1. プログラミングの基礎と Python 言語入門：創造的なデジタルスキル

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

1-1. プログラミングの基本と意義

コンピュータとプログラムの関係

- **コンピュータ**は，**プログラム**に従って動作
- **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

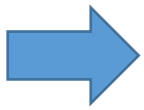
この関係が，コンピュータシステムの基礎

プログラムの本質

- **プログラム**を設計し作成するプロセス（プログラミング）は，**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
- 複雑な作業も**自動化**し，効率化することが可能

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

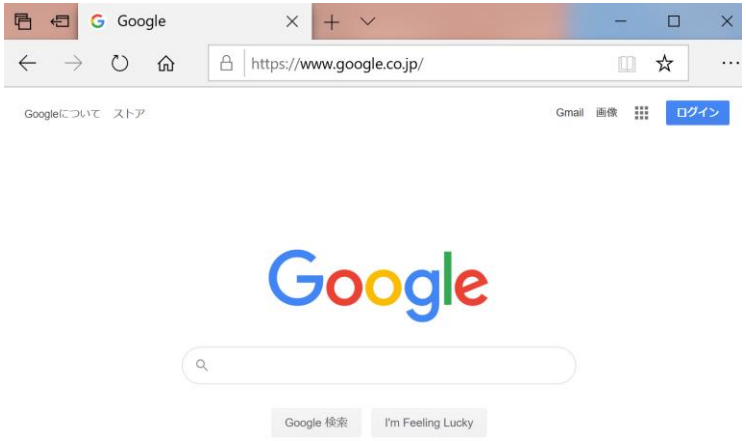
Python 言語のプログラム



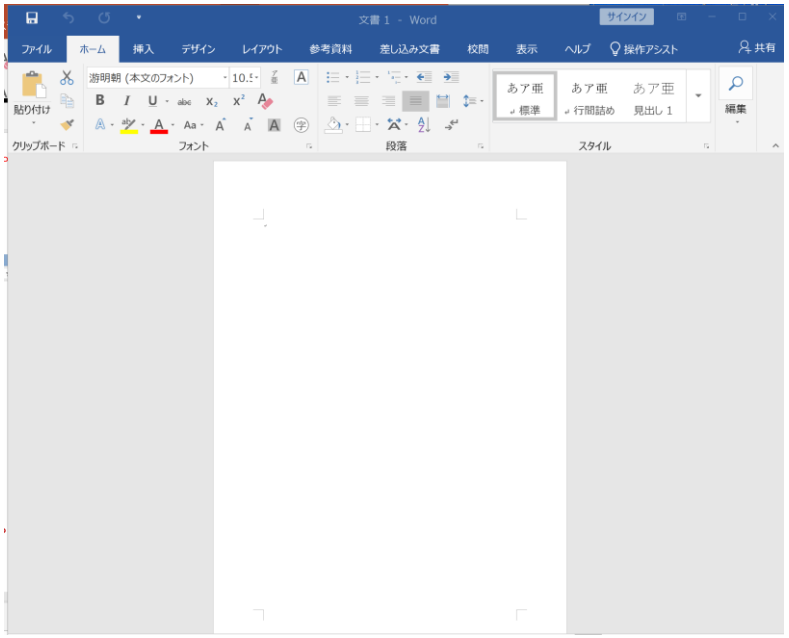
```
216.0
432.0
324.0
```

プログラムの
実行結果

① プログラムによるアプリケーションの実現



Web ブラウザ



ワープロ
（マイクロソフト・ワード）

プログラムが動作し，**アプリケーション**の機能を実現

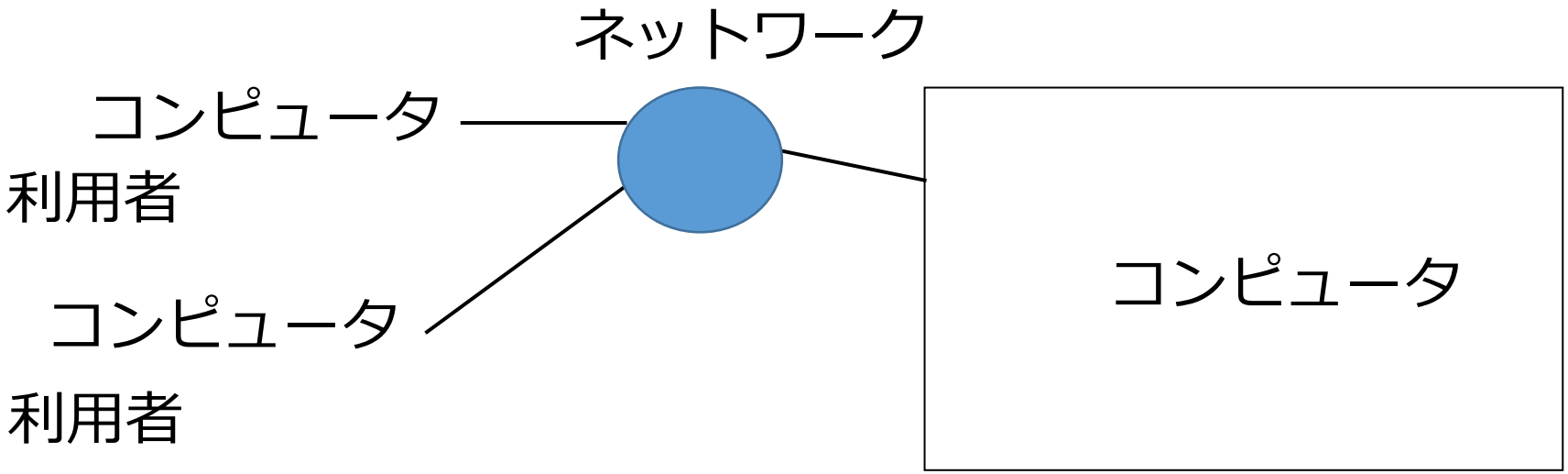
② プログラムによるコンピュータの制御

```
In [7]: from keras.models import Sequential
   ...: model = Sequential()
   ...: from keras.layers import Dense, Activation
   ...:
   ...: model.add(Dense(units=64, input_dim=len(x_train[0])))
   ...: model.add(Activation('relu'))
   ...: model.add(Dense(units=max(set(y_train)) - min(set(y_train)) + 1))
   ...: model.add(Activation('softmax'))
   ...: model.compile(loss='sparse_categorical_crossentropy',
   ...:               optimizer='sgd',
   ...:               metrics=['accuracy'])
   ...: model.fit(x_train, y_train, epochs=200)
   ...: score=model.evaluate(x_test, y_test, batch_size=1)
   ...: print(score)
   ...: model.predict(x_test)
   ...: model.summary()
Epoch 1/200
3/3 [==============================] - 0s 5ms/step - loss: 1.0583 - accuracy:
0.3200
Epoch 2/200
3/3 [==============================] - 0s 0s/step - loss: 1.0530 - accuracy:
0.3200
Epoch 3/200
3/3 [==============================] - 0s 0s/step - loss: 1.0485 - accuracy:
0.3200
```

Python 言語を使って
ニューラルネットワーク
を作成． AIシステムを構築

プログラムは，**コンピュータ**の動作を細かくコントール

③ プログラムによるコンピュータ間連携の実現



サーバは，サービスを提供するITシステム

コンピュータ同士の接続でも**プログラム**が必要.

プログラミングの４つの基本的特徴

① **プログラム**の内容によって，**コンピュータ**は**さまざまな作業を実行**できる

② **プログラム**を利用することで，多くの作業を**自動化**できる

③ **プログラム**で行った作業を**いつでも再現できる**.

④ **プログラム**は柔軟性があり，変更により，**プログラムの動作を簡単に調整できる**

プログラミングの魅力

- **未来の技術を学ぶ**ことは楽しい
- **プログラミング**は**クリエイティブな行為**
- 視覚的なプログラムを書くことで、ゲーム感覚をもって楽しみながら学習することも可能

プログラミングの達成感

- **自分のアイデアを形に**することで得られる**達成感**
- **自分でデザイン**し、問題が生じたときは自分で解決していく
- **自分の手でプログラムを完成させるプロセス**は、大いに**充実感**をもたらすもの

プログラミングの活用領域

- **プログラム**は**人間の力を増幅**し、私たちができることを大幅に広げる
- **シミュレーション、大量データ処理、AI連携、ITシステム制作**など、さまざまな活動で、**プログラミング**は役立つ
- **プログラム**を活用することで，**複雑な作業の自動化**が可能である

1-2. Python言語の特徴とプログラミングの可能性

Python 言語の概要

- **Python** は多くの人々に利用されている**プログラミング言語**の１つ
- **読みやすさ，書きやすさ，幅広い応用範囲**が特徴

```
   from keras.models import Sequentia
  : model = Sequential()
 .: from keras.layers import Dense, Ac
 .:
..: model.add(Dense(units=64, input_di
..: model.add(Activation('relu'))
...: model.add(Dense(units=max(set(y_tr
...: model.add(Activation('softmax'))
...: model.compile(loss='sparse_categor
...:               optimizer='sgd',
...:               metrics=['accuracy']
...: model.fit(x_train, y_train, epochs
...: score=model.evaluate(x_test, y_tes
...: print(score)
...: model.predict(x_test)
...: model.summary()
och 1/200
'3 [==============================] - 0s
3200
och 2/200
3 [==============================] - 0s
200
ch 3/200
 [==============================] - 0s
0
```

Python 言語の特徴

- **Python** は，**直感的で読みやすい**文法構造
- **シンプルなスクリプト**から、**高度なプログラム**まで，さまざまな規模の開発に対応できる**柔軟性**を備える

Python 言語の主な利点

文法のシンプルさ

直観的で読みやすい

例　**print** で簡単に**出力**

if や **else** で**条件分岐**

for や **while** で**繰り返し（ループ）**

字下げでブロック構造を示す

拡張性

多岐にわたる分野で利用が可能

例　**関数やクラスを定義**する **def** や **class**

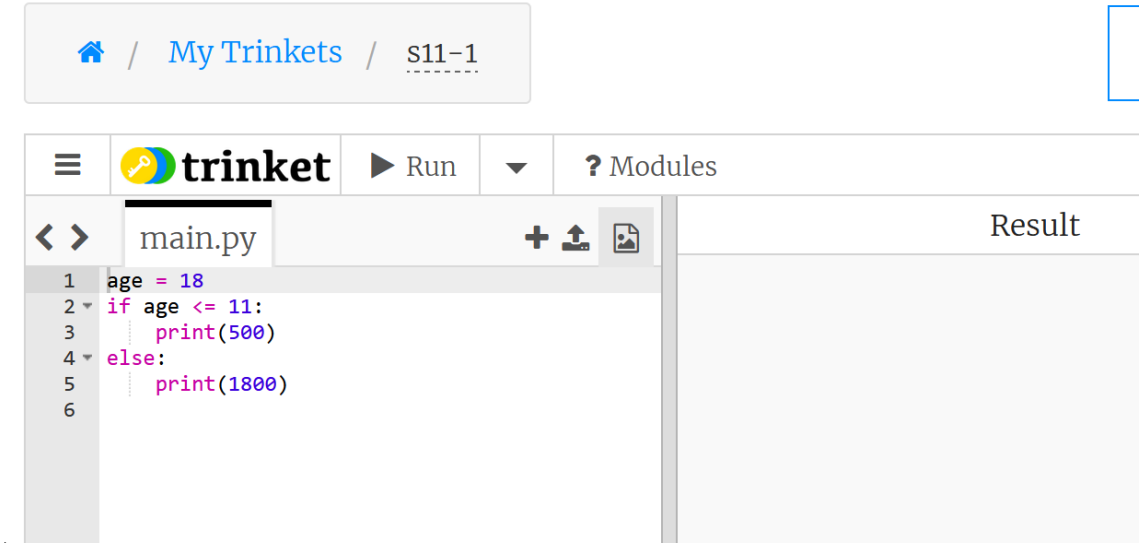
継承や**オブジェクトの属性名と値**を操作する **super** や **vars**

柔軟性

シンプルなスクリプトも、高度なプログラムも作成可能

Trinket の概要

- **Trinket は<u>オンライン</u>の Python、HTML 等の<u>学習サイト</u>**
- 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.
Trinket による Python プログラム実行

① trinket の次のページを開く

https://trinket.io/python/6c652f1c2f

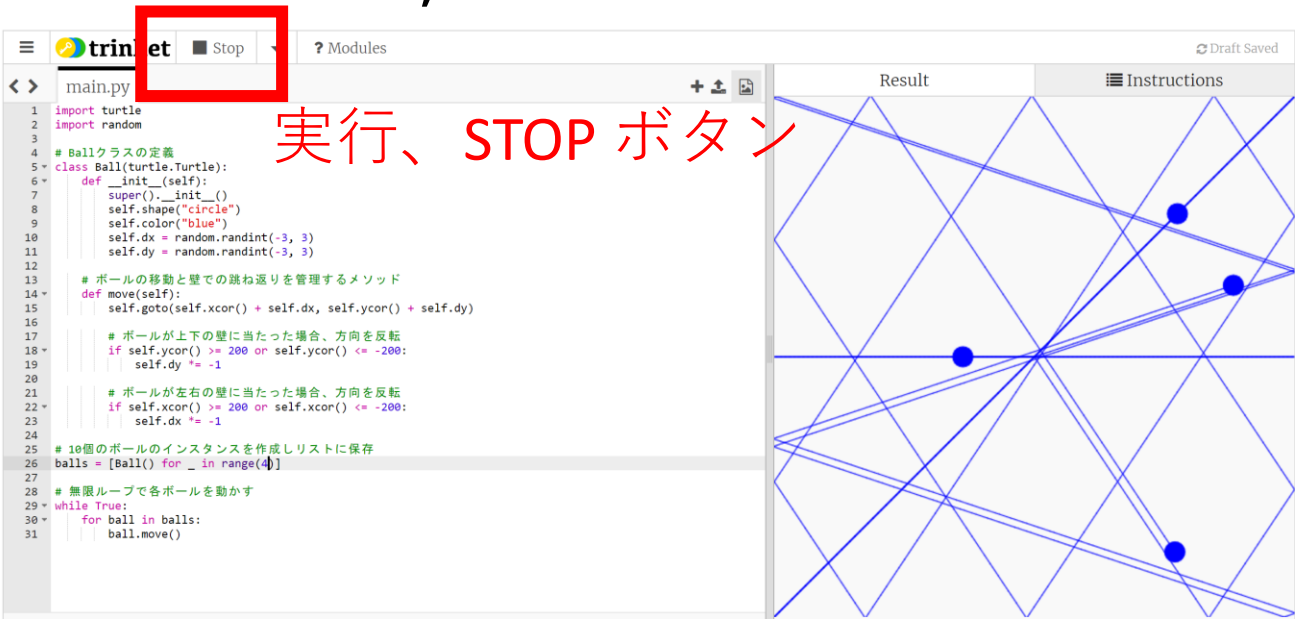
② 実行結果が，次のように表示されることを確認

- 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 trinket の次のページを開く

https://trinket.io/python/94d1563844

④ 実行結果が，次のように表示されることを確認



ボールが
壁に当たったら
反射する.

- 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

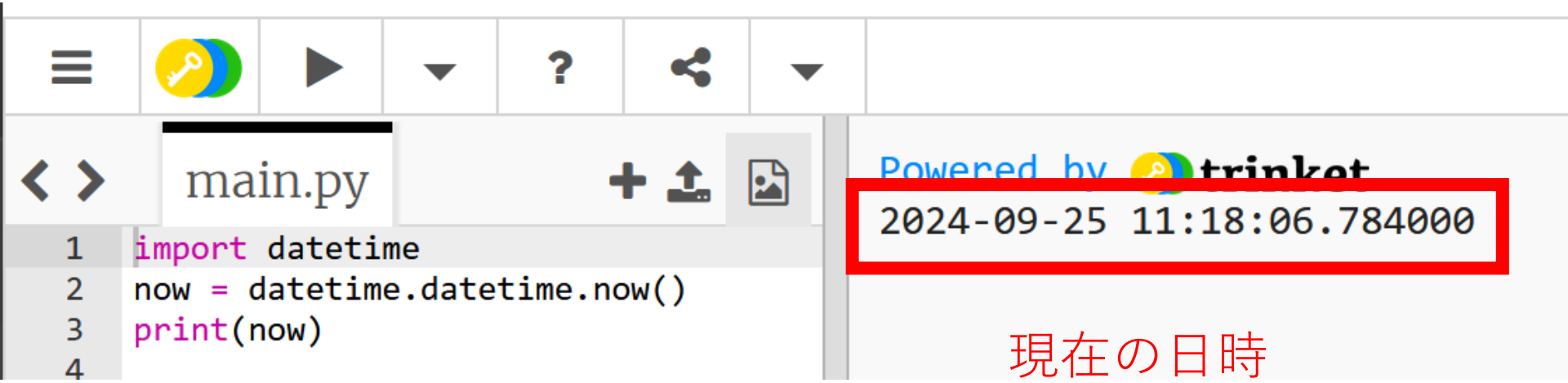
演習.
基本的な計算から高度な数学処理まで

オペレーティングシステム（コンピュータ）のタイマーを利用した**現在日時の表示**

① trinket の次のページを開く
https://trinket.io/python/2b804ab19a

② 実行結果が，次のように表示されることを確認

```
import datetime
now = datetime.datetime.now()
print(now)
```

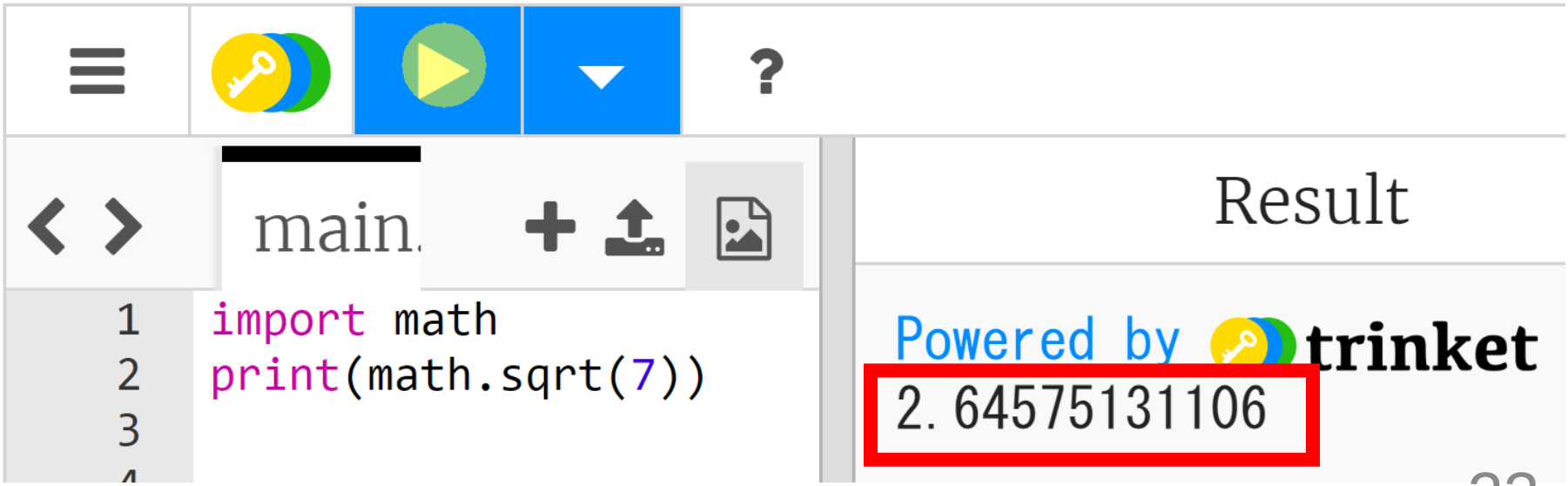


平方根：面積が 7 の正方形の一辺の長さ

③ trinket の次のページを開く
https://trinket.io/python/597e5771ff

④ 実行結果が，次のように表示されることを確認

```
import math
print(math.sqrt(7))
```

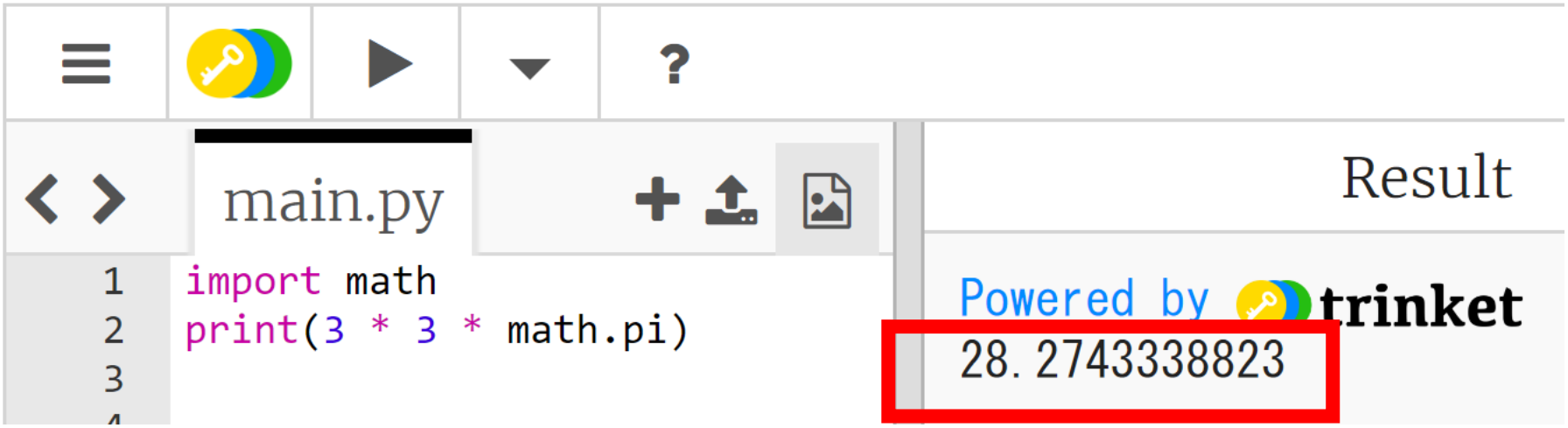


円周率：半径 **3** の円の面積は？

⑤ trinket の次のページを開く
https://trinket.io/python/4e3559f879

⑥ 実行結果が，次のように表示されることを確認

```
import math
print(3 * 3 * math.pi)
```

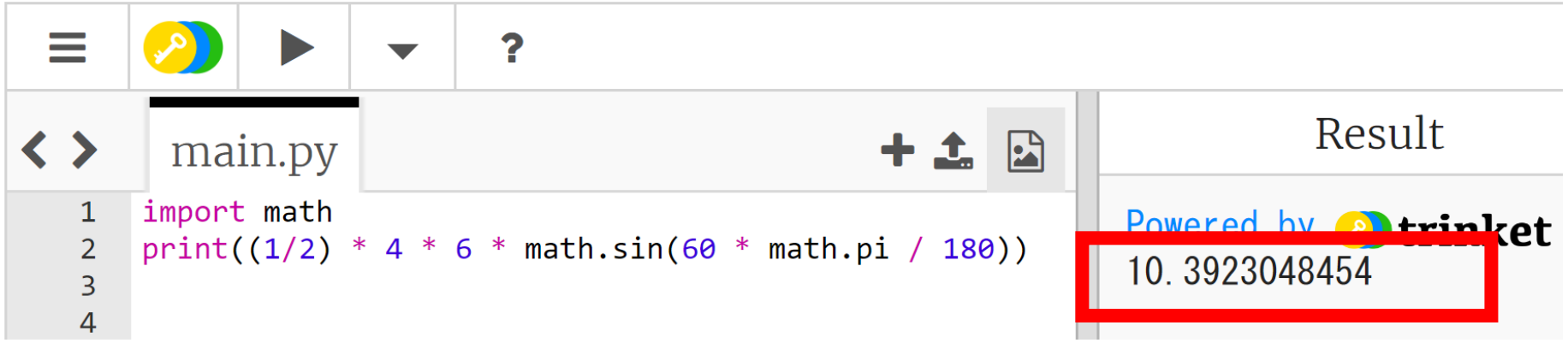


三角関数：三角形の２辺の長さが，**４**と**６**で，その間の角度が**６０**度のとき，面積は (1/2) × **４** × **６** × sin(**６０**)

⑦ trinket の次のページを開く
https://trinket.io/python/bdcce27488

⑧ 実行結果が，次のように表示されることを確認

```
import math
print((1/2) * 4 * 6 * math.sin(60 * math.pi / 180))
```



プログラミングの意義と可能性

- **プログラミング**は，**人間の力を増幅**し、私たちができることを大幅に広げる技術である
  - **シミュレーション**
  - **大量データ処理**
  - **AI連携**
  - **ITシステム制作**など
- プログラミングは**クリエイティブ**な行為
- さまざまな作業を**自動化**したいとき、**問題解決**したいときにも役立つ
- 論理的思考力の向上
- 問題解決能力の育成
- デジタル社会での必須スキル

プログラミングの応用分野

- **Web開発**

フロントエンド（HTML, CSS, JavaScript），バックエンド（Python, Django, Flask）

- **データ分析**

ビッグデータ処理，統計分析，データビジュアライゼーション（データの可視化）

- **人工知能**

自然言語処理，コンピュータビジョン（画像認識技術），予測モデリング

- **ゲーム開発**

2Dゲーム，3Dゲーム，モバイルゲーム

- **IoT（Internet of Things）**

センサーデータの収集と分析，スマートホームシステム

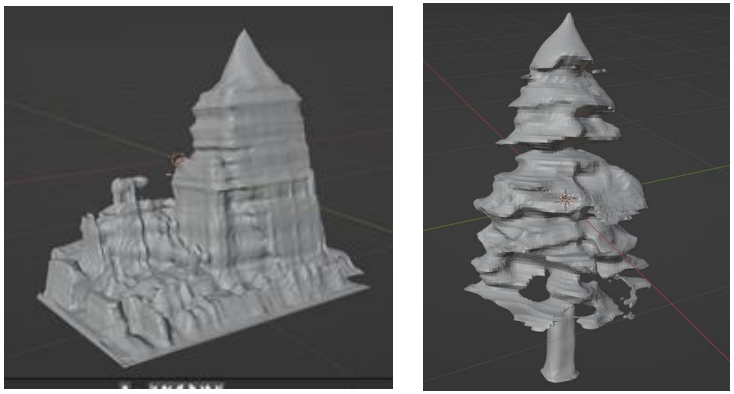
- **サイバーセキュリティ**

ネットワークセキュリティ，暗号化技術

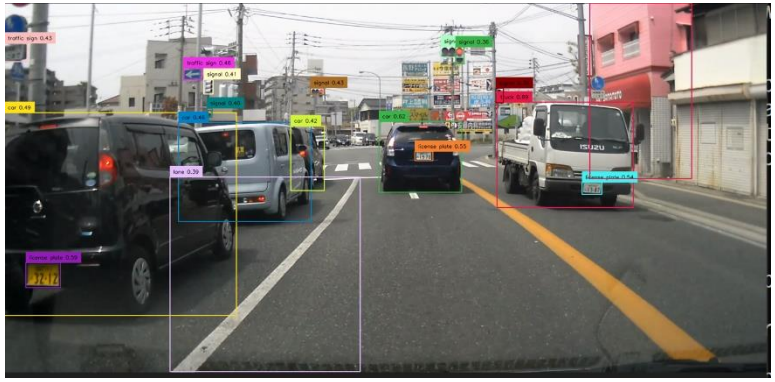
人工知能を学ぶ楽しさ

人工知能：人間の思考を模倣し、超えることを目指す挑戦。
人工知能は、現在進行形の最先端技術であり、未来に向けてもその発展が続く、刺激と興奮に満ちた分野である。

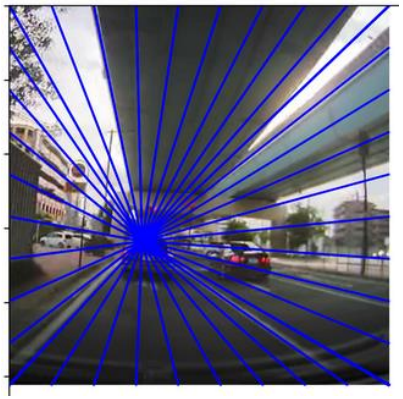
顔情報処理



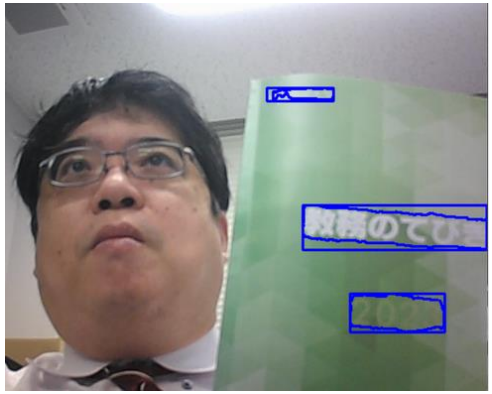
3次元データの合成



物体検出、指定されたキーワードはAIには初見（ゼロショット）



消失点推定



テキスト検出

Python プログラムのパソコン上での実行

Python プログラムはオンライン実行（例：Trinket）のほか，パソコンでも実行可能．（パソコンでの実行の場合には，Python 処理系のインストールが必要）

① Python プログラムのファイル保存

```
x = 100
if (x > 20):
    print("big")
else:
    print("small")
s = 0
for i in [1, 2, 3, 4, 5]:
    s = s + i
print(s)
```

作成した**Python プログラム**の**ソースコード**を，例えば「foo.py」という名前の**ファイルに保存**

② Python プログラムの実行

```
kaneko@www:/tmp$ python foo.py
big
15
```

プログラムを実行するには，シェル（例えば，Windows の場合はコマンドプロンプト）を開き，「python foo.py」のようなコマンドで実行

まとめ

- **Python**は**直感的で読みやすい**文法を持つプログラミング言語.
- **文法のシンプルさ**，**拡張性**，**柔軟性**が特徴的.
- **基本的な計算**から**高度なプログラミング**まで幅広く対応可能.
- **Web開発，データ分析，AI，ゲーム開発など応用分野が多岐**にわたる.
- **プログラミング**は**人間の能力を増幅**し，**創造的な活動を支援**する重要な活動である.