

```

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
train_me.py
143 def nn(drop, layer):
144     model = Sequential()
145
146     model.add(Conv2D(32, (3, 3),
147                     activation='relu',
148                     input_shape=input_shape))
149     model.add(Conv2D(32, (3, 3),
150                     activation='relu',
151                     input_shape=input_shape))
152     model.add(MaxPooling2D(pool_size=(2, 2)))
153     model.add(Dropout(0.2))
154
155     model.add(Conv2D(64, (3, 3),
156                     activation='relu',
157                     input_shape=input_shape))
158     model.add(Conv2D(64, (3, 3),
159                     activation='relu',
160                     input_shape=input_shape))
161     model.add(MaxPooling2D(pool_size=(2, 2)))
162     model.add(Dropout(0.2))
163
164     if layer >= 3:
165         model.add(Conv2D(128, (3, 3),
166                         activation='relu',
167                         input_shape=input_shape))
168         model.add(Conv2D(128, (3, 3),
169                         activation='relu',
170                         input_shape=input_shape))
171         model.add(MaxPooling2D(pool_size=(2, 2)))
172         model.add(Dropout(0.2))
173
174     if layer >= 4:
175         model.add(Conv2D(256, (3, 3),
176                         activation='relu',
177                         input_shape=input_shape))
178         model.add(Conv2D(256, (3, 3),
179                         activation='relu',
180                         input_shape=input_shape))
181         model.add(MaxPooling2D(pool_size=(2, 2)))
182         model.add(Dropout(0.2))
183
184     model.add(Flatten())
185     model.add(Dense(units=64, activation='relu'))
186     model.add(Dropout(drop))
187     model.add(Dense(units=num_classes, activation='softmax'))
188     # Summary
189     model.summary()
190     model.compile(loss=keras.losses.categorical_crossentropy,
191                  optimizer=keras.optimizers.Adam(),
192                  metrics=['accuracy'])
193
194     file_path = model_dir + str(model_num) + '_' + '{epoch:02d}.h5'
195     model_checkpoint = ModelCheckpoint(filepath=file_path,
196                                     monitor='val_loss',
197                                     verbose=1, save_best_only=False,
198                                     save_weights_only=False)
199
200     model.fit(x_train, y_train,
201             batch_size=batch_size,
202             epochs=epoch,
203             verbose=1,
204             validation_data=(x_test, y_test),
205             callbacks=[model_checkpoint, cb_my])
206
207     drop_range = 6
208     model_num = 0
209
210     acc_array = []
211     epoch_array = []
212     last_save = 0
213
214     for layer in range(2, 5):
215         for i in range(drop_range):
216             drop = i * 0.1
217             nn(drop, layer)
218             model_num += 1
219

```

1層目

畳み込み層 ×2  
プーリング層 ×1  
Dropout(0.2)

2層目

畳み込み層 ×2  
プーリング層 ×1  
Dropout(0.2)

3層目 layerが3以上の時実行

畳み込み層 ×2  
プーリング層 ×1  
Dropout(0.2)

4層目 layerが4以上の時実行

畳み込み層 ×2  
プーリング層 ×1  
Dropout(0.2)

5層目

全結合層 ×2  
平坦化 ×1  
Dropout(drop)

ニューラルネットワークを  
パラメータを変化させながら  
実行

SCT科目

AI研究入門

山口県立徳山高等学校

山口県立徳山高等学校  
TOKUYAMA Highschool Yamaguchi Pref. JAPAN

山口県立徳山高等学校  
SuperScienceHighschool

人工知能

AI

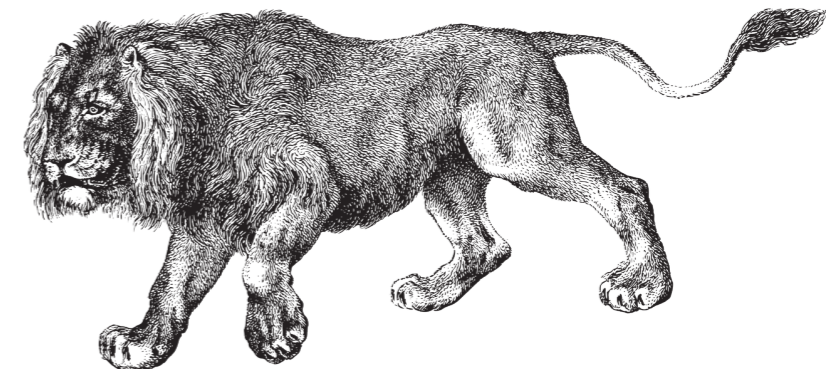
研究入門

Deep Learning Neural Network

Tensorflow OpenCV

AI and Data Science

By Python



TOKUYAMA Highschool Yamaguchi Pref. JAPAN

### 新たな体験をキミへ 電脳空間と人工知能と脳波とポクと

VR空間 AI 脳波 EEG

VR空間 VRと血圧に連動する健診体験 血圧測定器とVR空間の連携

人工知能 AI AIと脳波で学習効率を上げる AIと脳波で学習効率を上げる

脳波 EEG AIと脳波で学習効率を上げる AIと脳波で学習効率を上げる

研究内容

今後の予定

### 視覚障がい者補助 AIスマート口杖

約3人に1人

python powered

50cm未満の障害物を検知可能な口杖にします

本体

使用方法

結果・今後

約4000枚の画像

94%の検出率

### 2つのAIを用いた 打音による検査システムの開発

今、目の検査が

開発したシステム

検査モデルの構築とAI

2つのAIによるマッピング

検出結果

AIによる検査

AIによる検査

### AIによる検査システム

検査モデルの構築とAI

2つのAIによるマッピング

検出結果

AIによる検査

AIによる検査

視覚効果を考えたポスターづくり

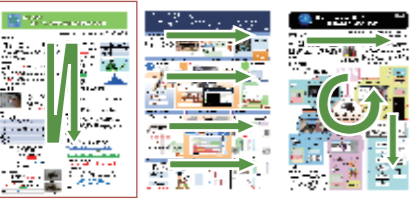
カラーチャート 補色を意識しよう



色の組み合わせ 情報を伝えやすい配色にしよう

<b>海</b>	<b>レトロ</b>
R 41 0 161 255	R 241 196 91 0
G 105 167 220 210	G 138 222 193 169
B 182 217 238 76	B 144 123 185 226
<b>ガーデン</b>	<b>おせち</b>
R 208 117 230 43	R 251 249 238 93
G 221 187 145 128	G 207 186 104 160
B 102 107 192 78	B 86 215 71 226
<b>サバンナ</b>	<b>北欧</b>
R 96 198 236 43	R 232 108 56 150
G 163 220 190 128	G 241 160 97 102
B 220 244 110 78	B 245 180 127 88

構成と説明の動線 読みやすい配置にしよう



ポスターは2段組が基本ですが様々な構成があり得ます。

ポスターづくりのキホン

- ポスターの下地は白、文字は黒が基本です。
- タイトル文字や項目、グラフ等の下地に軽めの色を入れて情報の交通整理(カテゴライズ)をしましょう。
- 濃い色の多用は避けるべきですが、強調箇所や矢印などに一部濃い色を使うと視線が誘導されて読みやすくなります。
- フォントの種類と大きさはできるだけ揃えて統一感を出しましょう。3種類程度が適当とされます。

アイコン 小さなワンポイントが大きな差をつける!

アイコンとは物事を簡単な絵柄で記号化したものです。アイコンが象徴する意味を考えて配置すると、ぐっと内容が分かりやすくなります。既製のアイコンを使うときは必ず利用条件(クレジット掲載等)を確認してください。



AI 研究入門 課題研究

### 開発したプログラムの一瞥

Python 学習-1st Step

Python 学習-2nd Step

Python 学習-3rd Step

Python 学習-4th Step

Python 学習-5th Step

Python 学習-6th Step

Python 学習-7th Step

Python 学習-8th Step

Python 学習-9th Step

Python 学習-10th Step



---

# A I 研究入門

山口県立徳山高等学校

---



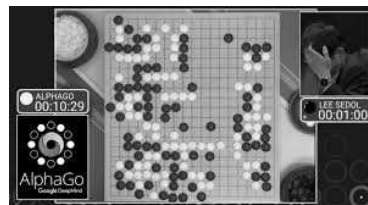
## もくじ

第1章	A I を体験しよう . . . . .	1
第2章	プログラムに触れてみよう . . . . .	5
第3章	プログラミングに挑戦しよう . . . . .	10
	・ print 関数	
	・ for 文	
	・ while 文	
	・ if 文	
	・ OpenCV(imread 関数,cvColor 関数、imwrite 関数)	
	・ OpenCV(resize 関数)	
	・ 変数、配列、構造体の読み込みと表示	
	・ web カメラの表示と活用	
第4章	A I をプログラミングしよう . . . . .	24
	・ 機械学習に必要な画像ファイルを集める	
	・ 機械学習して学習モデルを生成する	
	・ ニューラルネットワーク	
	・ web カメラを使って AI に判定させる	
	・ AI の性能を高めるために	
第5章	ワード練習 . . . . .	29
第6章	エクセル練習 . . . . .	36
資料	. . . . .	51

# 第1章 AIを体験しよう

## 1. AIとは

人工知能 (AI, Artificial Intelligence) とは、人間の知的ふるまいをコンピューターで動作するソフトウェアを用いて人工的に再現したものです。経験から学び、新たな入力や刺激に順応することで、人間が行うように柔軟に判断し、タスクを実行することを目指して開発されます。従来は細部まで人間がプログラムする必要がありましたが、2014年に脳の神経系を模した機械学習プログラムをもった Deep Learning という新しい手法の登場により、その汎用性と判定精度の高さから爆発的な普及をみせています。現在、車の自動運転、音声認識、医療画像の判定、メールの自動振り分けやネットショップのおすすめ商品のシステムに至るまで、すでに私たちの生活のあちこちで Deep Learning を用いた AI が活躍しており、一部では人間よりも正確な判断ができるようになってきました。未来の社会は、AI の利用がさらに拡大することが予想されています。

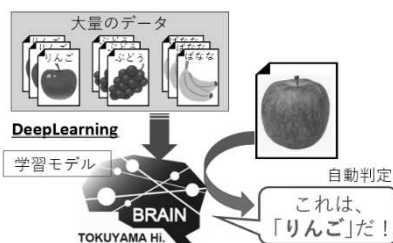


AI にルールだけを教えて後は自己学習を行い、2週間後にプロを打ち負かすレベルに成長した。Deep Learning が用いられている。

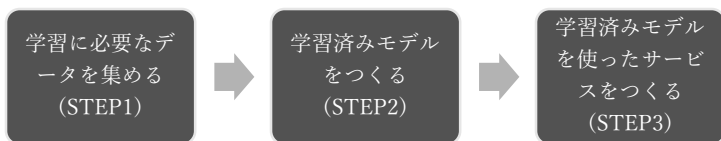
## 2. AIの開発と機械学習

私たちは机に置かれた「りんご」をどうやって認識するのでしょうか。例えば「全体的に赤い」「丸い」「上部にへこみ」等の特徴から判断しています。これまで見てきたたくさんのりんごに共通するものを特徴として記憶しておき、新たに見た物体を「りんご」として判断しているのです。

AI 開発はこれと似た手順を踏みます。まず、(STEP1)「りんご」および「他の物体」のデータを集め、(STEP2)コンピューターに「りんご」の特徴を機械学習させ、(STEP3)その学習 (学習済みモデル) を用いて新たな画像で「りんご」かどうかを判断させる、という3つの手順です。



(説明) Deep Learning とは脳の神経系に似せてつくられた学習プログラムで、大量のデータから個々の特徴を自動で見つけ出す機械学習方法の一つ。現在の AI 開発の主流。



機械学習とは、複数のデータからパターンや特徴を自動で見つけ出して判断の手順をつくる作業です。これはみなさんが試験に備えて問題演習を繰り返すことに似ていて、問題を解く毎に共通する解き方を身につけるのです。うまくいけば未知の問題に出会っても、学んだ解き方で正解することができます。つまり本番で高得点が狙える訳です。Deep Learning は機械学習の1つですが、大量のデータから特徴を見つけ出す、つまり勉強する方法が Neural Network という私たちの脳が行う処理に似せてつくられており、判定精度が高くとても秀逸です。

本講義は Deep Learning を用いて実際にオリジナルの AI をつくれるようになることを目的としています。

本講義を通じてプログラミングの基礎知識や技術を身につけ、まずは AI を実際に走らせてみましょう。そして最終項ではオリジナルの AI をつくります。AI で何をさせるか、具体的なタスクを考えながら楽しく学んでいきましょう。

### 3. AI を体験しよう

「習うより慣れろ」です。プログラムを作るには、プログラムを書き込むソフト（VSCode やメモ帳など）と、プログラムを実行するソフト（winPython Command Prompt や Jupyter など）が必要です。これらが詰め込まれたフォルダが配布した「winPython」です。

早速フォルダをのぞいてみましょう。

#### (1) 配布した「winPython」フォルダをのぞいてみよう。

「winPython」フォルダの中



「winPython」フォルダにはプログラムするために必要な様々なファイルが入っています。

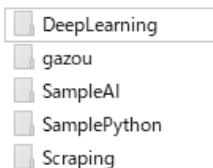
授業では次のフォルダおよびファイルをよく使います。

- ・ MyProgram フォルダ・・・プログラムやファイルを保存するフォルダ
- ・ Code.exe -ショートカット
  - ・・・プログラムを入力して保存するソフトウェア（エディタ）うまく動かないときは VSCode フォルダ内の Code.exe をダブルクリックしてください。
- ・ WinPython Command Prompt.exe
  - ・・・ Python 言語で書かれたプログラムを実行するソフトウェア

MyProgram フォルダには今から自分でつくるプログラムを保存します。この他、サンプルプログラムや画像操作に用いる写真が入っています。

- ・ DeepLearning フォルダ・・・ Chapter13 以降に使います
- ・ gazou フォルダ・・・ Chapter6 以降に使います

「winPython」中の「MyProgram」フォルダの中



ファイルやフォルダのおおよその位置や構成は分かりましたか？次はいよいよ AI を動かしてみましょう。





## 第2章 プログラムに触れてみよう

### (1) ■■ プログラムの構造理解 その1 変数の理解 ■■

- ① 「WinPython Command Prompt.exe」をダブルクリックして起動する。
- ② チェンジディレクトリを使って、起動した直後のカレントフォルダから「SamplePython」フォルダに移動する。  
(前ページを参考にしよう)
- ③ パイソンを使ってパイソンファイル「calc01.py」を実行する。  
(前ページを参考にしよう)

```
F:\winPython\MyProgram\SamplePython>python calc01.py
0
1
2
3
4
```

- ④ パイソンファイルの中身を見るソフトであるエディタ「Code.exe」をダブルクリックする。
- ⑤ 「File」→「Open File..」で「WinPython」-「MyProgram」-「SamplePython」内の「calc01.py」を開く。

```
【calc01.py】
for number in range(5):
    print(number)
```

【参考】「○○.py」は様々な方法で開いて編集することができます。

方法1. 「Code.exe」を使う (VSCoDe のアプリ版・上記)

方法2. 「メモ帳」を使う

方法3. ブラウザで <https://vscode.dev/> にアクセスする(VSCoDe の Web 版)

※方法1, 3は VSCoDe というプログラミング専用の編集アプリを用いるため、文法ミスやスペルミスを自動でチェックしてくれる便利な機能を持っています。

説明 (説明はきちんと読んで理解しよう!)

「calc01.py」はたった2行の Python 言語で書かれたプログラムです。このプログラムを実行した結果が、0,1,2,3,4 なのです。

1行目は「**for 文**」といい、命令を繰り返して実行します。繰り返す対象は、4つのスペースを空けた2行目です。「4つのスペース」で繰り返す対象を指定する方法は、Python 固有の文法です。

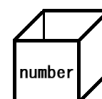
1行目の「range(5)」で[0,1,2,3,4]の数列をつくります。number は**変数**で、任意の半角文字を使うことができ、「for number in range(5)」で、「range(5)」でつくった数列を順番に変数「number」に代入を繰り返す、という命令になります。つまり繰り返し回数は5回となり、その間、number は0,1,2,3,4と変化します。

簡単に言えば、1行目「for number in range(5)」は、「i=0 から始めて1ずつ増やし、5を超えるまで繰り返す」、という意味になります。

2行目は「**print 関数**」といい、指定した値を**表示する機能**を持っています。ここでは変数 number を表示する命令とな

#### 重要用語・・・「変数」

変化しうる値を「変数」といい、任意の文字で表す。“箱”のイメージ





ります。2行目の冒頭に4つの空白があります。これは、for文が繰り返す命令の対象であることを示します。  
つまり、1,2行を実行すると、変数 number に代入された数値が順に表示されます。

### 試行

試行1 5を1000に変更してファイルを上書き保存し、実行してみよう。

試行2 次のように命令をくわえてファイルを上書き保存し、実行してみよう。("""の中身以外は全て半角)

```
for number in range(1000):  
    print(number)  
    print("にゃほにゃほ")
```

## (2) ■■■ プログラムの構造理解 その2 変数の理解 ■■■

- ① 「WinPython Command Prompt.exe」をダブルクリックして起動する。
- ② チェンジディレクトリを使って、起動した直後のカレントフォルダから「SamplePython」フォルダに移動する。
- ③ パイソンを使ってパイソンファイル「calc02.py」を実行する。→「calc02.py」を実行した結果が表示されます。
- ④ エディタ「Code.exe」をダブルクリックする。
- ⑤ 「File」→「Open File..」で「WinPython」-「MyProgram」-「SamplePython」フォルダ内の「calc02.py」を開く。

### 【calc02.py】

```
for number in range(10):  
    print(number*number)  
print("end")
```

・・・number=0 から始めて1ずつ増やし、10を超えるまで繰り返す  
・・・number\*number の計算結果を表示  
・・・end という文字列を表示

### 説明

「calc02.py」は先の「calc01.py」よりも1行増えました。「range(10)」で [0,1,2,...,9] の数列をつくり、「for number in」で変数 number に順に数列を代入する、という意味です。2行目の冒頭に4つの空白が、3行目の冒頭には空白がありません。(意味が分からない人は前ページの説明をもう一度読み直しましょう)

つまり、2行目は「for文」の対象ですが、3行目は「for文」の対象ではありません。

(1回目) number = 0 → 表示、(2回目) number = 1 → 表示、(3回目) number = 2 → 表示、  
・・・(10回目) number = 9 → 表示 して end を表示

となります。

end という表示が最後に1回だけ実行されるのは、3行目が、冒頭に4つの空白がないため for 文の対象となっていないからです。

### 【よく使う演算子】

#### ■四則計算

a + b …… 足し算    a - b …… 引き算    a \* b …… かけ算    a / b …… 割り算

(注意!) プログラミングで割り算を使うときは慎重に!

割り算のときは分母が0になると処理エラーが起き、プログラム全体がフリーズすることがある

### ■比較演算子

`a == b` . . . `a` が `b` と等しい      `a != b` . . . `a` が `b` と異なる  
`a < b` . . . `a` が `b` よりも小さい      `a <= b` . . . `a` が `b` 以下である

### 試行

- 試行1    0 から 1000 までの偶数を表示して最後に end を表示するプログラムをつくってみよう。  
試行2    print("end")の前に半角 4 つを加えるとどうなるか確認してみよう。  
試行3    プログラムを自由に変えて保存し、実行してみよう。

## (3) ■■ プログラムの構造理解 その3 配列の表示 ■■

### ① 「calc03.py」の実行

#### 【calc03.py】

```
import matplotlib.pyplot as plt    . . . 「matplotlib」ライブラリのうち「pyplot」機能を読み込み、「plt」と表す  
# data                            . . . 「#」を「コメントアウト」といい、プログラムの実行対象から外れる  
price = [100, 250, 380, 500, 700]    . . . 「price」という配列にデータを入力する  
number = [1, 2, 3, 4, 5]            . . . 「number」という配列にデータを入力する  
# plot(x,y)  
plt.plot(price, number)            . . . plot 機能を使って配列「price」と「number」をプロットする  
# title  
plt.title("price / number")        . . . グラフのタイトルを「price / number」とする  
# x-axis label  
plt.xlabel("price")                . . . x 軸のタイトルを「price」とする  
# y-axis label  
plt.ylabel("number")               . . . y 軸のタイトルを「number」とする  
# grid  
plt.grid(True)                     . . . グリッド線（格子線）を表示する  
# show the graph  
plt.show()
```

## 説明

このプログラムは、ライブラリの読み込みから始まります。ライブラリとはグラフの描画や複雑な数値計算、AI作成に必要な機械学習など、様々な機能をもつ外部のプログラム群です。Python はライブラリが充実しており、ライブラリを読み込むことである高度な技術を簡単に自分のプログラムに生かすことができます。

このプログラムで主要な部分は6行目のプロットする命令 (plt.plot) です。最後の行でグラフを表示する (plt.show()) のを忘れないようにしましょう。

また、ここでは配列が多用されています。配列とは変数が複数個横に並んだ“連なった箱”です。“箱”の中からデータを取り出したり、データを入力しておいたりと様々な命令を加えることができます。「calc01.py」や「calc02.py」でみた“range(10)”は10個の配列をつくり、順番に0,1,2…9の数列を入れる命令でした。

1個の変数に比べて配列の便利なところは、複数の変数を一度に呼び出したり、変更したりできる点です。

例えば、2,3,1,-10という4つの数字を記憶しようとする時、

変数であれば

```
number0 = 2
number1 = 3
number2 = 1
number3 = -10
```

としなければなりません、

配列であれば

```
number = [2,3,1,-10]
```

の一行で済みます。

この場合、左から2番目の“3”という数字を print 関数で表示するときは、変数であれば以下となります。

```
print(number1)          ... number1 という変数を表示
```

配列であれば以下となります。

```
print(number[1])        ... number という配列の内、左から2番目を表示
```

### 【注意】配列を用いた計算とスタックオーバーフロー

```
number = [2,3,1,-10]
```

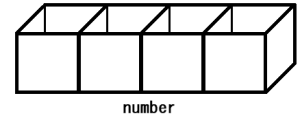
```
print(number[0]*number[3])    ... "-20"が表示される
```

```
print(number[2]*number[4])    ... 存在しない5番目の配列にアクセスしようとして"スタックオーバーフロー"が発生する。(プログラムがストップする)
```

### 重要用語・・・「配列」

変化しうる値をさらに複数をもとめたものを「配列」といい、任意の文字で表す。“連なった箱”のイメージ

箱には左から0,1,2…と数字で番号が振られる。



## 試行

試行1 数値を様々に変えてグラフを変化させてみよう。

試行2 縦軸と横軸を入れ替えてみよう。グラフのタイトル、x軸のタイトル、y軸のタイトルも変更しよう。

#### (4) ■■■ プログラムの構造理解 その4 配列へのデータ追加 ■■■

① 「calc04.py」の実行 (「calc03.py」と同じ箇所はコメントしていません)

```

【calc04.py】
import matplotlib.pyplot as plt
# data
x_value = []         ・・・x_value という名前の空 (から) の配列をつくる
y_value = []         ・・・y_value という名前の空 (から) の配列をつくる
for i in range(100): ・・・for 文。変数 i = 0 から1つ増やしながら 100 回実行する
    y_value.append(*i) ・・・配列 y_value に i×i の値を追加する
    x_value.append(i)  ・・・配列 x_value に i の値を追加する
# plot(x,y)
plt.plot(x_value, y_value)
(以降は calc03.py と同じ)
```

#### 説明

前の「calc03.py」と比べると何が違うでしょうか。まず目につくのが[]ですね。これは何もデータが入っていない配列を表します。つまり、3、4行目で空(から)の配列をつくって、後でデータを追加します。

5行目はfor文です。「calc01.py」で使ったのを覚えていますか?for文は繰り返し実行するプログラムです。

6、7行目で空だった配列y\_valueおよびx\_valueにデータを追加(append)しています。for文の最初の処理ではi=0ですから、y\_valueには0×0=0が、x\_valueには0がそれぞれ追加されます。for文の次の処理ではi=1ですから、y\_valueには1×1=1、x\_valueには1がそれぞれ追加されます。これを10回繰り返すと、y\_value=[0,1,4,・・・,81]、x\_value=[0,1,2,・・・,9]となりますよね。つまり、for文と計算によって配列のデータを生成することができます。

9行目で配列をプロットします。これ以降は「calc03.py」と同じです。

#### 配列にデータを追加 配列名.append(追加するデータ)

```

【 pyplot でよく使う機能 】
plt.plot()・・・折れ線グラフ   plt.scatter()・・・散布図   plt.bar()・・・縦棒グラフ
plt.xlim(0,100)・・・x軸の範囲を0~100に固定   plt.ylim(-10,10)・・・y軸の範囲を-10から+10に固定
plt.grid()・・・グリッド線を表示
```

#### 試行

- 試行1 数値を様々に変えてグラフを変化させてみましょう。
- 試行2  $y = -2x^2 + 4x + 3$  のグラフをつくりましょう。その際、グラフの形がよくわかるように、軸の範囲を指定してみましょう。
- 試行3 pyplotに表示されるボタンを使ってつくったグラフ画像をSamplePythonのフォルダに保存してみましょう。

### 重要文法・・・【 for 文の記述 】

for i in range(100): 変数 i = 0 から始めて 1 ずつ増やし、100 を超えるまで繰り返す

for i in range(5, 10): 変数 i = 5 から始めて 1 ずつ増やし、10 を超えるまで繰り返す

for i in range(-10, 10, 3): 変数 i = -10 から始めて 3 ずつ増やし、10 を超えるまで繰り返す

## 第3章 プログラミングに挑戦しよう

いよいよ自分でプログラムを作成していきます。最終目標は個々で考えたオリジナル AI を作成することです。Chapter の順番にスキルを身につけていきましょう。分からないことは人に相談しても構いませんが、必ず自分で作業し、理解しながら先に進めてください。特に後半の Chapter ではそれまでの理解が求められます。全てを人頼みになる状況にならないためにも、プログラミングでは以下を守ってください。

### ■■■■ 【プログラミングで必ず守ること】 ■■■■

- ・ 必ず自分で操作すること。また他人のプログラムをつくらないこと。
- ・ 早さよりも「なぜそう動くのか」をきちんと理解しながら進めること。
- ・ 指定されたファイル名で保存すること。 例) test10.py ファイル名はすべて半角です
- ・ 「プログラムが動かない！」ときはまず以下を疑うこと
  - ① 全角が混じっている（全角の空白が入っていると見えません！）
  - ② 数字やスペルが間違っている
  - ③ " , ; . が間違っている（1個だけ“が全角、なんてややこしい間違いもよくあります）
  - ④ 実行するファイル名やフォルダが違う



## Chapter1 print 関数を使う

print 関数 … 文字や数値を表示する

プログラム ・ 保存ファイル名 <code>test10.py</code>	説明
<code>print("hello world!")</code>	print 関数は””に挟まれた文字や数値を表示する

### 【実行結果】

hello world !

### 【説明】

print 関数は” ” で囲まれた中身をそのまま表示します。通常ソースコードは半角文字のみで全角文字は使えませんが、” ” 内は例外的に全角文字を使用することができます。””自体は半角です。

**情報の表示①** `print("hello world!")` ※例外的に””内は全角文字を使用することができる

【課題】 「私の名前は〇〇です」と表示させてみよう。(〇〇はあなたの氏名) `test11.py`

### 「 hello world! 」

hello world! を表示するプログラムは、最初  
に作るプログラムです。「どれくらいプログラ  
ミングできるの？」の質問に対して「全然。  
hello world レベルだよ」と答えると、初心者  
だけどプログラムを少し動かした経験がある  
よ、という意味になります。



## Chapter2 変数と print 関数を使う

print 関数 … 文字や数値を表示する

プログラム ・ 保存ファイル名 test20.py	説明
bunsho = "hello world!" print(bunsho)	変数 bunsho に"hello world!"という文字を代入する print 関数は""に挟まれた文字や数値を表示する

### 【実行結果】

hellow world !

### 説明 print 関数と変数

実行結果は Chapter1 と同じです。bunsho という変数に文字を代入してから print 関数で表示しています。bunsho は別の文字に変えてもかまいません。Chapter1 と比べて print() に "" が無いのは、bunsho が変数だからです。

### 情報の表示② print(変数や配列、構造体)

例) 変数を使用したプログラム例

なぜ【実行結果】になるのかを理解しましょう。

```
a = 2
```

```
print(a)
```

```
a = 100
```

```
print(a)
```

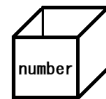
### 【実行結果】

```
2
```

```
100
```

### 重要用語…「変数」

変化する値を「変数」とい  
い、任意の文字で表す。「箱」の  
イメージ



【課題1】3行目に print("bunsho") を加えるとどうなるか確認しましょう。test21.py

【課題2】test20.py の1行目を bunsho = "hello world!" + "こんにちは" に変更しましょう。test22.py

【課題3】test20.py の bunsho を適当な文字に変えて同じ実行結果にしましょう。test23.py



## Chapter3 for文を使う

for文 … 条件を満たす間、処理を繰り返す

プログラム・保存ファイル名 test30.py	説明
<pre>for i in range(5):     print(i)</pre>	変数 i = 0 から始めて1ずつ増やし、5 を超えるまで繰り返す 変数 i を表示する

: を忘れずに!

print の前に半角の空白を 4 つ分いれます

### 【実行結果】

```
0
1
2
3
4
```

### 重要文法・・・【for文の記述】

for i in range(100): 変数 i = 0 から始めて1ずつ増やし、100 を超えるまで繰り返す

for i in range(5, 10): 変数 i = 5 から始めて1ずつ増やし、10 を超えるまで繰り返す

for i in range(-10, 10, 3): 変数 i = -10 から初めて3ずつ増やし、10 を超えるまで繰り返す

### 【説明】 for文

(1行目) for文といいます。for文とは”:"と4つの空白で指示する命令を指定した条件の中で繰り返して実行します。iは変数です。指定した条件とは i in range(5)で変数 i が数列[0,1,...4]の間、という条件を意味します。

(2行目) for文が繰り返し実行する内容です。”:”以下にforよりも半角4つ分を空けて記述します。半角の空白4つ分はpythonのルールですので必ず守りましょう。全角の空白を入れないように注意しましょう。

このプログラムの実行により、最初は i=0 で print(i)を実行し、0が表示されます。次に i に1を足し合わせて print(i)を実行し、1が表示されます。これを繰り返し、i=5になった時点でfor文の条件に合わなくなるため繰り返す作業が終了します。つまり0から4までを表示するプログラムとなります。

【課題1】 -10の表示から1ずつ増やして100を表示してみよう。test31.py

【課題2】 「こんにちは!」のみを100回表示するプログラムをつくってみよう。test32.py





## Chapter4 文字列の結合

異なる型を string 型に読み替えて結合する

プログラム・保存ファイル名 test40.py	説明
<pre>for i in range(0,3):     text = "今の数字は" + str(i) + "です"     print(text)</pre>	変数 i に 0 を代入し、1 ずつ増やして 3 回繰り返す 変数 text に "今の数字は" と i と "です" を結合して代入する 変数 text を表示する

### 【実行結果】

今の数字は 0 です

今の数字は 1 です

今の数字は 2 です

**説明** 変数の型と型の読み替え【必ず理解しよう！】

このプログラムは変数の型に注意しなければなりません。変数の型とは変数の種類のことです。型には int 型（整数値）、float 型（小数を含む値）、string 型（文字列）など様々です。変数の型は、プログラム初心者が最初に理解につまずく問題です。

1 行目で変数 i は最初に整数 0 が代入されます。この作業により、変数 i は自動で「int 型」になります。

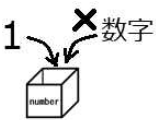
2 行目で変数 text は "今の数字は" という文字列が入るため、自動で「string 型」となります。しかし、続いて text に「int 型」である変数 i を加えることはできません。これは型がことなるからです。そこで、str(i) によって、変数 i のデータを「string 型」に読み替える工夫をしています。こうすることで、text = 「string 型」 + 「string 型に読み替えた int 型」 + 「string 型」となり、「string 型」である変数 text にデータを代入することを可能にしています。

Python は他の言語と比べると「変数の型」を意識することは少ないと言われていいます。しかし、このプログラムを例にとれば、10\*i は計算できても 10\*text は計算できません。思わぬところで失敗しないためにも「変数の型」は理解しておく必要があります。

なお、str(i) としても、例えば「int 型」の変数 i に入っている 0 という数値を「string 型」に見なしているだけで、変数 i そのものは「int 型」のままです。ややこしいですが、これは後の総合演習で使いますのでしっかりと理解してください。

### 重要用語・・・「変数の型」

変数には「型」があり、異なる型を代入することはできない。いったん「1」という数値をいれると自動で「int」型となり「あいうえお」という文字列は入れられない



### 重要用語・・・【「変数の型」の種類】

int 型・・・整数値 例) 0, -1, 2, 1000

float 型・・・小数を含む値 例) 1.34, -0.012

string 型・・・文字列 例) あ, 徳山, hohu

型の変更・・・例) bunsho = str(a)

【課題 1】 test40.py を変更して「奇数は 1 です」という表現から始めて、0 から 1000 までの奇数を全て表示してみよう。

test41.py



## Chapter5 while 文と if 文

while 文 … 条件を満たす間、処理を繰り返す

if 文 … 条件を満たすとき、処理を実行する

プログラム・保存ファイル名 <code>test50.py</code>	説明
<pre>i = 0 while True:     print(i)     i = i + 1     if i &gt; 4:         break print("終わり")</pre>	<p>変数 <code>i</code> に 0 を代入する</p> <p>【while 文】 break しない限り処理を繰り返す</p> <p>変数 <code>i</code> を表示する</p> <p>変数 <code>i</code> に <u>1</u> を追加する</p> <p>【if 文】 変数 <code>i</code> が 4 よりも大きいとき</p> <p>break を実行して while 文の処理を止める</p> <p>“終わり”と表示する</p>

### 【実行結果】

```
0
1
2
3
4
終わり
```

半角 8 つ分を空けます

### 【 `i = i + 1` 】

すでに何らかの値が入っている変数に 1 を加算する処理です。数学的にはありえない等式ですが、プログラミングでは多用されます。

この演習で大切なことは、これまでの知識を利用してきちんとプログラムを読み解くことです。最初に注意した通り、早さよりも「なぜそう動くのか」をきちんと理解しながら進めてきたか、【課題】でチェックします。

### 【説明】 while 文

while 文は条件を満たすとき繰り返し処理を行います。ここでの条件は True であり、break という命令を与えない限り延々と処理を繰り返します。for 文が指定した回数を繰り返し処理するのに対して、while 文と True の組み合わせは、その回数を指定しないのが特徴です。(ただし工夫すれば両者は同じことができます)

### 重要文法 …【条件の表現】

`A > 100` … 変数 `A` が 100 より大きい

`A >= 100` … 変数 `A` が 100 以上

`A == 100` … 変数 `A` が 100 に等しい

`A >= -3 && A <= 3` … `A` が -3 以上 3 以下


`A >= 3 || B >= 3` … `A` が 3 以上または `B` が 3 以下

### 【説明】 条件処理を行う if 条件:

if 文は条件を満たすとき処理を行います。逆に条件を満たさないときは処理されません。ここでの条件は「`i > 4`」です。変数 `i` が順当に増えていって 4 を表示した後、`i` が 5 になります。すると if 文の対象となり break が命令されるため、while 文の繰り返し処理が停止します。

【課題 1】 while 文を使って「数字は 0 です」という表現から始めて、0 から 1000 までの数字を全て表示し、最後に「以上」を表示して終了するプログラムをつくってみよう。 `test51.py`

【課題 2】 while 文を使って「奇数は 1 です」という表現から始めて、0 から 1000 までの奇数を全て表示し、最後に「以上」を表示して終了するプログラムをつくってみよう。 `test52.py`

	<p>Chapter6 画像を表示する</p> <p>OpenCVの活用1 imread関数、imshow関数、waitKey関数</p>
---	---

プログラム・保存ファイル名 test60.py	説明
<pre>import cv2 img = cv2.imread("./gazou/test.jpg") cv2.imshow("picture image",img) cv2.waitKey(0) cv2.destroyAllWindows()</pre>	<p>OpenCV ライブラリの読み込み gazou フォルダの test.jpg を構造体 img に代入 img を"picture image"という名称のウィンドウに表示 何らかのキーが入力されるのをずっと待つ 生成したウィンドウを抹消する</p>

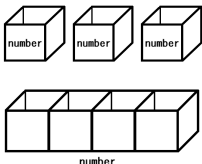
【実行結果】



何かのキーを押すと画像が消える

**重要用語 …「構造体」**

複数の変数や配列をまとめたもの



**説明** 画像と構造体

数字や文字列などは変数や配列に入れることができました。これに対して皆さんがスマホなどで見慣れている画像は小さな画素の集まりで、1つ1つが赤緑青の色情報を持ち、さらに縦横の画素数がいくつか、圧縮方法は何かなどたくさんのデータの塊です。このようなデータを構造体といいます。とはいえ、扱いはこれまでの変数や配列とあまり変わりはなく、imread関数によって画像(test.jpg)を読み込み、imgに代入しています。構造体を読み込んで代入するため、imgも自動で構造体となります。

**説明** imread関数、imshow関数、waitKey関数

OpenCVはintelが開発した画像や動画を扱う非常に多機能なライブラリです。最初にimport cv2とすることで、OpenCVの機能をプログラムで利用できるようになります。OpenCVライブラリのimread関数、imshow関数という意味で、cv2.imreadやcv2.imshowと表現します。関数の使い方にはそれぞれ決まったルールがあります。

**画像ファイルの読み込み** cv2.imread(<画像ファイル名と場所を指定(string型)>)

**画像の構造体の表示** cv2.imshow(<表示するウィンドウの名称(string型)>,<表示する画像の構造体名>)

**入力キーの待機** cv2.waitKey(<キー入力を待つ時間(ms) (int型)>>)

※0を指定するとずっと待ち続ける

【課題1】 test60.pyを変更し、test.jpgの画像を表示して1s間表示した後に自動で消えるプログラムをつくってみよう。

test61.py

【課題2】 ペイントを使って自作した画像をgazouフォルダにtest2.jpgで保存し、この画像をmy picture imageという名称のウィンドウに表示してみよう。test62.py

【課題3】 test.jpgをpicture imageという名称のウィンドウに1s間表示した後に消え、次に自動でtest2.jpgをmy picture imageという名称のウィンドウに1s間表示した後に自動で消えるプログラムをつくろう。test63.py



## Chapter7 構造体の情報

変数の表示

配列の表示

構造体の表示

プログラム・保存ファイル名 <code>test70.py</code>	説明
<pre>import cv2 img = cv2.imread("./gazou/test.jpg") i = 15 number = [10,20,-10,60] print(img.shape[0]) print(i) print(number[2])</pre>	<p>OpenCV ライブラリの読み込み</p> <p>構造体 <code>img</code> に <code>test.jpg</code> のデータを代入</p> <p>変数 <code>i</code> に 15 を代入</p> <p>配列 <code>number</code> に <code>[10,20,-10,60]</code> を代入</p> <p>構造体 <code>img</code> のうち、高さの画素数を表示</p> <p>変数 <code>i</code> を表示</p> <p>配列 <code>number</code> のうち、2 番目(左から <u>3 番目</u>)のデータを表示</p>

### 【実行結果】

```
3744
15
-10
```

#### 説明 読み込みと表示

Chapter6 が画像が表示されていたのに対して、Chapter7 はなんとも味気ない実行結果となります。しかしここでは、プログラミングでは避けて通れない代入と読み込みを学びます。なお、変数、配列、構造体ともに `i`, `number`, `img` 以外でも任意の文字列を設定できます。

変数 `i` はこれまで何度も読み込んだり表示させたりしているので省略します。

配列 `number` は `[ ]` と、で区切られた複数のデータを代入しています。第2章(4)で示したように `append` を使って後でデータを追加することもできます。左から番号が決まっており、0、1、2... となります。データの読み込みにはルールが決まっており、`number[2]` で配列番号の2のデータ(つまり左から3番目)を表します。なお、よくやってしまうミスとして、値が存在しない配列番号を指定する(例えば上記の場合、`number[10]`など)と、`out of range` というエラーが出てプログラムがストップしてしまいます。配列番号の指定には特に注意が必要です。

#### 配列の読み込み 配列名[int型]


例) `number[4]` ... 配列 `number` の配列番号4のデータ(左から5番目)を読み込む

構造体 `img` の代入の仕方は先の Chapter6 で示した通りです。読み込みには決まったルールがあります。

#### 構造体の読み込み 配列名.メソッド ※メソッドは別で定義されたルールがあります。今回は深入りしません。

例) `img.shape[0]` ... 画像構造体 `img` のうち、形 (`shape`) に関する情報の中の0番目を読みなさい、という意味になります。0番目には画像の縦の画素数が、1番目には画像の横の画素数が情報として入っています。

【課題】 `test70.py` を変更し、`test.jpg` を構成する画素のうち、上から100番目、左端から200番目の画素の色情報だけを表示するプログラムをつくりなさい。この画素の色情報は `img[100][200]` で取得できます。 `test71.py`

	<p>Chapter8 画像の操作</p> <p>OpenCV の活用2 cvtColor関数、 imwrite 関数</p>
<p>プログラム ・ 保存ファイル名 test80.py</p>	<p>説明</p>
<pre>import cv2 img = cv2.imread("./gazou/test.jpg") img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) cv2.imshow("color", img) cv2.imshow("gray", img_gray) cv2.imwrite("./gazou/test_gray.jpg", img_gray) cv2.waitKey(0) cv2.destroyAllWindows()</pre>	<p>OpenCV ライブラリの読み込み test.jpg を構造体 img に代入 <b>〔説明〕</b> 構造体 img を表示 構造体 img_gray を表示 <b>〔説明〕</b> 何らかのキーが入力されるのをずっと待つ 生成したウィンドウを抹消する</p>

【実行結果】



**説明** cvtColor 関数

構造体 img には 2 行目で読み込んだカラー画像が代入されています。OpenCV ライブラリの機能である cvtColor 関数は、画像が代入された構造体の色情報を変更します。使い方のルールは以下の通りです。

**画像構造体をもつ色情報のグレー化** cv2.cvtColor(<構造体の名称>,cv2.COLOR\_BGR2GRAY)

**画像構造体をもつ色情報の HSV 化** cv2.cvtColor(<構造体の名称>,cv2.COLOR\_BGR2HSV)

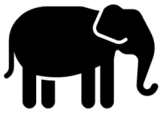
HSV とは色表現の 1 つで Hue 色相,Saturation 彩度,Value 明度を表し、画像認識等の研究分野で一般的に使われます

**説明** imwrite 関数

imwrite 関数は画像構造体をファイルにして保存する機能です。使い方のルールは以下の通りです。

**画像構造体をファイルにして保存** cv2.imwrite(<ファイル名と保存場所(string 型)>,<構造体の名称>)

【課題】 Chapter6 で自作した画像 (test2.jpg) を my picture というウィンドウ名で表示するとともに、グレー化して my picture gray というウィンドウ名で表示し、さらに test2\_gray.jpg というファイル名で保存しなさい。test81.py

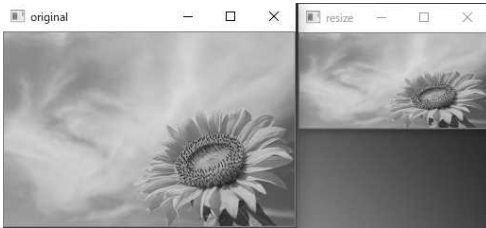


## Chapter9 画像の操作

### OpenCV の活用3 resize 関数

プログラム・保存ファイル名 test90.py	説明
<pre>import cv2 img = cv2.imread("./gazou/test.jpg") cv2.imshow("original", img) img_resize = cv2.resize(img, (200, 100)) cv2.imshow("resize", img_resize) cv2.waitKey(0) cv2.destroyAllWindows()</pre>	<p>OpenCV ライブラリの読み込み test.jpg を構造体 img に代入 構造体 img を表示 構造体 img の大きさを変更して構造体 img_resize に代入 構造体 img_resize を表示 何らかのキーが入力されるのをずっと待つ 生成したウィンドウを抹消する</p>

#### 【実行結果】



この学習で基礎編は終了です。これまで理解しながら進められているかを【課題】でチェックします。

#### 説明 resize 関数

構造体 img には 2 行目で読み込んだカラー画像が代入されています。OpenCV ライブラリの機能である resize 関数は、画像が代入された構造体の大きさを変更します。使い方のルールは以下の通りです。

#### 画像構造体がもつ大きさの変更 方法①

`cv2.resize(<構造体の名称>, (横の画素数(int 型), 縦(int 型)))`

例) `cv2.resize(img, (300, 100))`

#### 画像構造体がもつ大きさの変更 方法②

`cv2.resize(<構造体の名称>, dsize=None, fx=横の倍率(float 型), fy=縦の倍率(float 型))`

例) `cv2.resize(img, dsize=None, fx=0.3, fy=0.4)`

【課題 1】 test90.py を変更して縦横をそれぞれ 0.5 倍にして表示しなさい。 test91.py

【課題 2】 test90.py を変更して大きさを変更した画像を test\_resize.jpg というファイル名で保存しなさい。 test92.py

## Chapter1~9までの総復習

さて、これまでPythonプログラミングの基礎を学んできました。ここで、以下の課題に取り組み、これまでの復習をしましょう。



**【課題3】** test90.py に for 文を加えて縦の倍率のみが0.1倍、0.2倍、・・・4.0倍にした画像をそれぞれ0.1s秒ずつ表示しなさい。test93.py

ヒント1 この課題は for 文を使って繰り返す処理を行います。Chapter3を復習しましょう。

```
for i in range(10):
```

とすれば、変数 i=0 は1ずつ増えながら i=9 になるまで全部で10回処理を繰り返すのでしたね。

ヒント2 変数はどこに使えばよいのでしょうか。縦の倍率のみを拡大するので、

```
cv.resize(img, dsize=None, fx = 1.0, fy = float(i))
```

とすれば、i の変化に合わせて構造体 img のサイズが変更されます。float() で i で困るのは resize 関数で指定する数値は float 型だからです。i は int 型のため float() で困うことで float 型と読み替えています。

ヒント3 0.1秒ずつはどう表現したらよいでしょうか。Chapter6を復習しましょう。

```
cv2.waitKey(0)
```

とすれば、キー入力を延々と待ち、0を1000と変えると1s待つのでしたね。

以上のヒントを使って、次に続けて課題3のプログラムを完成させましょう。

```
import cv2
img = cv2.imread("./gazou/test.jpg")
cv2.imshow("original", img)
for
```

**【課題4】** test93.py を変更してウィンドウの名称が「縦倍率0.1倍」から始めて0.1sごとに「縦倍率4.0倍」にしてそれぞれの倍率を変更した画像を表示しなさい。(成功すると大量のウィンドウが表示されます)

ヒント 課題3は表示されるウィンドウは original と resize の2種類です。img は次々と倍率を変更していますが、resize という同一のウィンドウ名のため、情報の上書きが繰り返されてウィンドウのサイズがアニメーションするように変化します。これに足して課題4はウィンドウ名を変えることになるため、情報が上書きされず、大小のウィンドウが残ることになります。

ウィンドウ名を課題に指定されたように変更するにはどうすればよいでしょうか。Chapter6にある通り、imshow 関数で指定するウィンドウ名は string 型です。変数は int 型ですよ。Chapter4で学んだ異なる型の結合をヒントにウィンドウ名を変化させて課題のプログラムを完成させましょう。test94.py



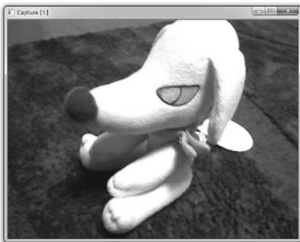
Chapter10 webカメラの表示 ※ここからはwebカメラが必要です



webカメラの表示と活用1

プログラム・保存ファイル名 test100.py	説明
<pre>import cv2 capture = cv2.VideoCapture(0) capture.set(3, 640) capture.set(4, 480) capture.set(5, 15) while True:     ret, img = capture.read()     cv2.imshow("web camera", img)     k = cv2.waitKey(1)     if k == 27:         break capture.release() cv2.destroyAllWindows()</pre>	<p>OpenCV ライブラリの読み込み 0 番目の web カメラのデータをオブジェクト capture とする 横の画素数を 640 に設定 縦の画素数を 480 に設定 1s 間あたりに取得する画像枚数を 15 枚に設定</p> <p><b>説明</b></p> <p>①オブジェクト capture の画像データを構造体 img に代入 ②構造体 img を表示 ③1ms のキー入力を待機し、入力されたキーを変数 k に代入 ④もし ESC キー(k=27)が押されたとき、 ⑤break して while 文のループを抜ける ⑥オブジェクト capture と web カメラを切り離す 全てのウィンドウを閉じる</p>

【実行結果】



【予備知識】cv2.VideoCapture(0)で0番目のwebカメラとオブジェクトcaptureを接続します。PCに複数のwebカメラがつながっている場合は、0を1や2に変えてみてください。

いよいよwebカメラの登場です。webカメラを使うと連続した静止画を手軽に得ることができます。2行目でwebカメラとオブジェクトcaptureとつなぎます。オブジェクトとは部品のイメージです。特にwhile文に注目です。

**説明** while文とwebカメラの表示

- ①の説明 構造体imgにwebカメラからの画像が代入されます。retやcapture.read()の意味は省略します。
- ②の説明 imshow関数で構造体imgを表示しています。これまで何度も使いましたね。
- ③の説明 これまでと少し異なり、1msだけ待ってその間に入力されたキーを変数kに代入します。
- ④の説明 if文です。変数kが27(ESCキーの固有番号)のとき
- ⑤の説明 オブジェクトcaptureとwebカメラを切り離します。これをしないと他のソフトウェアがwebカメラを利用できないため、webカメラを利用した後は必ずreleaseします。

【課題】test100.pyを変更してcolorというウィンドウ名に元のカラーの映像を、grayというウィンドウ名にグレーに変えた映像をそれぞれ表示しなさい。(ヒント;Chapter8) test101.py





## Chapter11 web カメラの画像保存



### web カメラの表示と活用2

プログラム・保存ファイル名 test110.py	説明
<pre>import cv2 capture = cv2.VideoCapture(0) capture.set(3, 640) capture.set(4, 480) capture.set(5, 15) while True:     ret, img = capture.read()     cv2.imwrite("./gazou/webcam.jpg", img)     cv2.imshow("web camera", img)     k = cv2.waitKey(1)     if k == 27:         break capture.release() cv2.destroyAllWindows()</pre>	<p>:</p> <p>(Chapter10 と同じ)</p> <p>:</p> <p>構造体 img を webcam.jpg という名称で保存</p> <p>:</p> <p>(Chapter10 と同じ)</p> <p>:</p>

#### 【実行結果】



**説明** web カメラの画像をファイルに保存する

先の Chapter10 にたった 1 行追加しただけです。構造体 img には画像データが入っており、Chapter8 で学んだとおり、`imwrite` 関数を使ってファイルにして保存しています。

適当なところで ESC キーを押して止めてみましょう。gazou フォルダに `webcam.jpg` という名称のファイルがあるはずです。ダブルクリックすると画像が表示されます。しかし、画像は 1 枚だけですよね。imwrite 関数は while 文の中で何度も実行されているのに、保存されているのはたった 1 枚。これはなぜでしょう。

これを解決するプログラムをつくるのが、次の Chapter12 です。

**【課題】** test110.py を変更して color というウィンドウ名にカラーの映像を、gray というウィンドウ名にグレーに変換した映像を表示し、それぞれ `webcam_color.jpg` および `webcam_gray.jpg` という名称で保存しなさい。test111.py



## Chapter12 画像の連続保存

### web カメラの表示と活用3



保存ファイル名 `test120.py`

問1 MyProgram フォルダに images という名称のフォルダをつくりなさい。

問2 前の Chapter11 における test110.py を変更して、web カメラの画像を webcam0.jpg から始まり、ESC キーが押されるまで webcam1.jpg, webcam2.jpg…と連続で保存するプログラムをつくりなさい。ただし画像を保存する場所は、MyProgram フォルダ内の images フォルダとします。

問3 問2 ができたら、さらにプログラムを変更して、縦横の画素数がそれぞれ 128 と 128 の正方形の画像にして問2 と同様に保存するプログラムをつくりなさい。

#### 【実行結果】

images フォルダの中に web カメラの映像が正方形の画像ファイルになって大量に保存される。

#### 【ヒント】

##### Chapter4 文字列の結合

例) `text = “今の数字は” + str(i) + “です”` ※text は string 型

##### Chapter8 imwrite 関数・ファイルの保存

`cv2.imwrite(<ファイル名と保存場所(string 型)>, <構造体の名称>)`

例) 構造体 `img_gray` を `gazou` フォルダに `test_gray.jpg` という名称のファイルで保存  
`cv2.imwrite("./gazou/test_gray.jpg", img_gray)`

##### Chapter9 resize 関数・画像のサイズ変更

`cv2.resize(<構造体の名称>, (横の画素数 (int 型), 縦 (int 型)))`

例) 構造体 `img` の画像のサイズを横 200、縦 100 の画素数に変更して構造体 `img_resize` に代入する。

`img_resize = cv2.resize(img, (200, 100))`




#### 説明 test120.py のプログラムと AI

test120.py が完成すると、web カメラを使って大量の画像データを取得することができます。これは AI をつくる上で強力な武器になります。

第 1 章で AI を体験しました。そのとき SampleAI フォルダにある images フォルダの中に大量の人の顔が映った画像データがあったのを覚えていますか？これは AI をつくる機械学習に必要なデータなのです。

test120.py を使って自分で画像データを用意し、images フォルダの画像と置き換えるとどうなるでしょうか。そう、アイデア次第で自分だけのオリジナル AI をつくることができます。次の Chapter ではよいよ AI の作成が始まります。

## 第4章 AI をプログラミングしよう

	<p>Chapter13 機械学習に必要な画像ファイルを集める</p> <p>&lt;AI 作成の手順&gt;</p>  <pre>graph LR; A[学習に必要なデータを集める (STEP1)] --&gt; B[学習済みモデルをつくる (STEP2)]; B --&gt; C[学習済みモデルを使ったサービスをつくる (STEP3)];</pre>
<b>課題</b>	
<p>手順1 MyProgram フォルダに次の3つのプログラムファイルをコピーする。 「train.py」および「load_images.py」・・・「images」フォルダのデータを機械学習するためのファイル 「prediction.py」・・・ 機械学習して生成する学習済みモデルで web カメラに写ったものを判定するファイル</p> <p>手順2 何を分類するのかを決めて必要であればモノを用意する。</p> <p>手順3 「test120.py」を利用して images フォルダに画像データを集める。以下の点に注意すること。</p> <ul style="list-style-type: none"><li>①種類毎にフォルダの名称を変えて分類する。例えば apple, orange, peer の写真であれば images フォルダの中は右のようになる。</li><li>②1つのフォルダに100枚程度の画像を保存する。</li><li>③様々な角度から撮った画像を選ぶ。ファイルを選んで「delete」キーを押すと削除可能。</li></ul> <div style="text-align: right;"></div>	

### 【実行結果】

images フォルダの中に分類したフォルダがあり、中に100枚程度の画像ファイルが入っている。

### 説明 「AI はデータが全て」 AI とデータサイエンス

「AI はデータが全て」といわれます。特徴を機械学習するためのデータはAIの精度を大きく左右します。そのためには、「大量」かつ「質の高い」画像が必要です。ここでいう「質の高い」とはどういうことでしょうか。

例えば人物を特定するAIをつくったとしましょう。正面の写真しか学習しなかった場合、できあがったAIは正面なら大丈夫でしょうが、少しでも斜めを向くととたんに精度が落ちます。また、光の当たり具合や着ている服によっても精度が大きく変わります。つまり、ここでの「質が高い」とは、例えば同一人物であっても正面だけでなく、多様な画像が揃っているということになります。

しかし、どうしても画像を増やせない状況もあります。この場合、例えば明るさを変更したり、少し回転してみたりなど画像を操作して増やせば、ある程度は多様な画像を「水増し」することができ、AIの精度向上が期待できます。「水増し」はAI作成で一般的に使われる手法です。

このように目的に対して様々なアプローチを検討し、より適切な情報の処理手法を模索する科学をデータサイエンスといい、AIの研究開発に欠かせない学問分野となっています。



## Chapter14 機械学習して学習済みモデルを生成する

### <AI 作成の手順>



### 課題

手順1 「train.py」を VSCode で開く。

Code.exe を実行→「File」-「Open File..」で「WinPython」-「MyProgram」-「train.py」

手順2 「train.py」プログラム中の num\_classes の数を「images」フォルダ内のフォルダ数にする。

例) images フォルダが右であれば num\_classes=3 とする。

手順3 プログラム中の epoch の数を 2 にしてファイルを保存する。

epoch = 2



手順4 「WinPython Command Prompt.exe」を起動して「train.py」を実行する。(時間がかかります)

```
python train.py
```

### 【実行結果】

学習済みモデル「mymodel.h5」ファイルと「whoiswho.txt」が生成

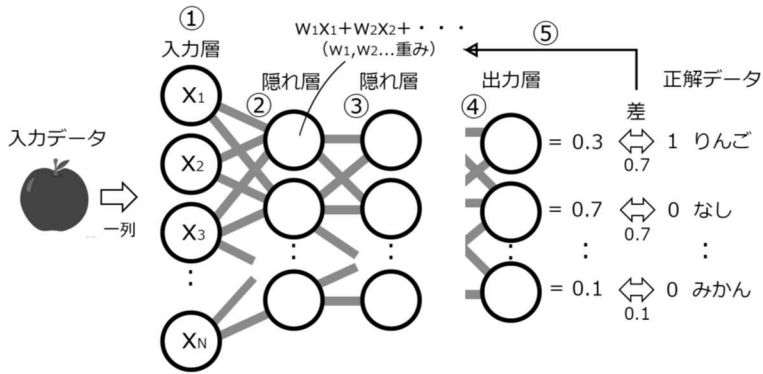
機械学習しているときに表示される val\_acc の値が AI の正答率を表す

### 説明 ニューラルネットワーク

機械学習とは大量のデータから特徴を自動で抽出する作業です。例えばリンゴと梨の場合、丸いという点は共通しているため特徴になりませんが、色の違いに気づけば見分けがつかうかもしれません。リンゴと消しゴムであれば丸いと四角は特徴になりうるでしょう。このように、データによって特徴は異なり、これを自動で探る作業が機械学習です。

ではどうやって特徴を見つけ出すのでしょうか。これには様々な手法が提案されていますが、「AI 研究入門」で用いているニューラルネットワークについて簡単に説明しましょう。

- ① まずある画像を構成する数値情報 (RGB の数値) をすべて一列に並べます。これを入力層といいます。
- ② 数値に何らかの重みを付けて足し合わせるなどして新たな数値の 1 列をつくり、これを隠れ層といいます。
- ③ 隠れ層では得られた数値を一定値以上で増加する関数に代入するなどして新たな数値をつくり、②と同様に、何らかの重みを付けて足し合わせるなどして新たな数値の 1 列をつくり、2 段目の隠れ層をつくり、
- ④ これを繰り返し、最終的には入力した画像が何なのかを確率で表現するようにします。これを出力層といいます。このとき、正解の間に差ができます。
- ⑤ 正解との間の差が小さくなるように層から層に値を受け渡す際の重みを変更します。最初はデータによる影響がないため差が大きいです、これを何度も繰り返すと出力層と正解の差が小さくなっていきます。つまり、データの影響を受けて重みが最適化されるわけです。ここで未知の画像を入力層にすると、最適化された重みで隠れ層を通過し、出力層で得られる確率により、未知の画像を正しく判定することが期待されます。



ニューラルネットワークによる機械学習のしくみ

この層と層の情報の伝搬が、生物における神経シナプスや情報伝達物質を介した伝搬と似ているため、ニューラルネットワークと呼ばれます。この機械学習の手法が優れている点は、画像だけでなく、音声や株価や気象情報など、およそ数値で表されるデータ全般を扱えること、層の数や層間の計算等に自由度があり、データの特性に応じてネットワークを変更できることなど数多くあります。

Google社はこうした機能をさらに使いやすくパッケージ化してtensorflowというライブラリで公開しました。「train.py」はこのtensorflowを使用してニューラルネットワークを利用した機械学習を実現しています。

Chapter15 web カメラを使ってAIに判定させる

<AI 作成の手順>

学習に必要なデータを集める (STEP1)

➔

学習済みモデルをつくる (STEP2)

➔

学習済みモデルを使ったサービスをつくる (STEP3)

**課題**

手順1 「prediction.py」をVSCodeで開く。  
 Code.exe を実行→「File」-「Open File..」で「WinPython」-「MyProgram」-「prediction.py」

手順2 「prediction.py」プログラム中の num\_classes の数を「images」フォルダ内のフォルダ数にする。  
 例) images フォルダが右であれば num\_classes=3 とする。

手順3 「WinPython Command Prompt.exe」を起動して「prediction.py」を実行する。

```
python prediction.py
```

apple  
 orange  
 peer

説明



うまく判定できましたか？

# AIの性能を高めるために

自分でつくったAIは満足のいくものでしたか？

はじめてつくったモノがうまく動いたときは、うれしさとともに様々な気づきや問題があるものです。プログラミングの世界にも、ものづくりの原動力となる感動や、質を向上させる工夫の積み重ねが存在します。問題を1つ1つ解決し、よりよいモノにしましょう。

一般に、AIの性能を高めるために見直すべき点は次の大きな3つです。

## 1. 学習ファイル(画像)の見直し

学習のために用意する画像ファイルは結果に大きく影響します。試験対策でいうと解いた問題の種類や質、量に相当します。目的に合った画像を準備すべきです。

- ① フォルダ名やフォルダの種類は適切か。フォルダ名と中に入れる画像は一致しているか。
- ② 画像の質は十分か。余計なモノは映っていないか。
- ③ 画像の数は十分か。

## 2. 学習プログラムの見直し

学習する方法も結果に大きく影響します。試験対策でいうと勉強方法に相当します。学習の基本はドリル、つまり繰り返し回数が重要です。しかしがむしゃらに繰り返し学習の回数を増やしても結果がよくなるとは限らず、むしろ下がっていく傾向があります。また、脳の思考回路に相当するニューラルネットワークを工夫すると正答率が劇的に向上することもあり、学習プログラムの特性はかなり「人間的」です。

- ① 「train.py」における epoch 数を増やす。
  - ② 「train.py」におけるニューラルネットワークを最適化する。
- ②は「train.py」のニューラルネットワークの部分を理解する必要があり、Pythonのさらなる勉強が必要です。

## 3. 組み込みプログラムの見直し

学習済みモデルの使い方も結果に大きく影響します。試験対策でいうと本番における頭の使い方に相当します。まず、問題文がどこなのか、何を問われているのかを把握する必要があります。また、正答の候補が複数あったとき、どう解答すればよいかを判断する必要があります。

- ① 人物判定の場合、「prediction.py」で「顔」のみを切り出すプログラムを付け加える。
  - ② 「prediction.py」で判定確率が最も高いものを正答とする。
- ①、②ともに「prediction.py」をきっちり理解する必要があり、Python全般およびDeep Learning、OpenCV等のさらなる勉強が必要です。





ワード編

エクセル編

資料



## 第5章 「ワード練習」



「ワード (Word)」は文書を作成するソフトです。P Cの基礎技術と文書作成のルールを身につけましょう。なお、作成中はできるだけブラインドタッチに努めてください。

### 【授業におけるワードの設定】

余白・・・・・・・・標準  
1行の文字数・・・・40字  
1ページの行数・・・・36行  
ルーラー、グリッド線・・・・表示

### 【保存ファイル名】

保存するファイル名は以下のルールに従ってください。

00 徳山太郎ワード練習①.docx (出席番号は半角)

### 【保存場所】

指示された場所

### 【作成上の注意点】

- ・文字寄せには「左揃え」「中央揃え」「右揃え」および「インデント」を使い、スペース（空白）を用いては文字を揃えてはいけません。
- ・与えられた文書をよく読み、数字の全角、半角の違いに注意すること
- ・基本的にフォントの種類は全て「MS 明朝」とする。
- ・フォントのサイズは、タイトルが「12」、他は「10.5」とする。

### 【ビジネス文書の「型」】

国を問わず、上司や取引先、組織のメンバー等に渡すことを目的としたビジネス文書は決まった「型」があります。「型」を知り、「型」になれましょう。

フォーマルな文書は、次の情報を順番に記述します。

- ①日付を**右寄せ** (通常、文書の発信日にする)
- ②情報発信**先** を**左寄せ** (「〇〇 様」、「〇〇 各位」と表現する)
- ③情報発信**元** を**右寄せ** (組織名、役職、氏名を示す。2行になることが多い)
- ④タイトル を中央揃え (「ご案内」「お知らせ」「お願い」を入れてどういう内容かをタイトルで示すと分かりやすい)
- ⑤拝啓 — 敬具 (「拝啓」は左寄せ改行、「敬具」は内容の末尾に改行して右寄せ)
- ⑥時候の挨拶 (文書発行日によってある程度決まっている)
- ⑦内容 (何らかの情報を示す際は「下記の通り」を入れる)
- ⑧記 — 以上 (「記」は中央揃え、「以上」は情報の末尾に改行して右寄せ)



# ワード練習①

2024年5月1日

要の会 各位

全角スペース

2. 発信先・左揃え

4. タイトル・中央揃え

1. 日付・右揃え

要の会事務局  
幹事 徳山太郎

3. 発信元・右揃え  
「役職等」「氏名」の順

全角スペース

初夏の交流会について（御案内）

時候の挨拶

拝啓

新緑の候、ますますご健勝のこととお慶び申し上げます。平素は、会の活動にご理解ご協力をいただき、厚く御礼を申し上げます。

この度、下記の通り「初夏の交流会」を開催する運びとなりました。会員の皆様お誘い合わせの上、ご参加いただけますようお願い申し上げます。

「下記の通り」→「記」

中央揃え

記

「拝啓」→「敬具」

敬具

日 時 2020年7月1日（金）19:00～  
場 所 新宿ABCホテル（蘭の間）  
参加費 10,000円  
連絡先 事務局担当 徳山（0123-45-6789）

インデントで揃える  
スペースを使わない！

「記」→「以上」

以上

通常、数字は一桁は全角、二桁以上は半角にする。  
例) 6月12日  
全角 半角  
ただし、ビジネス文書では読みやすさが優先されるため、全て全角になることもある。

## ワード練習②

令和2年6月10日

営業担当 各位

タイトルの後に()で文書内容の性質を示すと分かりやすい  
例) (御案内),(お知らせ),(お願い),(提案)など

経理部 山田一郎

### 消費税変更に伴う留意点について (注意喚起)

各営業担当者は、消費税の変更に伴い、下記の通りに価格改訂に抜けがないかの最終確認に留意して、見積書の作成、請求業務等に誤りのないよう担当への周知徹底をしていただきますようお願いいたします。

なお、ご不明な点がございましたら経理部 山田 (内線 123) までお問合せください。

### 記

#### 1. 留意点

- (1) 値札変更及び代理店への新税率による価格改訂の徹底
- (2) カタログ価格改訂の確認

#### 2. 表記のしかたについて

(例)

変 更 前	変 更 後
¥10,800 (税込)	¥11,000 (税込)

通常、表を差し込むときは、両端を文章よりも内側にする。

以上

令和4年10月10日

販売店担当者様

株式会社徳山商事  
厚生部 山田一郎

秋の商談会参加について（お知らせ）

拝啓

秋晴の候、各販売店担当者様におかれましては、ますますご清栄のこととお喜び申し上げます。

この度、下記の通り「秋の商談会」を開催する運びとなりました。つきましては、地図をご参照の上、当日お時間までにご集合の程、よろしくお願い致します。

敬具

記

1. 集合日時 令和4年10月29日 9:30
2. 集合場所 周南市役所前バス停付近
3. 備考 弊社が用意したバスにご乗車ください



以上

「ペイント」は万能ツール  
文書に図を入れると分かりやすくなる。  
Windows 標準の「ペイント」アプリは簡単な図を描いたり、図表をコピーまたは貼り付けたりするのに便利である。すぐに使える場所に置いておこう。

## ワード練習④

問題 これまでの学習をもとにして、以下の情報を元にビジネス文書を作成しなさい

- 文書発信日 令和4年5月11日
- 情報発信先 情報企画部社員
- 情報発信元 情報企画部 経理担当 徳山華子
- タイトル 歓送迎会の集金について
- 内容
  - ・ 歓送迎会の実施に伴って一人¥3,000 集める
  - ・ 4月30日までに徳山華子に封筒に記名して提出

# FAX

宛先

送信元

鈴木商事

山口県立徳山高校

経理担当 徳山花子 様

理数科主任 徳山太郎

Tel. 0123-45-6789

Tel. 0123-45-0000

Fax 0123-45-6787

Fax 0123-45-0001

内容 文房具の発注数の変更について（ご確認）

大変お世話様になっております。

電話でお知らせしたとおり、先日発注した物理の参考書の冊数に変更が生じたので、ご確認をお願いいたします。

変更前

→

変更後

45冊

47冊

## ワード練習⑥ “ポスター作成”

問題 文字の大きさ、色、写真やそれらの配置などに工夫をこらしながら、ポスターを作成しなさい。

1年1組1番 徳山 太郎

# 寄生虫博物館

に行ってきました！



場所 東京都目黒区

内容 寄生虫をテーマにした博物館

説明 珍しい寄生虫がずらり並んでいます。中でも圧巻なのが5m のサナダムシの標本です。これが体内にいたなんて想像したくもありません。修学旅行でぜひ立ち寄ってください。

## 第6章 エクセル練習



「エクセル (Excel)」は表を使って集計や計算をしたり、グラフを作成したりするソフトです。基礎技術を身につけるとともに、表やグラフを分かりやすく表現することに努めましょう。

### 【エクセルの基礎知識】



#### ■ 「セル」と「セル番号」

エクセルの枠を「セル」※①といい、1つ1つに場所を表す「セル番号」※②があります。セル番号は横軸はアルファベット、縦軸は数字で表します。

#### ■ 「セル」の大きさ

「セル」の大きさは変えることができます。たとえば横軸の間※③にカーソルを移動すると横幅を変更するマークが現れて調整できます。また「セルを結合して中央揃え」※④を使うと複数のセルを1つに結合することができます。

#### ■ 「セル」を線で囲む

「セル」は周囲を線で囲むことができます。囲みたいセルを選択して格子のアイコン※⑤をクリックします。線の種類や太さを変更することもできます。

### 【保存ファイル名】

保存するファイル名は以下のルールに従ってください。

00徳山太郎エクセル練習①.xlsx (出席番号は半角)

### 【保存場所】

指定された場所

### 【作成上の注意点】

- ・ グラフ以外のフォントサイズは「11」ポイントとする。
- ・ 必要であれば表のタイトルは「セルを結合して中央揃え」を使う。
- ・ 数値は全て半角とする。エクセルはPCに計算させることが多い。全角の数字が混じると思わぬエラーが出ることもある。原則として「エクセルは半角」を守ろう。

## エクセル練習① 四則計算

問 網がけのセルに適切な数式を入力して計算結果を表示せよ。

- ・フォントは全て 11ポイント
- ・「三角形面積」「台形面積」は2つのセルを「セルを結合して中央揃え」で結合する
- ・表の通りに太枠を設定する

### ■計算式①

100	200	加算	
50	20	減算	
40	15	乗算	
20	5	除算	
10	100	減算	

### ■計算式②

底辺	20	三角形面積	
高さ	10	台形面積	
上底	20		
下底	30		

### ■計算式③

x	0	1	2	3	4	5	6
$y=2x$							
$y=x*x$							
$y=4x+2$							
$y=x*x+1$							

メモ



## エクセル練習② 四則計算とSUM関数

問 網がけのセルに適切な数式を入力して計算結果を表示せよ。

- ・フォントは全て11ポイント
- ・¥マークおよびカンマ表示は右クリック→「セルの書式設定」で設定
- ・タイトルは「セルを結合して中央揃え」を利用
- ・消費税は10%として計算する
- ・総合計は「SUM」を用いる
- ・計算式2の売値は計算式1の売値を参照する
- ・表の通りに太枠を設定する

### ■計算式1

お中元のリスト

商品No.	商品名	価格	消費税	売値
A001	サーモン	¥5,000		
A002	コーヒーギフト	¥4,000		
A003	フルーツジュース	¥2,500		
A004	サラダ油セット	¥2,000		
A005	乳製品詰め合わせ	¥3,400		

### ■計算式2

売り上げリスト

商品No.	商品名	売値	販売数	合計
A001	サーモン		8	
A002	コーヒーギフト		11	
A003	フルーツジュース		15	
A004	サラダ油セット		4	
A005	乳製品詰め合わせ		8	
			総合計	

メモ

## エクセル練習③ 四則計算とAVERAGE、オートフィル

問 網がけのセルに適切な数式を入力して計算結果を表示せよ。

- ・フォントは全て 11ポイント
- ・タイトルは「セルを結合して中央揃え」を利用
- ・総合計は「SUM」、平均点、クラス平均点は「AVERAGE」を用いる
- ・オートフィルを用いて全員分を計算する。

オートフィルは、最上位のセルに計算式を入れた後、セルの右下に表示される「+」を下に引く

- ・平均点、クラス平均点は小数第一位まで表示する
- ・計算式 2 のケースは 1 ケース12本入りを表す
- ・表の通りに太枠を設定

### ■計算式 1

2 学期中間テストの結果

氏名	国語	数学	英語	社会	理科	総合点	平均点
A	84	64	84	97	69		
B	61	77	51	88	78		
C	78	51	58	79	87		
D	53	64	88	71	69		
E	78	84	88	81	80		
クラス平均点							

### ■計算式 2

売り上げ

商品名	単価	ケース	バラ	本数	金額
ビール	¥350	11	15		
ワイン	¥1,500	5	20		
サイダー	¥150	18	0		
ジュース	¥130	31	21		
			総合計		

メモ

## エクセル練習④ オートフィルと絶対参照

問 網がけのセルに適切な数式を入力して計算結果を表示せよ。

- ・ フォントは全て 11ポイント
- ・ タイトルは「セルを結合して中央揃え」を利用
- ・ 「平均」は「AVERAGE」を用いてオートフィルを使って全員を計算する。
- ・ 「クラス平均との差」は自己の平均と「クラス平均」の差を表す。「クラス平均」を絶対参照した数式を入れた後に、オートフィルで全員分を計算する。

絶対参照は「=C14-\$C\$19」など、オートフィルで変更されたくないセル番号に「\$」をつける。

- ・ 計算式2は、「xの値」を絶対参照してオートフィルでyの値を計算する。
- ・ 表の通りに太枠を設定する。

### ■ 計算式 1

腕立て伏せの回数

氏名	1回目	2回目	平均	クラス平均との差
A	21	19		
B	61	55		
C	14	10		
D	28	4		
E	11	0		
		クラス平均		

### ■ 計算式 2

xの値	
$y=\sqrt{x}$	
$y=1/x$	
$y=x$	
$y=x*x$	
$y=x*x*x$	

メモ

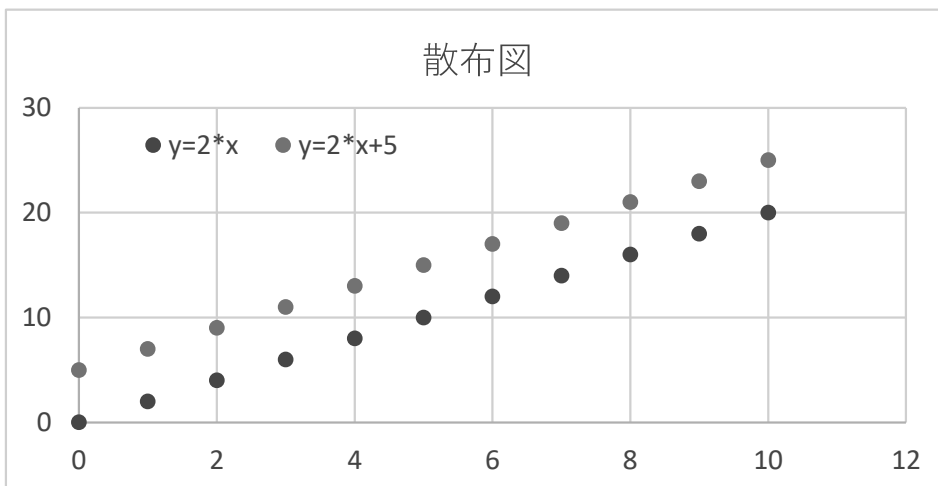
## エクセル練習⑤ 散布図

問 空欄を埋めてそれぞれ散布図を描け

- ・グラフは散布図
- ・グラフのタイトルは14ポイント
- ・凡例や軸の数値は全て11ポイント

### ■計算式①

x	0	1	2	3	4	5	6	7	8	9	10
$y=2*x$											
$y=2*x+5$											



### ■計算式②

x	0	1	2	3	4	5	6	7	8	9	10
$y=x*x$											
$y=\sqrt{x}$											

### ■計算式③

x	-10	-8	-6	-4	-2	0	2	4	6	8	10
$y=-2*x$											
$y=x*x$											

メモ

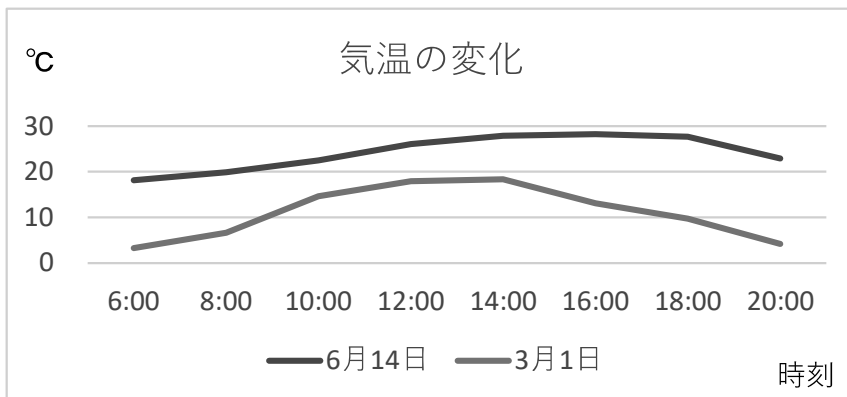
## エクセル練習⑥ 折れ線グラフと棒グラフ

問 折れ線グラフと棒グラフを描け

- ・「気温の変化」のグラフは折れ線グラフ、「入場者数」は棒グラフ
- ・グラフのタイトルは14ポイント
- ・凡例や軸の数値は全て11ポイント
- ・テキストボックスを使用し、「気温の変化」は、縦軸に“°C”の単位、横軸に“時刻”と表示し、「入場者数」は、縦軸に“人”の単位、横軸に“イベント日”と表示

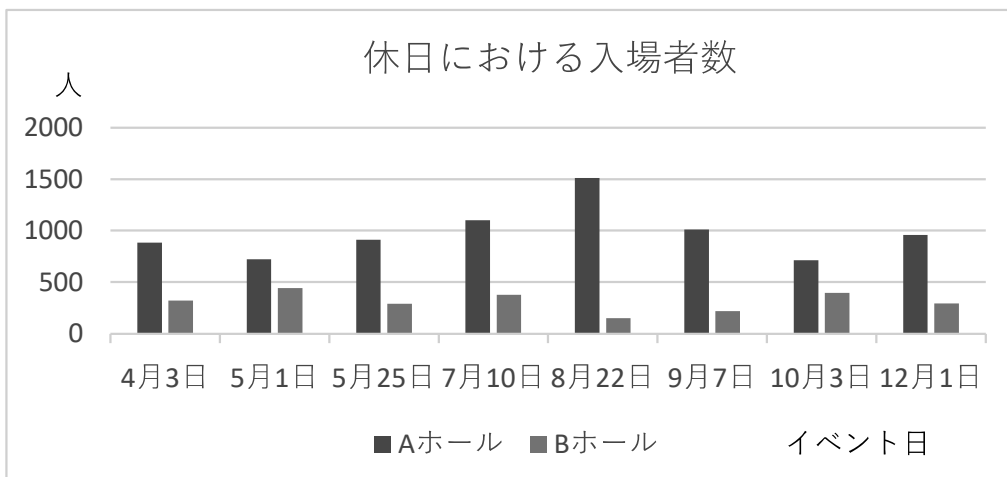
### ■気温の変化

	6:00	8:00	10:00	12:00	14:00	16:00	18:00	20:00
6月14日	18.2	19.9	22.5	26.1	27.9	28.3	27.7	22.9
3月1日	3.3	6.6	14.6	17.9	18.4	13.1	9.7	4.2



### ■入場者数

	4月3日	5月1日	5月25日	7月10日	8月22日	9月7日	10月3日	12月1日
Aホール	884	721	910	1100	1511	1010	711	959
Bホール	321	443	290	377	150	219	395	291



**【重要】**「折れ線グラフ」の使用は連続量のデータのみのみ。下段の入場者数は不連続量のため「折れ線グラフ」にしてはいけない。

ワークシート ( ) 年 ( ) 組 ( ) 番氏名 ( )

実験目的 ばね振り子において、おもりの質量と周期の関係を調べる。

実験装置 実験スタンド、ばね、おもり、ストップウォッチ

実験手順

- ①実験スタンドにばねの一端を固定し、他端におもりをとりつける。
- ②つりあいの位置から少し引いて静かに離し、ストップウォッチで周期を測定する。
- ③おもりの質量を変えて同様に実験を行う。

問 以下の表を作り、測定値を入力せよ。

おもりの質量 (g)	周期(s)				
	1回目	2回目	3回目	4回目	平均

問 上記の表を使って 横軸が「おもりの質量」、縦軸が「周期」の散布図を作れ。  
(平均を含めないこと)

(参考) 科学実験のグラフでよく使う機能

「グラフの縦と横を入れ替えたい」

グラフを選択した状態で [デザイン] タブをクリックし、データグループにある [データの選択] をクリックする。[系列] をクリックして [編集] をクリックし、[系列 X の値] と [系列 Y の値] を入れ替えて [OK] をクリックする。

「グラフの軸に単位を表示したい」

グラフをクリックし、「軸ラベル」にチェックを入れる。表示された「軸ラベル」をクリックして「質量(g)」などと編集する。

「軸ラベルを縦書きにしたい」

「軸ラベル」を右クリックして「軸ラベルの書式設定」をクリックする。「文字のオプション」をクリックし「テキストボックス」のアイコンをクリックして「文字列の方向」を「縦書き」に変更する。

「横軸を反転させたい」

グラフの横軸を右クリックして「軸の書式設定」を選び、「軸のオプション」の「軸を反転する」にチェックを入れる。

## エクセル練習⑦ 誤差やゆらぎの表示

科学実験等では「測定値」（実験器具などで測定した量）と「真の値」（理論的に導かれる量や定義した量）を分けて考える。グラフなどで「測定値」を示すとき、「測定値」が「真の値」に対してどれだけ信頼できるかを次の3つで表す。

1. "N値"・・・「測定値」の総数
2. "R2値"・・・近似直線（近似曲線）を引いたときの「決定係数」
3. "エラーバー"・・・「測定値」のゆらぎ（次のエクセル練習⑧で紹介する）

レポートや論文においては、上記が一つも示していないグラフは「見る価値がない」とさえいわれる。科学的に価値のあるデータの示し方を学んでいこう。

### ■必ずグラフにつけよう その1 "N値"・・・「測定値」の総数

グラフをつくるのに使った「測定値」の総数をグラフ上に示す。NはNumbersの頭文字を表す。

N=81

- 【手順】（テキストボックスはよく使うので覚えよう！）  
「挿入」→「テキストボックス」

### ■必ずグラフにつけよう その2 近似直線の"R2値"・・・近似直線の妥当性

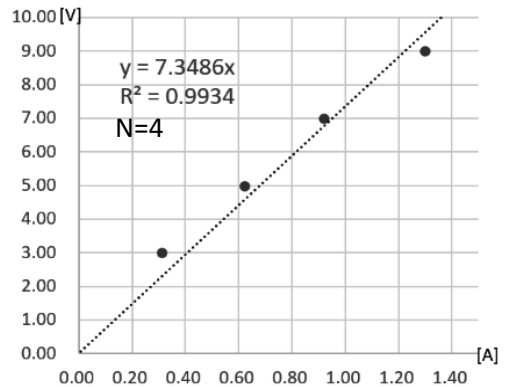
グラフのデータに近似直線や近似曲線を引いたとき、その直線や曲線の妥当性を表す数値を"R2値"（アールジジョウチウチ）または（ ）という。

右はグラフのデータに一次関数の近似直線を引いたときに得られた関数と"R2値"である。"R2値"は（ ）に近いほど確からしい関数といえる。

【手順】（今後よく使うので操作に慣れよう！）

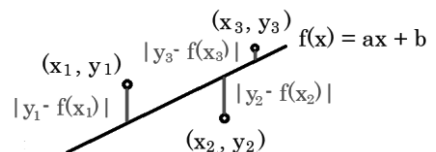
- ① グラフをクリックして「+」-「近似曲線」にチェックを入れ、矢印の先にある「その他のオプション」を選ぶ。
- ② 「線形近似」を選び、「グラフに数式を表示する」および「グラフにR-2乗値を表示する」にチェックを入れる。
- ③ 「前方補外」「後方補外」に適当な値を入れ、グラフの両端いっばいに近似直線を描く。

※「切片」にチェックを入れて値を入力すると、強制的に切片を通る直線で近似できる。



#### （参考）近似直線（曲線）を引く技術「最小自乗法」

図において、各測定点と仮に引いた近似曲線のy軸の距離を算出し足し合わせる。定数a,bを様々に変更して、距離の和が最小となるようなa,bを見つける。この方法を「最小自乗法」という。



#### （参考）「決定係数」の算出方法

「決定係数」は、右のように近似した関数と測定値の差を合計して得る。1に近いほど差が小さく、0に近いほど差が大きい。

$$\text{決定係数 } R^2 = 1 - \frac{\sum (y_i - f(x_i))^2}{\sum (y_i - \bar{y})^2}$$

## エクセル練習⑧ 誤差やゆらぎの表示

### ■必ずグラフにつけよう その3 "エラーバー"・・・「測定値」のゆらぎを表現する

多数の「測定値」をそのままグラフに表示すると、煩雑で分かりにくくなる場合がある。そこで、同一条件の「測定値」を平均するなどして「代表値」のみを表示すると見通しが立ちやすくなる。しかしこの場合、「測定値」がもっていたゆらぎや誤差（「真の値」からのずれ）の情報が消失してしまう。

そこで、平均値などの「代表値」を使って表示する場合、元の「測定値」がどの程度、ゆらぎや誤差を持っていたのかを「代表値」にエラーバーを付け加えて表現する。

【手順】（代表値を「平均値」とし、エラーバーを「測定値」の最大値と最小値で表現する場合）

①以下の表を作り、ワークシートの「測定値」を利用して値を入力する。

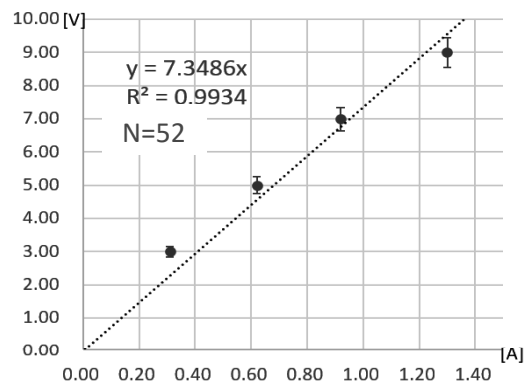
おもりの質量	周期の平均値 (代表値)	測定値の最大値と代表値の差	測定値の最小値と代表値の差

②グラフをクリックして「+」-「誤差範囲」にチェックを入れ、矢印の先にある「そのほかのオプション」を選ぶ。

③表示された「エラーバー」のうち、横線を選んでDeleteキーで消去し、縦線のみにする。

④誤差の範囲で「ユーザー設定」-「値の指定」を選ぶ。

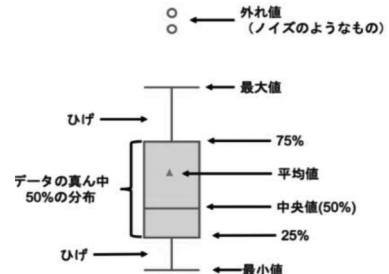
⑤表示される「正の誤差の値」に表の「最大値との差」、「負の誤差の値」に「最小値との差」をそれぞれドラッグして指定する。



エラーバーの例

#### (参考) 箱ひげ図

「測定値」の"ゆらぎ"を表現する方法として「箱ひげ図」を利用する場合があります、エラーバーよりもさらに細かな「測定値」のゆらぎや誤差を表現することができます。「箱ひげ図」は、実生活（地価調査など）や人文科学分野の研究（アンケート調査など）で使われている。





## エクセル練習⑨ 誤差やゆらぎの表示

■相対誤差・・・「測定値」が「真の値」からどれだけずれているかを表す

16歳のお兄さんは身長が165cmであり、10歳の弟の身長は135cmである。16歳男子の平均身長は170.0cmであり、10歳の平均身長は140cmである。兄も弟も平均から5cm低いが同等の差とはいえない。

「真の値」（と定義する量）から「測定値」がどれだけ離れているかを表す量としてよく「相対誤差」が用いられる。

$$\text{相対誤差(\%)} = \frac{\text{測定値} - \text{真の値}}{\text{真の値}} \times 100$$

「相対誤差」が大きいほど、「測定値」は「真の値」から外れていることになる。

問1 以下の表を作り、平均身長を「真の値」として「相対誤差」を計算せよ。

	身長(cm)	平均身長(cm)	相対誤差(%)
兄			
弟			

問2 以下の表を作り、1～6の乱数の平均値を「真の値」として「相対誤差」を計算し、表を埋めよ。

試行回数	1～6の乱数	相対誤差
平均		
1		
2		
3		
4		
5		

問3 問1の試行回数を100回にするとどうなるか確認せよ。

(参考) 「乱数」・・・RANDBETWEEN(1,100) で1～100の整数値の乱数を表示

(参考) 「絶対参照」・・・参照するセルを変えず固定する方法【重要】

「オートフィル」（一つのセルに計算式を入れて、「+」を使って他セルに同様の計算をコピーする方法）を使うと、自動でセル番号が変更される。このとき、計算式の中に「\$A\$1」と入れると、「オートフィル」を用いても常に「A1」セルは固定される。これを「絶対参照」という。

## エクセル練習⑩ SUM, AVERAGE, IF関数

中間考査の結果

氏名	国語	数学	英語	理科	社会	総合点	平均	判定
Aくん	84	78	79	56	88			
Bくん	65	97	75	67	44			
Cさん	78	56	53	86	57			
Dさん	67	79	83	46	68			
Eくん	55	80	57	88	78			
クラス平均								

問1 「総合点」をSUM関数、「平均」および「クラス平均」をAVERAGE関数を用いて計算せよ。

問2 IF関数を用いて、各生徒の平均がクラスの平均以上のときに「判定」欄に"やったね"、平均未満のときに"がんばろう"を表示せよ。

IF関数の使い方・・・次の関数を入力セルは「A1の値がA2の値以上のとき○、未満のときに×」が表示される

`=IF(A1>=A2,"○","×")`

問3 個人の平均点が高い生徒順に並べよ

並べ替えの手順・・・並べ替えたいデータの範囲を選択した後に

「データ」->「並び替え」

メモ

## エクセル練習⑪ リスト機能とCOUNTIF関数

同窓会の出欠

	出欠
岩国 一郎	
柳井 二郎	
下松 三郎	
徳山 花子	
山口 太郎	
周南 次郎	
参加人数	0

○  
×

←この○と×のセルを参照してリストをつくる

問1 「出欠」欄をリスト機能を使ってプルダウンで○または×を選択できるようにして、岩国一郎、徳山花子、周南二郎を○、それ以外を×にせよ。

リスト機能のしかた・・・プルダウンさせたいセルを選択した後、  
「データ」->「データの入力規則」->「リスト」

問2 「参加人数」にCOUNTIF関数を使って○の人数を計算せよ。

COUNTIF関数の使い方・・・次の関数を入力したセルには、A1からA5の範囲でB1の値と一致するセルの合計が表示される

=COUNTIF(A1:A5,B1)

メモ

## エクセル練習⑫ VLOOKUP関数

住所録

氏名	都道府県	住所	年齢
岩国 一郎	山口	岩国市通津町	21
柳井 二郎	岡山	岡山市栄町	24
下松 三郎	広島	東広島市通町	29
徳山 花子	山口	山口市湯田町	25
山口 太郎	東京	台東区水町	28
周南 次郎	大阪	堺市中央町	26

氏名	都道府県	住所	年齢

問1 「氏名」をリスト機能を使って氏名を選択できるようにせよ。

問2 VLOOKUP関数を使って問1で選択した氏名に対する都道府県、住所、年齢を表示せよ。

### VLOOKUP関数の使い方

次の関数を入力したセルには「C1からF5の範囲でA1の値と一致した行の左から2番目の列の値」が表示される

**=VLOOKUP(A1,C1:F5,2,FALSE)**

	A	B	C	D	E	F
1	さしすせそ		あいうえお	1	6	11
2			かきくけこ	2	7	12
3			さしすせそ	3	8	13
4			たちつてと	4	9	14
5			なにぬねの	5	10	15
6						

例えば上記の関数をA2に入力すると、「3」が表示される

メモ

## エクセル練習⑬ まとめと演習

中間考査の結果

氏名	国語	数学	英語	理科	社会
岩国 一郎	84	78	79	56	88
柳井 二郎	65	97	75	67	44
下松 三郎	78	56	53	86	57
徳山 花子	67	79	83	46	68
山口 太郎	55	80	57	88	78
周南 次郎	53	68	99	95	71
クラス総合点					
クラス平均					

氏名	国語	数学	英語	理科	社会

問1 「クラス総合点」および「クラス平均」をSUMやAVERAGE関数を用いて計算せよ。

問2 「氏名」をプルダウンで選べるようにし、VLOOKUP関数を使って選択した氏名に対する各教科の点数を表示せよ。

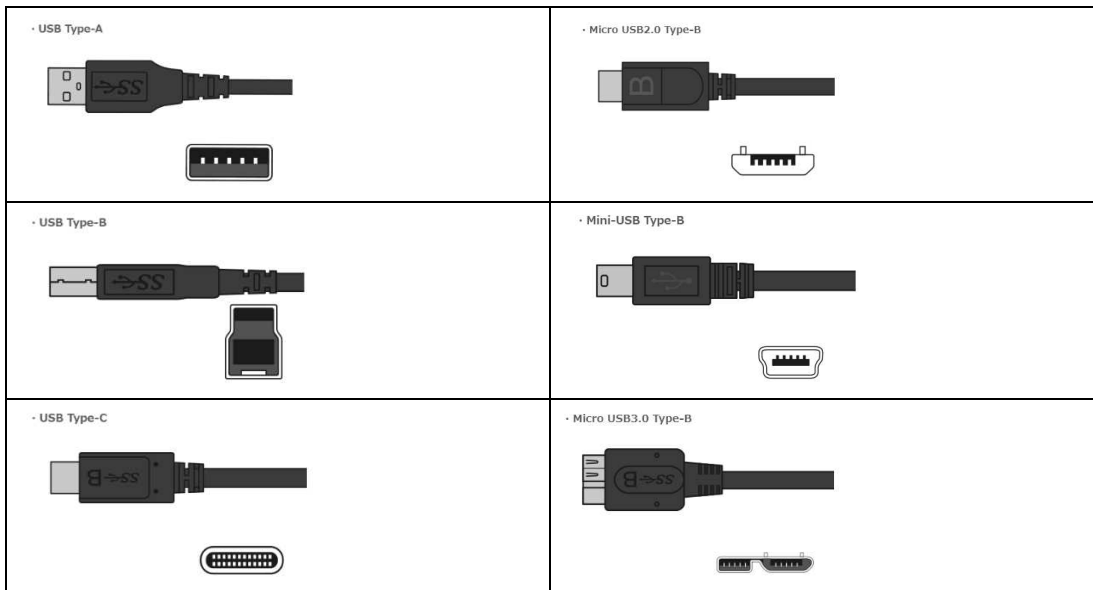
問3 問2で選択した氏名の各教科の点数を棒グラフで表示せよ。（縦軸；点数、横軸；教科名）

メモ

## USB メモリと転送速度

### 1. USB とは

- ・ USB ( )・・・端子の名称
- ・ 主に機器間の ( ) や機器の ( ) に使用される。
- ・ 形が異なる USB 端子が6種類もある。形状が異なるとつなぐことができない



### 2. USB の転送速度

機器間のデータを転送する速度が異なる

1996年 USB1.0 12Mbps

2000年 USB2.0 480Mbps

2008年 USB3.0 5Gbps

2013年 USB3.1 10Gbps

### 3. フォーマット

USB メモリを使うには最初にフォーマット ( ) する必要がある。これを行うと、元のファイルは ( ) 消去される。

データの伝送速度を表す量

bps( )

1Byte は 8bit だから

8bps = 1Byte/s である。

つまり、16GB=16×10<sup>9</sup>Byte のファイルを

USB1.0 で転送すると理論上、

$16 \times 10^9 \times 8 / (12 \times 10^6) = 11 \times 10^3 \text{ s} = 3 \text{ 時間}$

かかるが、

USB3.1 で転送すると、

となる。

# ファイル名と拡張子

## 1. 拡張子とは

拡張子からそのファイルが何の種類かが分かる。種類が分かれば、閲覧したり再生するソフトウェアが分かるため便利である。ただし、ファイル名を変更してもファイルの中身は変わらないが、拡張子を変更すると、中身は変わらないのに種類を強制的に変えるため、閲覧したり編集できなくなる可能性がある。

## 2. 拡張子の種類

### ①文書を表す拡張子

docx                      txt

### ②画像を表す拡張子

jpg                        png

### ③音楽を表す拡張子

mp3                        wav

### ④映像を表す拡張子

mp4                        avi

## 3. PDF

PDF または pdf は（株）Adobe が開発したファイルの規格で、フォントやレイアウトを崩さずに相手にファイルを送れるとても便利な特徴を持つ。

ワードやパワーポイントのファイルは、受け取る相手がワードやパワーポイントのソフトウェアを持っていないと閲覧できない。また、使用したフォントを相手を持っていないと、「文字化け」という現象が起きて情報を読み取ることができなくなる。

その点 pdf は、閲覧するための「acrobat reader」（無料）などが PC や Mac、スマホなどに搭載され、事実上の世界標準となっており、文字等の情報をベクトル情報として保存するため文字化けの心配がない。

そこで、ワードやパワーポイント等でレポートやポスターの原稿を作成した後、pdf の拡張子を指定して保存し、生成された pdf の拡張子のファイルを相手に送れば、こうした問題を回避することができる。

このように、「pdf で送る」は、「データを紙の上に書いた情報のように送る」ことができる、相手を思いやる方法といえる。



## AI 関連プログラム

`load_images.py` images フォルダから学習画像を読み込んで学習できる形に変更 (前処理という) するプログラム「train.py」と自動で連動する。

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Jun 30 16:41:01 2018
4 @author: sprin
5 """
6
7 import numpy as np
8 import glob
9 from keras.preprocessing.image import load_img, img_to_array, ImageDataGenerator
10 import re
11 import cv2
12 from PIL import Image
13 from PIL import ImageEnhance
14 import random
15
16 def load_images_from_labelFolder(path, img_width, img_height, train_test_ratio=(9,1)):
17     pathsAndLabels = []
18     label_i = 0
19     data_list = glob.glob(path + '%%*')
20     datatxt = open('whoiswho.txt', 'w')
21     print('data_list', data_list)
22     for dataFolderName in data_list:
23         pathsAndLabels.append([dataFolderName, label_i])
24         pattern = r"%%(.*)"
25         matchOB = re.finditer(pattern, dataFolderName)
26         directoryname = ""
27         if matchOB:
28             for a in matchOB:
29                 directoryname += a.groups()[0]
30         datatxt.write(directoryname + "," + str(label_i) + "\n")
31         label_i = label_i + 1
32     datatxt.close()
33     allData = []
34     for pathAndLabel in pathsAndLabels:
35         path = pathAndLabel[0]
36         label = pathAndLabel[1]
37         imagelist = glob.glob(path + "%%*")
38         for imgName in imagelist:
39             allData.append((imgName, label))
40     allData = np.random.permutation(allData)
41
42     train_x = []
43     train_y = []
44
45
46     for (imgpath, label) in allData:
47         ### ImageEnhanceで画像処理はPIL形式
48         ### OpenCVで画像処理はSRC形式
49
50         #Load image to PIL (RGB) (not openCV SRC(BGR))
51         img_pil = load_img(imgpath, target_size=(img_width, img_height))
52         #PIL to SRC
53         img_src = np.asarray(img_pil)
54
55         ###Normal #####
56         #PIL to array for train
57         imgarry = img_to_array(img_pil) # 画像ファイルを学習のためにarrayに変換する。
58         train_x.append(imgarry)
59         train_y.append(label)
60
61     threshold = (train_test_ratio[0]*len(train_x))/(train_test_ratio[0]+train_test_ratio[1])
62     test_x = np.array(train_x[threshold:])
63     test_y = np.array(train_y[threshold:])
64     train_x = np.array(train_x[:threshold])
65     train_y = np.array(train_y[:threshold])
66
67     return (train_x, train_y), (test_x, test_y)
```



`train.py` ニューラルネットワークによって images フォルダにある画像を機械学習して「学習済みモデル」(mymodel.h5 と whoiswho.txt)を生成するプログラム。「load\_images.py」と自動で連動する。

```
1 from __future__ import print_function
2 from tensorflow.python import keras
3 from tensorflow.python.keras.datasets import mnist
4 from tensorflow.python.keras.models import Sequential
5 from tensorflow.python.keras.layers import Dense, Dropout, Flatten
6 from tensorflow.python.keras.layers import Conv2D, MaxPooling2D
7 from tensorflow.python.keras import backend as K
8
9 import argparse
10 from load_images import load_images_from_labelFolder
11 import matplotlib.pyplot as plt
12
13 batch_size = 20
14 num_classes = 2
15 epoch = 2 #default 30
16 img_rows, img_cols = 128, 128
17
18 parser = argparse.ArgumentParser()
19 parser.add_argument('--path', '-p', default='./images')
20 args = parser.parse_args()
21
22 (x_train, y_train), (x_test, y_test) = load_images_from_labelFolder(args.path, img_cols, img_rows, train_test_ratio=(4,1))
23
24 if K.image_data_format() == 'channels_first':
25     x_train = x_train.reshape(x_train.shape[0], 1, img_rows, img_cols)
26     x_test = x_test.reshape(x_test.shape[0], 1, img_rows, img_cols)
27     input_shape = (1, img_rows, img_cols)
28 else:
29     x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 3)
30     x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 3)
31     input_shape = (img_rows, img_cols, 3)
32
33 x_train = x_train.astype('float32')
34 x_test = x_test.astype('float32')
35 x_train /= 255
36 x_test /= 255
37
38
39 # convert class vectors to binary class matrices
40 y_train = keras.utils.to_categorical(y_train, num_classes)
41 y_test = keras.utils.to_categorical(y_test, num_classes)
42
43 model = Sequential()
44
45 #test 9 sessions 20190122 max acc , low cross entropy #epoch 30
46 model.add(Conv2D(32, (3, 3),
47                 activation='relu',
48                 input_shape=input_shape))
49 model.add(Conv2D(32, (3, 3),
50                 activation='relu',
51                 input_shape=input_shape))
52 model.add(MaxPooling2D(pool_size=(2, 2)))
53 model.add(Dropout(0.4))
54
55 model.add(Conv2D(64, (3, 3),
56                 activation='relu',
57                 input_shape=input_shape))
58 model.add(Conv2D(64, (3, 3),
59                 activation='relu',
60                 input_shape=input_shape))
61 model.add(MaxPooling2D(pool_size=(2, 2)))
62 model.add(Dropout(0.6))
63
64 model.add(Flatten())
65 model.add(Dense(units=64, activation='relu'))
66 model.add(Dropout(0.6))
67 model.add(Dense(units=num_classes, activation='softmax'))
68
69 # Summary
70 model.summary()
71
72 model.compile(loss=keras.losses.categorical_crossentropy,
73              optimizer=keras.optimizers.Adam(),
74              metrics=['accuracy'])
75
76 ##change
77 model.fit(x_train, y_train,
78         batch_size=batch_size,
79         epochs=epoch,
80         verbose=1,
81         validation_data=(x_test, y_test))
82
83 score = model.evaluate(x_test, y_test, verbose=0)
84 print('Test loss:', score[0])
85 print('Test accuracy:', score[1])
86
87 model.save('mymodel.h5')
```

prediction.py 「学習済みモデル」を元に web カメラの画像を判断するプログラム

```
1 from keras.preprocessing.image import load_img, img_to_array
2 from keras.models import Sequential, load_model
3 import numpy as np
4 import cv2
5 from PIL import ImageFont, ImageDraw, Image
6 import time
7 def main():
8     #####
9     # number of folders in images
10    num_classes = 5
11    #####
12    img_rows, img_cols = 128, 128
13    ### Read whoiswho.txt ###
14    ident = [""] * num_classes
15    for line in open("whoiswho.txt", "r"):
16        dirname = line.split(",")[0]
17        label = line.split(",")[1]
18        ident[int(label)] = dirname
19    ### Read mymodel.h5 ###
20    model = load_model("mymodel.h5")
21    ### camera_setting
22    capture = cv2.VideoCapture(0)
23    capture.set(3, 640) # Width
24    capture.set(4, 480) # Height
25    capture.set(5, 15) # FPS 15or30
26    ### Read font ###
27    font = ImageFont.truetype('C:\WINDOWS\FONTS\MSGOthic.TTC', 24)
28    #####
29    while(True):
30        ### Capture Web Camera #####
31        ret, img = capture.read()
32        if(ret == False):
33            print("Capture Failed. ")
34            break
35        key = cv2.waitKey(1)
36        if(key == 27):
37            print("exit.")
38            break
39        #####
40        Datas = []
41        imgarray = []
42        #resize for h5 model
43        dst = cv2.resize(img, (img_rows, img_cols))
44        time.sleep(0.1)
45        #build up data for h5 model
46        CV_im_RGB = dst[:, :, :-1].copy()
47        pillow_img = Image.fromarray(CV_im_RGB)
48        Datas.append(pillow_img)
49        for data in Datas:
50            imgarray.append(img_to_array(data))
51        imgarray = np.array(imgarray) / 255.0
52        imgarray.astype('float32')
53        #predict by data
54        preds = model.predict(imgarray, batch_size=imgarray.shape[0])
55        time.sleep(0.1)
56        try:
57            for pred in preds:
58                #show pred KAKURITSU
59                for i in range(num_classes):
60                    print("i=", i, ", ", ident[i], "=", preds[0][i])
61                #show pred name
62                max_pred_i = np.argmax(preds)
63                print("This is {}".format(ident[max_pred_i]))
64                #draw on picture Japanese fonts
65                img_pil = Image.fromarray(img)
66                draw = ImageDraw.Draw(img_pil)
67                draw.text((20, 20), ident[max_pred_i], font = font, fill = (100,
68                100, 0))
69                draw.text((20, 50), str(int(100*preds[0][max_pred_i]))+"%", font
70                = font, fill = (100, 100, 0))
71                img = np.array(img_pil)
72            except:
73                print("error2")
74            #show cam image after put on text
75            cv2.imshow("cam", img)
76 if __name__ == '__main__':
77     main()
78     capture.release()
79     cv2.destroyAllWindows()
```

## スクレイピング

「MyProgram」の「Scraping」というフォルダに「scraping.py」というプログラムがあります。これを使うと入力したキーワードを元に google の画像検索を利用して自動で画像ファイルをダウンロードします。この作業を「スクレイピング」といいます。

注意；scraping.py は同じフォルダにある chromedriver.exe を用います。動作しない場合は、ブラウザ chrome のバージョンを確認し、対応した chromedriver.exe をダウンロードして上書きしてください。

## AI のプログラミング環境を構築する方法

徳山高校の S S H 科目「AI 研究入門」では、プログラム環境を USB メモリ内につくっています。これにより、PC に様々なソフトを直接インストールすることなく、USB メモリを PC に差し込むだけでプログラミングできるようになります。

ただし、この USB メモリでプログラミングするには、上記の条件を満たす PC と USB メモリが必要です。

### 【必要な PC 性能】

- ・ windows10 64bit (32bit 不可)
- ・ Core i3 以上の CPU
- ・ 4GB 以上のメモリ

### ■手順

- ① 「WinPython」の 64bit 版をダウンロードする。

<https://sourceforge.net/projects/winpython/> の「Files」を選んで「WinPython64〇〇.exe」を探してダウンロードする。

- ② ①を USB メモリに解凍後、「WinPython Command Prompt.exe」を起動する。

- ③ `pip install opencv-contrib-python` を入力してエンターキーを押す（ライブラリ opencv-contrib-python のインストール）

- ④ ③と同様に、ライブラリ tensorflow および keras をインストールする

- ⑤ 「VSCode.zip」をダウンロードする。

<https://azure.microsoft.com/ja-jp/products/visual-studio-code/> の「今すぐダウンロード」

- ⑥ ダウンロードしたファイルを解凍後、「VSCode」フォルダの直下に「data」という名称のフォルダを新たに作成してから「Code.exe」を実行する。

- ⑦ VSCode の画面左端アイコンの「Extensions」を選び、Search 欄を利用して python および Japanese Language Pack をインストールする。インストール後にいったん VSCode を閉じる。

- ⑧ 「VSCode」を USB メモリにコピーする。

- ⑨ 徳山高校の HP から「MyProgram.zip」をダウンロードして USB メモリに解凍する。

# ポスターセッションとポスター

## 1. ポスターセッションとは

ポスターセッションとは、研究成果や製品等を発表する方法のひとつで、現在様々な発表会で主流となっています。つまり、ポスターセッションやその評価は、大学に入っても、社会人になってもずっとつきあっていく可能性が高く、今のうちからしっかりとスキルを磨いておく必要があります。

ポスターセッションは、通常たくさんのポスターが同時に掲示され、発表者はその前に立って発表します。口頭発表と比べて、観客は興味関心に応じて自由に観覧し、発表者と観客の物理的な距離が近いという大きな違いがあります。そこで、次の点に注意してポスターセッションしましょう。

### 【ポスターセッションにおける発表の注意点】

- ① 発表時間は5分以下とする。3分程度が望ましい。
- ② 事前に質問内容の想定と回答の準備を行う。
- ③ できるだけ実験道具、開発装置、動画、写真等を準備する。配布用資料を用意してもよい。
- ④ ポスターのどこを説明しているのかを、ときおり手を使って示す。
- ⑤ 質問に対しては「結論」→「具体」あるいは「論拠」の順で答える。
- ⑥ 終始明るい表情に留意する。相手の目を見て話す、語尾をはっきりいう、「あー」「えー」などの口癖があればできるだけ修正する。
- ⑦ ジェスチャーを織り交ぜたり、声の大きさに強弱をつけたりして、より分かりやすく工夫する。

これらの注意点は、すぐに身につかないものもあります。しかし、繰り返し練習したり日頃から意識したりすることで改善できる内容でもあります。身につければ一生モノですので、ぜひ発表の機会を活用して努力を重ねてください。



## 2. ポスターづくり

ポスターセッションに使用するポスターは、通常A0サイズやA1サイズなどに大判印刷します。従って、ポスターは発表内容をまとめた「資料」ですが、同時にポスターセッションの「顔」となります。数多くのポスターが貼られる中で、内容をきちんと評価してもらうためには、特に次の点に注意してポスターをつくりましょう。

## 【ポスターづくりの注意点】

- ① タイトルが重要。フォントは大きく、内容を端的に表すと同時に目を引くものにする。  
「瀬戸内海における石油分解菌について」  
→「瀬戸内海で発見！石油分解菌」
- ② 「タイトル」「氏名」に続いて、「背景」「目的」「方法」「結果」などを項目ごとに明記する。
- ③ 白地に黒字が基本だが、グラフや本文の下地に薄い色をつけて内容の交通整理（カテゴライズ）したり、強調したい箇所に濃い色やアイコンを使用するなど、表現に工夫を行う。
- ④ カラーチャートを使って色の組み合わせを工夫するとともに、フォントの大きさや種類を揃える。  
例) タイトル 24 ポイント、項目 14 ポイント、本文 10 ポイント（A4サイズで作成の場合）
- ⑤ 表よりもグラフが分かりやすい。グラフを使用する場合は、必ず単位を明記する。グラフの種類（折れ線、度数分布、棒グラフ）や誤差やサンプル数の表記は理数の基本。
- ⑥ デザインは「いいポスター」を真似る。

これらの注意点は、特に発表準備に時間がないときに見過ごされやすくなります。しかし、どんなに内容が良くても（あるいはその逆であっても）、ポスターの見目が評価に大きく影響するという現実があります。ポスター作成は複数人で取り組むことが多いですので、ぜひ役割分担してよいものに仕上げてください。もちろんポスターを使った練習が必要なのは言うまでもありません。

なお、学会やコンテストのポスターセッションで評価が高い発表は、「どんな発表をするかをイメージしてつくられたポスター」です。発表の流れとポスター記述の流れがぴたり一致しているのです。これは口頭発表やポスターセッションの違いに関係なく、発表を何度か経験し、練習を積まないと身につけません。ぜひジョブスなどの一流プレゼンターを参考にしてください。

## 【一流プレゼンテーション】

### 1. ジョブスによる iPhone の発表

<https://www.youtube.com/watch?v=90gRARwFxCh&t=26s>



### 2. 滝川クリステルによる東京オリンピック誘致プレゼン

<https://www.youtube.com/watch?v=6hgggygKWwhg>



### 3. 第 15 回日銀グランプリ⑤学習院大学プレゼンテーション

<https://www.youtube.com/watch?v=yVqf21fPzxl>



本冊子は山口県立徳山高等学校のスーパーサイエンスハイスクール（SSH）事業における学校設定科目「AI 研究入門」のテキストとして制作しました。

山口県立徳山高等学校は、文部科学省より SSH に研究指定されています。

「AI 研究入門」  
著作 末谷健志  
教諭・山口県立徳山高等学校



<http://www.tokuyama-h.ysn21.jp/bcms/index.php/tokko>

上記の学校 HP から本冊子の PDF やプログラミングに必要な環境ファイルをダウンロードできます。

初 版 2019 年 12 月  
第 2 版 2020 年 10 月  
第 3 版 2022 年 3 月