
- **left**( <角度> )      **左回りに回転**



# 演習

## 繰り返し，変数値をマスターしていく

ページ 15, 16

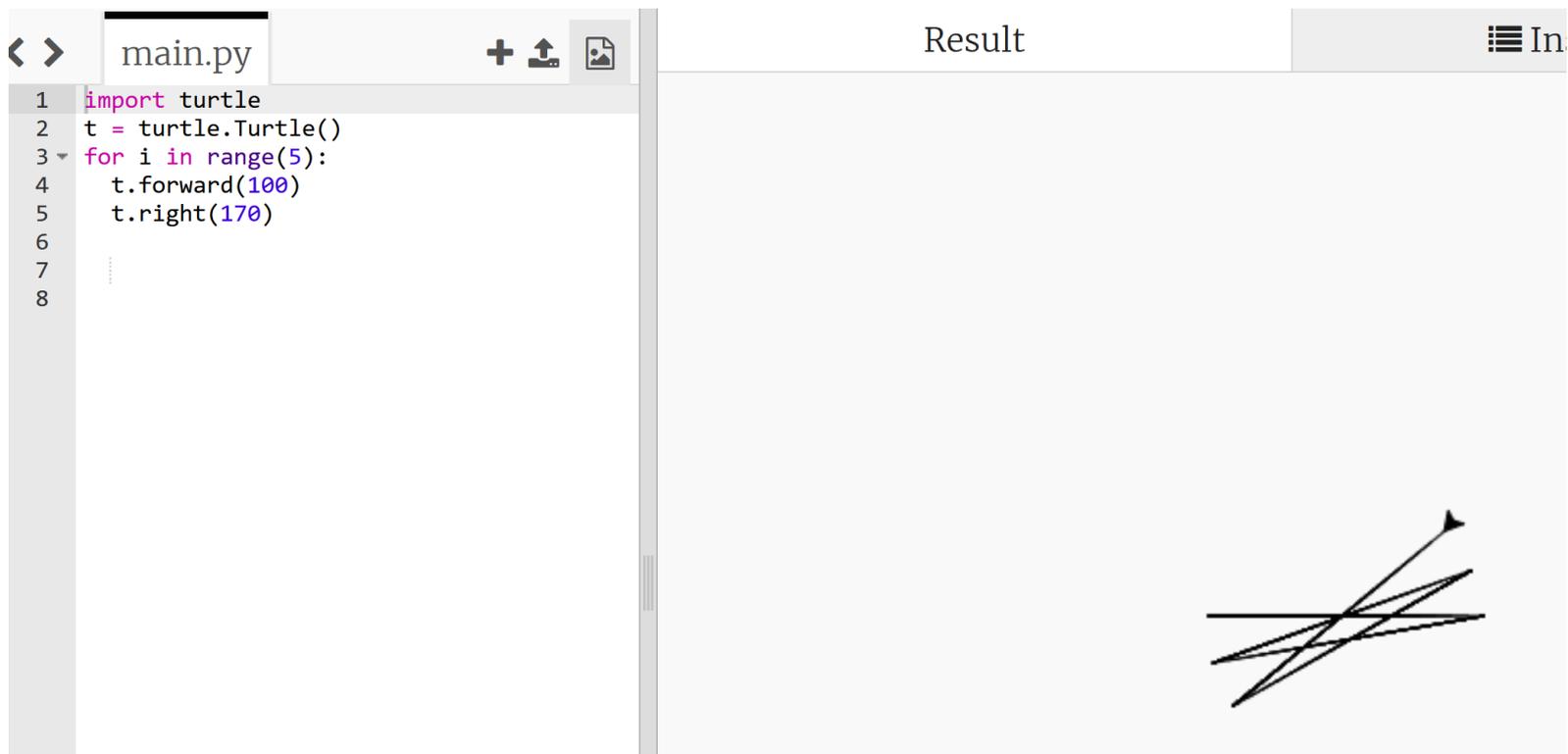
### 【トピックス】

- trinket の利用
- for による繰り返し
- for i in range での i の値の変化
- 自主的にプログラムを書き換えて，変化をみる

① trinket の次のページを開く

<https://trinket.io/python/895c3ea5b6>

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 code editor with a file named 'main.py'. The code is as follows:

```
1 import turtle
2 t = turtle.Turtle()
3 for i in range(5):
4     t.forward(100)
5     t.right(170)
6
7
8
```

The right side of the editor shows the 'Result' tab, which displays a drawing of a five-pointed star. The star is formed by five overlapping lines, each 100 units long, with a 170-degree angle between consecutive lines. The lines are black and the background is white.

③ trinket の次のページを開く

<https://trinket.io/python/0d8dbc1139>

④ 実行結果が、次のように表示されることを確認



```
main.py
1 import turtle
2 t = turtle.Turtle()
3 for i in range(5):
4     t.forward(i * 20 + 100)
5     t.right(170)
6
7
8
```

Result



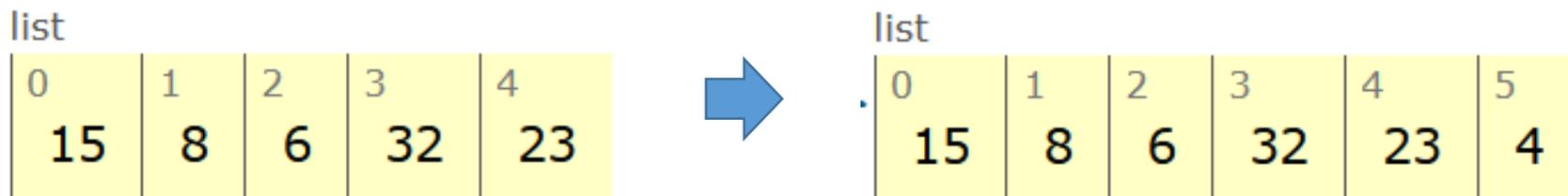
⑤ プログラム内の「5」や「20」や「100」や「170」をいろいろ書き換えて実行してみる

# Python のリスト



- リストは、**複数の要素を保持**できる
- リストの要素には順序を持ち、**順序の番号は 0 から開始**する
- リストの要素は変更可能（**新しい要素の挿入**，**既存の要素の削除**が可能）

4 を末尾に挿入



8 の削除



# Python でのリストの組み立て

list

0	1	2	3	4
15	8	6	32	23

リストの組み立てを行う Python プログラム

```
x = [15, 8, 6, 32, 23];
```

## 演習

資料：19～25

### 【トピックス】

- リスト
- 繰り返し
- for

# 1. 月の日数



6月 は **30**日 までである。 7月 は **31**日 までである。

※ うるう年のことは考えないことにする

① trinket の次のページを開く

<https://trinket.io/python/88a728c3cb>

② 実行する。 **30, 31** が表示されることを確認

```
main.py
1 days = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
2 print(days[6])
3 print(days[7])
4
```

Powered by  
30  
31

③ 8月や9月について表示できるように、プログラムを変更し実行してみる

trinket Run Modules

```
main.py  
1 days = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]  
2 print(days[6])  
3 print(days[7])  
4
```

Powered by  
30  
31

リストの  
組み立て

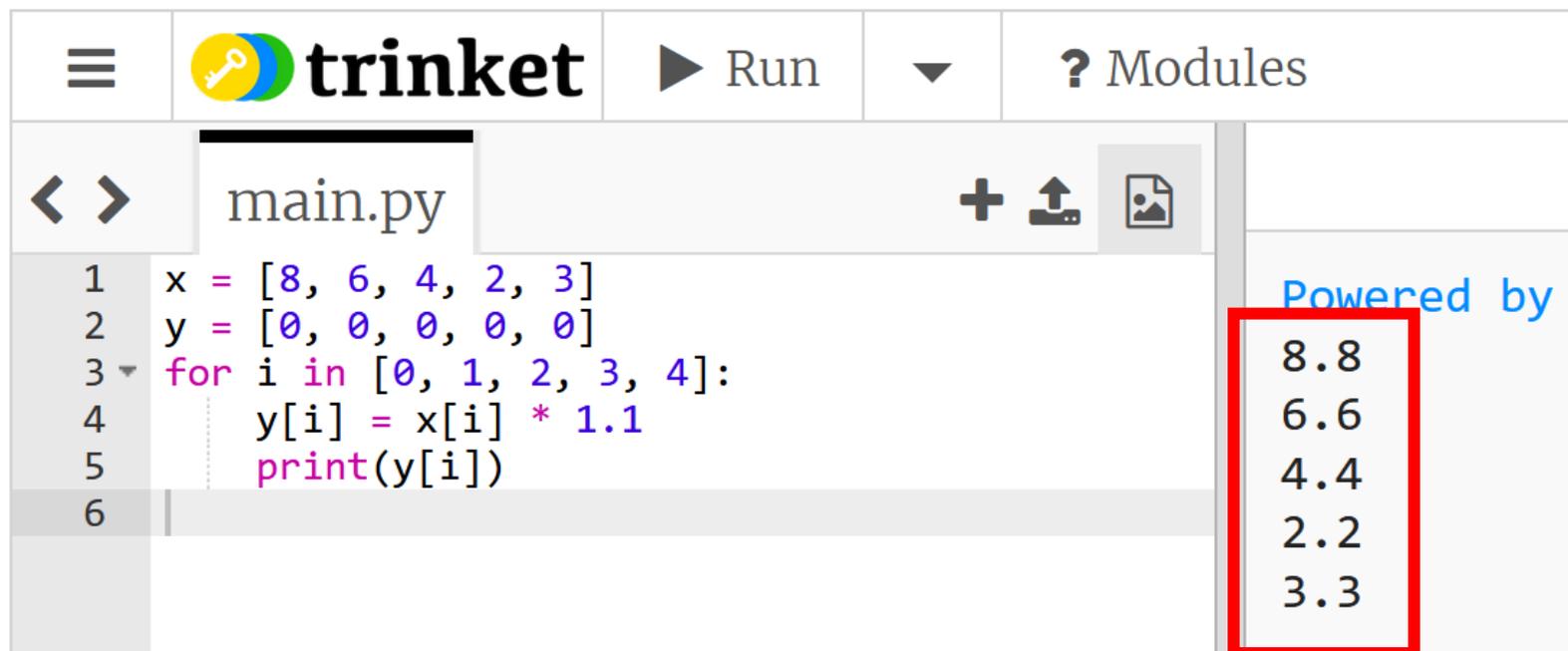
6番の要素、7番の要素の表示

## 2. 計算の繰り返し

① trinket の次のページを開く

<https://trinket.io/python/cc2c13d793>

② 実行する。結果が5つ表示されることを確認



```
1 x = [8, 6, 4, 2, 3]
2 y = [0, 0, 0, 0, 0]
3 for i in [0, 1, 2, 3, 4]:
4     y[i] = x[i] * 1.1
5     print(y[i])
6
```

Powered by

8.8  
6.6  
4.4  
2.2  
3.3

## リストの 組み立て

```
main.py  
1 x = [8, 6, 4, 2, 3]  
2 y = [0, 0, 0, 0, 0]  
3 for i in [0, 1, 2, 3, 4]:  
4     y[i] = x[i] * 1.1  
5     print(y[i])  
6
```

Powerrec  
8.8  
6.6  
4.4  
2.2  
3.3

「 $y[i] = x[i] * 1.1$ 」を  
 $i$  の値を変えながら  
5回繰り返す

### 3. 重力と落下距離

物体を落とすと  $9.8 \times (\text{時間})^2 \div 2$  の分, 落ちていく.

(空気抵抗は無視する). 時間は 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10  
⇒ 同じ式の計算を 11 回繰り返し

① trinket の次のページを開く

<https://trinket.io/python/e27702ef75>

② 実行する。結果が表示されることを確認

```
main.py
1 x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2 for t in x:
3     print((9.8 / 2) * t * t)
4
```

Powered by  trinket

0.0  
4.9  
19.6  
44.1  
78.4  
122.5  
176.4  
240.1  
313.6  
396.9  
490.0

# 4. リストの組み立て, 要素の挿入と削除



```
x = [15, 8, 6, 32 ,23]
print(x)
x.append(4)
print(x)
x.remove(8)
print(x)
```

① trinket の次のページを開く

<https://trinket.io/python/f33df42c8d>

② 実行する。結果が表示されることを確認

```
main.py
1 x = [15, 8, 6, 32 ,23]
2 print(x)
3 x.append(4)
4 print(x)
5 x.remove(8)
6 print(x)
```

Powered by trinket

```
[15, 8, 6, 32, 23]
[15, 8, 6, 32, 23, 4]
[15, 6, 32, 23, 4]
```

# リストのプログラム例



≡  trinket ▶ Run ▼ ? Modules

< > main.py



Powered by  
June

```
1 # リストの作成
2 months = ["", "January", "February", "March", "April", "May", "June",
3 "July", "August", "September", "October", "November", "December"]
4 print(months[6])
```

# リストを演習できる オンラインサイトの紹介

# 「リスト」を演習できる オンラインサイトの紹介



リストは、同じ型の要素の並び

① ウェブブラウザを起動する

② 次の URL を開く

<https://visualgo.net/ja>

③ 「**Linked List** (連結リスト)」をクリック



The screenshot shows the Visualgo website homepage. The main navigation bar includes the logo "VISUALGO.NET/EN" and a "LOGIN" button. Below the logo is the tagline "visualising data structures and algorithms through animation" and a search bar. The NUS Computing logo is also visible. A "Do You Know?" section provides information about new features. At the bottom, there are three category tiles: "Sorting", "Bitmask", and "Linked List". The "Linked List" tile is highlighted with a red border and shows a diagram of a linked list with nodes labeled 'H' and 'T'.

VISUALGO.NET/EN LOGIN

Competitive Programming 4 Book 1 Book 2 Buy Now!

Search...

NUS Computing  
National University of Singapore  
Featured story: Visualizing Algorithms with a Click

Do You Know? Next Random Tip

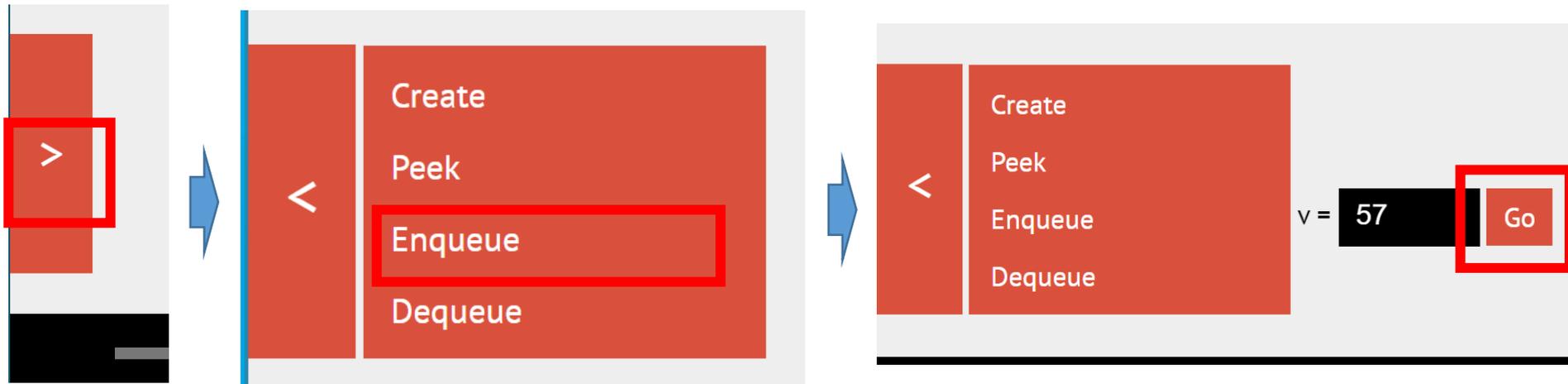
Please check the newest features of VisuAlgo: 1) User accounts system for NUS students and verified CS lecturers worldwide (and also read the latest Privacy Policy popup at the bottom right corner), 2) More mobile-friendly setup, 3) More polished e-Lecture notes to reach 'NUS standard', and 4) Trilingual capability (/en, /zh, or /id).

Sorting Training  
array algorithm bubble select

Bitmask Training  
bit manipulation set cs3233

Linked List Training  
stack queue doubly deque

- ④ 説明が出る場合がある. **ESC** キーを押して, 説明を消す
- ⑤ 左下のメニューで「**Enqueue** (入れる)」をクリックし, 「**Go**」をクリック



⑥ 末尾に挿入されるので、確認する



# 全体まとめ



- **for** を用いて**特定の処理を繰り返す**ことができる。
- 「for i in range(10)」は**0から9までの値**を i に**順次代入**する
- **タートルグラフィックス**は**カーソルで絵を描く**ツールで、goto、forward、backward、right、leftなどのメソッドで操作
- Pythonの**リスト**は、**複数の要素を一度に保持**できる。リストの要素は**順序を持ち**、番号は0から始まる。

```
months = [ "", "January", "February", "March", "April",  
"May", "June", "July", "August", "September", "October",  
"November", "December" ]
```

```
print(months[6])
```

- **繰り返し**: 繰り返し計算では、**特定の計算を何度も行う**。