

# パソコンに学習させてみよう

```
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 100
正答率: 98.00%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 50
正答率: 94.00%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 10
正答率: 92.14%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 1
正答率: 32.89%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 5
正答率: 77.93%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 120
正答率: 100.00%
```

# この教材でできること

- ・ 機械学習について理解できる
- ・ ライブラリについて理解できる

## どんな教材？

パソコンに花の特長を学習させて実際に花の種類を答えさせるような学習をさせる

# 目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう
- ③コードを書いてみよう
- ④答え合わせをしてみよう

# 目次

①コードを書く前の準備をしよう

②完成系を確認してみよう

③コードを書いてみよう

④答え合わせをしてみよう

# ①コードを書く前の準備をしよう

このページは環境設定などの準備をするためのページ

# ①コードを書く前の準備をしよう

環境設定や使うファイル、ライブラリなどの準備を行う

```
(base) PS C:\Users\小西 翔 > conda activate learning  
(learning) PS C:\Users\小西 翔 >
```

# ①コードを書く前の準備をしよう

ライブラリは必要であればその都度追加しよう！

# ①コードを書く前の準備をしよう

下記のURLから花のデータをダウンロードしておこう！ダウンロード場所はどこでもいいけど、プログラムファイルのパスの近くがおすすめだよ

<https://www.koni-prog-online.com/textpage/webpage/web/level7/python16/Iris.csv>

# ①コードを書く前の準備をしよう

機械学習とは:コンピューターが大量のデータの中から法則性を見つけて学習する統計的な技術

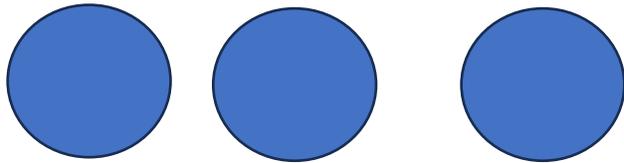
例:1 2 4 8 16 ...

コンピューターが数字の上がり幅を見て2倍ずつ増えていることを自分で見つけ出すよ

では、1024の次に来る数字は？

# ①コードを書く前の準備をしよう

人が見たら円か四角形かすぐわかるけど、そもそも円と四角形の違いは何だろう？それをパソコンに学習させてみよう！



これらは円だ！



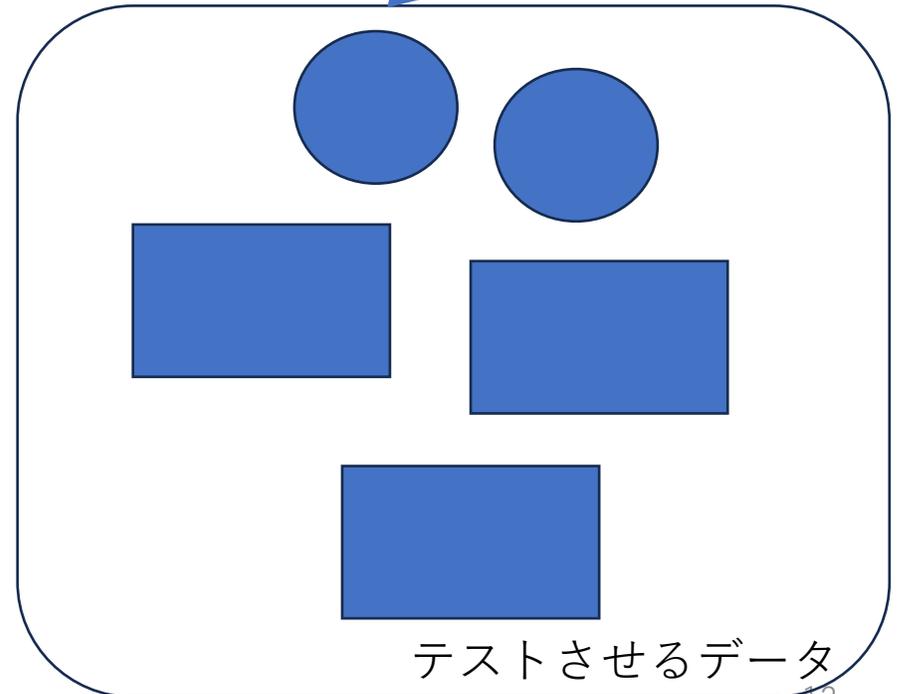
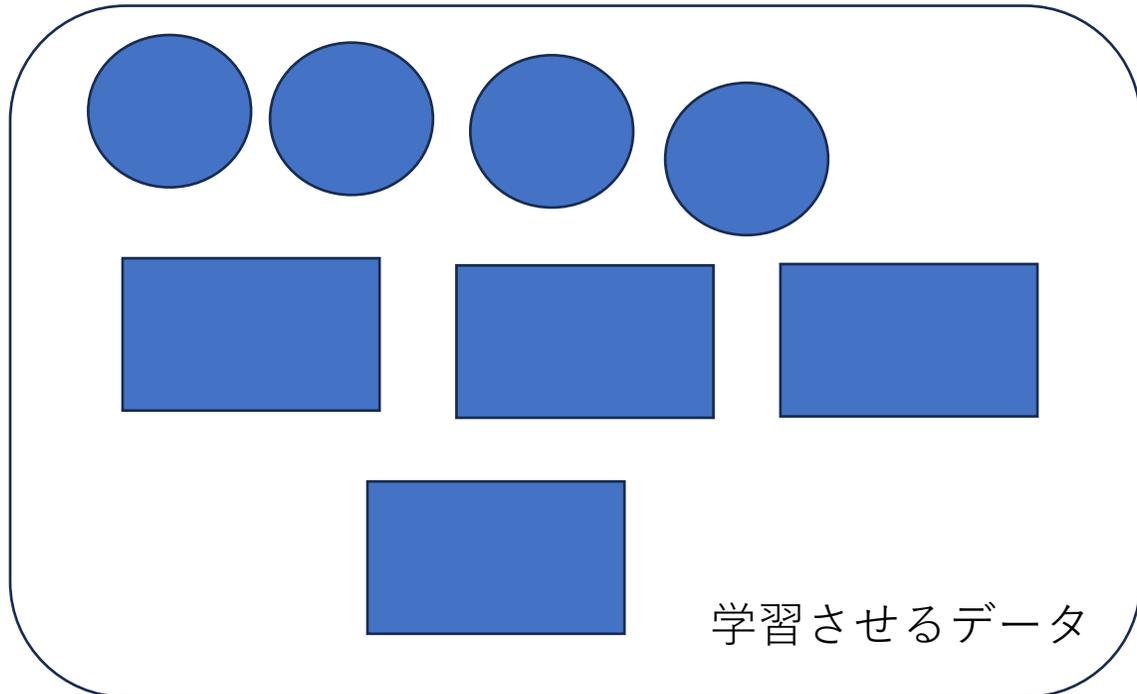
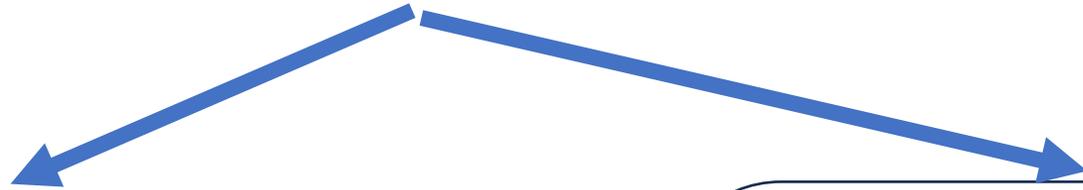
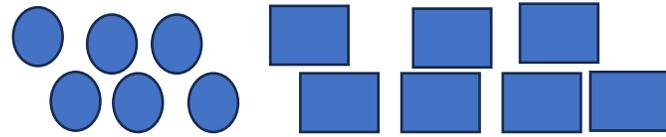
これらは四角形だ！

# ①コードを書く前の準備をしよう

ここに複数の図形があるよ！機械学習の手順はこの中から学習させるデータとテスト用のデータに分けるところから始まるよ！

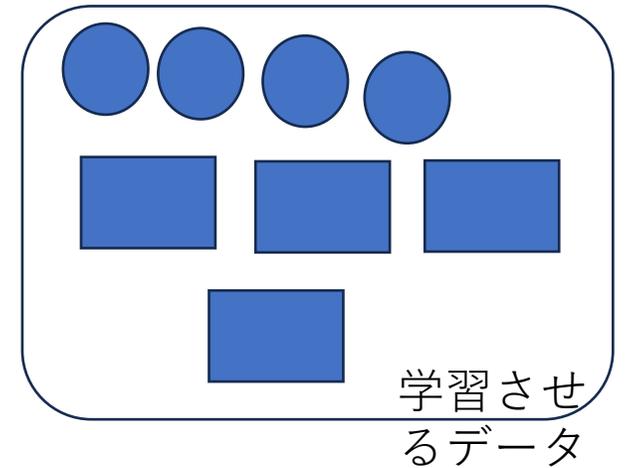
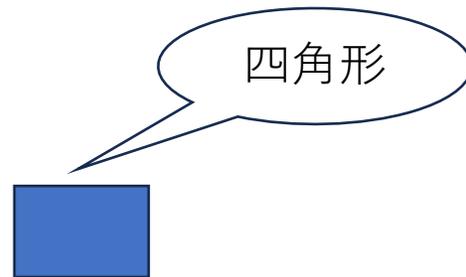
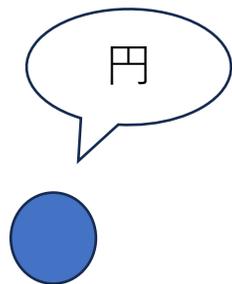


# ①コードを書く前の準備をしよう



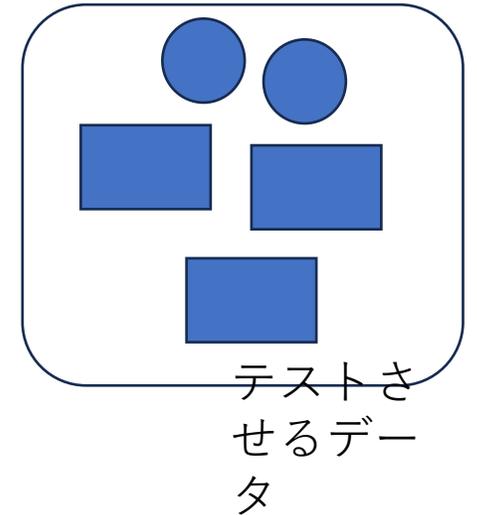
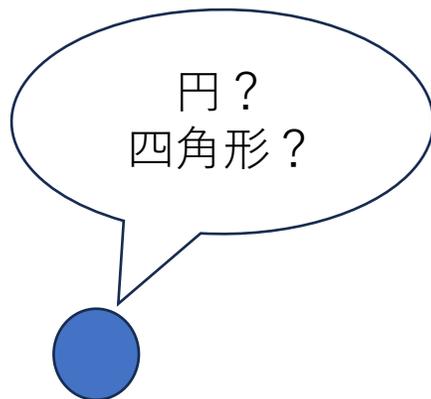
# ①コードを書く前の準備をしよう

まずは人(パソコン)が円か四角形かを判断させる。



# ①コードを書く前の準備をしよう

学習を終わらせた次は「テストさせるデータ」を使ってパソコンに判断させる



# ①コードを書く前の準備をしよう

終わったらコンピューターがどのくらい正解したかを確認する

この一連の流れを機械学習というよ！

学習させるデータを「教師データ」、テストさせるデータを「テストデータ」というよ

# 目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう**
- ③コードを書いてみよう
- ④答え合わせをしてみよう

## ②完成系を確認してみよう

このページでは完成系を見てどのようなコードを書いたらいいかをイメージしてみよう

## ②完成系を確認してみよう

ファイルを実行して、教師用データ数を入力したらコンピューターが学習して正解した確率を出力できるようにする。

```
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 100
正答率: 98.00%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 50
正答率: 94.00%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 10
正答率: 92.14%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 1
正答率: 32.89%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 5
正答率: 77.93%
(learning) PS C:\Users\小西 翔\python\python16> python .\learning.py
学習データ数を入力してください (1~149の範囲) : 120
正答率: 100.00%
```

# 目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう
- ③コードを書いてみよう
- ④答え合わせをしてみよう

## ③コードを書いてみよう

このページではどのようなコードを書いたらいいかヒントが書かれているよ

## ③コードを書いてみよう

ダウンロードしたCSVファイルを読み込もう！

## ③コードを書いてみよう

学習させる特徴量とラベルを指定しよう

暗黙の了解として特徴量を $X$ 、ラベルを $y$ と置くよ

# 説明

特徴量: データの特徴となる部分

ラベル: データが何であるかの判定部分

例)

特徴量: 「角があるかどうか」や「角が4つあるか」など

ラベル: 「円」or「四角形」

# ③コードを書いてみよう

教師用データの数を決める

# ③コードを書いてみよう

教師用データを使って学習させる

## ③コードを書いてみよう

コンピューターにテストさせて正答率の計算をさせる

# 目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう
- ③コードを書いてみよう
- ④答え合わせをしてみよう

## ④答え合わせをしてみよう

答えのコードを確認してどのようにしているか見れるよ

## ④答え合わせを試してみよう

ライブラリは以下のものを書いたよ

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.ensemble import RandomForestClassifier
4 from sklearn.metrics import accuracy_score
5
```

## ④答え合わせをしてみよう

次にファイルを選択するよ

```
5  
6 # CSVファイルの読み込み  
7 csv_file = "Iris.csv" # CSVファイルのパスを指定  
8 df = pd.read_csv(csv_file)  
9
```

## ④答え合わせを試してみよう

特徴量とラベルの設定をするよ

```
9  
10 # 特徴量とラベルを指定  
11 X = df[['Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width']] # 特徴量  
12 y = df['Species'] # ラベル  
13
```

## ④答え合わせをしてみよう

教師用データ数を決めるよ

```
13
14 # 学習データ数を指定
15 n = int(input("学習データ数を入力してください (1~149の範囲): "))
16 if n < 1 or n > 149:
17     raise ValueError("学習データ数は1~149の範囲で指定してください。")
18
```

## ④ 答え合わせを試してみよう

教師用データで学習させていくよ

```
19 # データを分割
20 X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=n, random_state=42, shuffle=True)
21
22 # モデルの作成と学習
23 model = RandomForestClassifier(random_state=42)
24 model.fit(X_train, y_train)
25
```

## ④答え合わせをしてみよう

コンピューターにテストさせて正答率を表示するよ

```
26 # テストデータで予測
27 y_pred = model.predict(X_test)
28
29 # 正答率の計算
30 accuracy = accuracy_score(y_test, y_pred) * 100
31 print(f"正答率: {accuracy:.2f}%")
32
```

# お疲れさまでした

テキストは終了です。

あとは自分なりにアレンジを付け加えていこう！