

# 「Python プログラミング基礎その 1」

九州大学 大学院システム情報科学研究院  
情報知能工学部門  
データサイエンス実践特別講座  
備瀬 竜馬, 末廣 大貴, Diego Thomas

復習

# Google Colaboratory

← → ↺ [https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja#scrollTo=xitplqMNk\\_Hc](https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja#scrollTo=xitplqMNk_Hc) ☆

Colaboratory へようこそ

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

+ コード + テキスト | ドライブにコピー

接続 | 編集

目次 | コード スニペット | ファイル X

Colaboratory の概要

はじめに

その他のリソース

機械学習の例: Seedbank

セクション

Colaboratory へようこそ

Colaboratory は、完全にクラウドで実行される Jupyter ノートブック環境です。設定不要で、無料でご利用になれます。

Colaboratory を使用すると、コードの記述と実行、解析の保存や共有、強力なコンピューティングリソースへのアクセスなどをブラウザからすべて無料で行えます。

Colaboratory の概要

この 3 分間の動画では、Colaboratory の主な機能の概要をご紹介します。

Get started with Google Colaboratory (Coding T...

後で見る 共有

Intro to Google Colab

Coding TensorFlow

自分のGoogle  
アカウントで  
ログイン

ファイルを  
セーブできる  
ようになる

# Jupyter notebookを使う

Jupyter notebook: Pythonの開発環境の一つ

ファイルを  
クリック

Python3の新しい  
ノートブック  
をクリック

※) 数字はPython  
のバージョン

The screenshot shows the Google Colaboratory web interface. The browser address bar displays `https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja`. The page title is "Colaboratory へようこそ". The "File" menu is open, showing options like "Python 3 の新しいノートブック" (highlighted with a blue box and an arrow), "Python 2 の新しいノートブック", "ノートブックを開く...", "ノートブックをアップロード...", "名前を変更...", "ゴミ箱に移動", "ドライブにコピーを保存...", "コピーを GitHub Gist として保存...", "GitHub にコピーを保存...", "保存" (with keyboard shortcuts Ctrl+S and Ctrl+M S), "変更内容を保存して固定", "変更履歴", ".ipynb をダウンロード", ".py をダウンロード", "ドライブプレビューを更新", and "印刷" (with keyboard shortcut Ctrl+P). The main content area features a "Colaboratory へようこそ" message, a video player titled "Get started with Google Colaboratory (Coding T...)", and a "Coding TensorFlow" logo. The video player shows a man smiling and the text "Intro to Google Colab".

Colaboratory へようこそ

Ctrl+S  
Ctrl+M S  
Ctrl+P

Get started with Google Colaboratory (Coding T...)

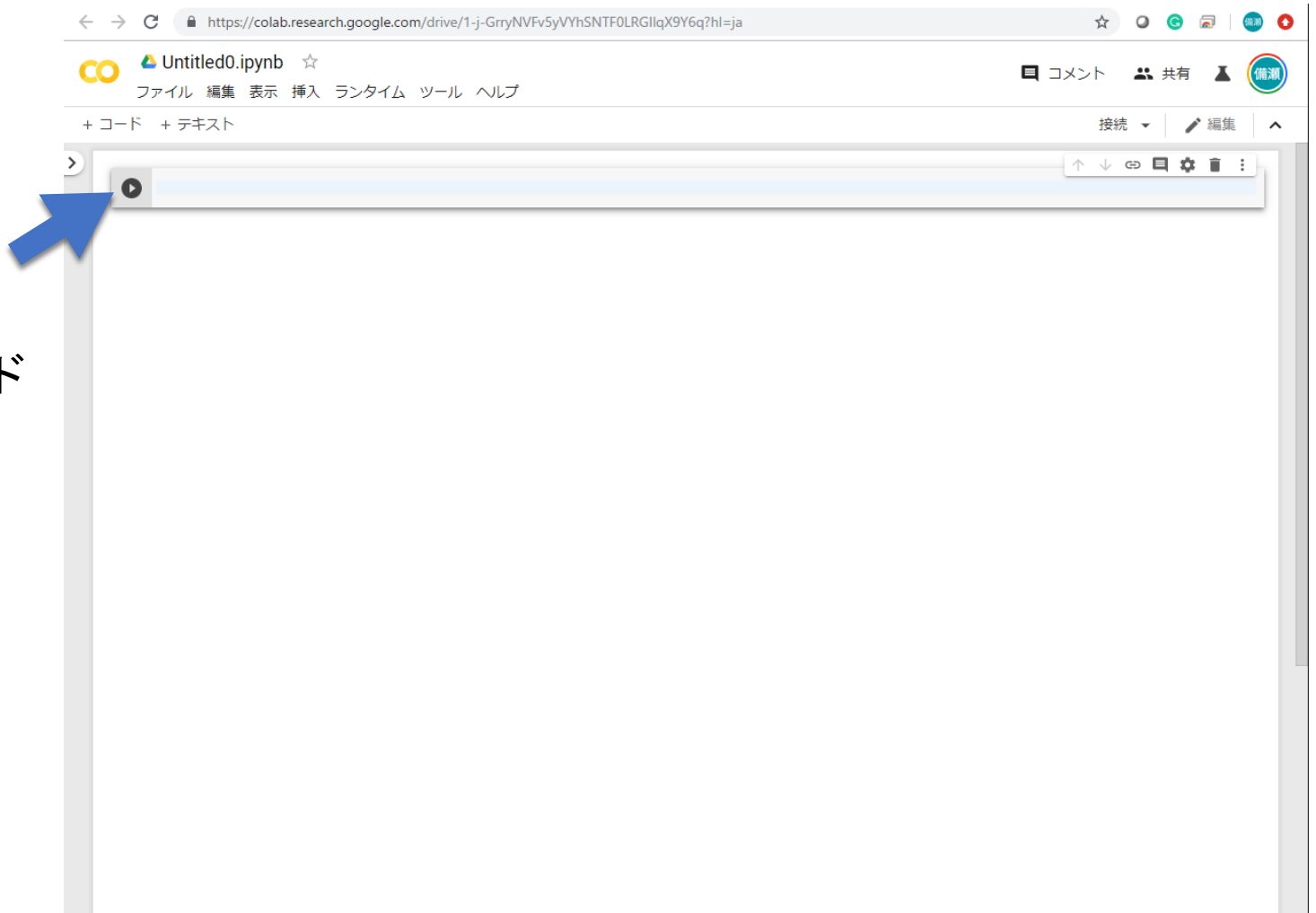
Intro to Google Colab

Coding TensorFlow

はじめに

# Cell

Cell  
プログラムの  
入力フィールド  
Cellの単位で  
実行される



list

# list

- 数字のリスト：[]の中に数字をコンマ区切りで記載
- 例：

```
In [19]: a = [0,1,2]
```

```
In [20]: print( a )  
[0, 1, 2]
```

```
In [21]: b = [2, 0.5, 4, 2, 1]
```

```
In [22]: print( b )  
[2, 0.5, 4, 2, 1]
```

# list

- 数字以外でも、好きなものをリストにできます

- 例 1 : 文字列のリスト

```
In [336]: c = ["Hello", "World"]
```

```
In [337]: print(c)  
['Hello', 'World']
```

- 例 2 : 文字列と数字のリスト

```
In [370]: d=["A", 52, 1.7]
```

名前、体重、身長をリストにする

```
In [371]: print(d)
```

```
In [371]: ['A', 52, 1.7]
```

- 例 3 : リストのリスト

```
In [368]: e=[["A", 52, 1.7], ["B", 75, 1.6]]
```

リストのリスト

```
In [369]: print(e)
```

```
In [369]: [['A', 52, 1.7], ['B', 75, 1.6]]
```



# list

- リストの要素の呼び出し

例：

```
In [46]: b = [2, 0.5, 4, 2, 1]
```

```
In [50]: print(b[0]) 0番目の要素を出力  
2
```

```
In [51]: print(b[4]) 4番目の要素を出力  
1
```

※) pythonでは、0から数える

# list

- リストのリストにも次のようにすれば呼び出せる

```
In [374]: e=[["A", 52, 1.7], ["B", 75, 1.6]]
```

```
In [375]: print(e[1][1])
```

1番目の要素（リスト）の

```
In [375]: 75
```

1番目の要素を出力

# list

## ● リストの初期化と要素追加

例：

```
In [114]: list=[]
```

初期化

```
In [115]: list.append(1)
```

Listの要素に"1"を追加

```
In [116]: list.append(2)
```

Listの要素に"2"を追加

```
In [117]: print(list)  
[1, 2]
```

※) list.append(x)で、listの最後にxが追加される

# list

- リストのリストにも次のようにすれば追加できる  
例：

```
In [379]: e=[["A", 52, 1.7], ["B", 75, 1.6]]
```

```
In [380]: e[0].append(27)    e の 0 番目のリストに27を追加
```

```
In [381]: print(e)
```

```
In [381]: [['A', 52, 1.7, 27], ['B', 75, 1.6]]
```

# 練習 1 : list

- 以下の3人のリストを作ろう

- A君 : 1.7m, 52kg
- B君 : 1.6m, 75kg
- C君 : 1.8m, 73kg

ヒント) [[名前、体重、身長], [...], [...]]

- BMIを計算し、BMIsという名前のリストに下記の順で入れてみよう

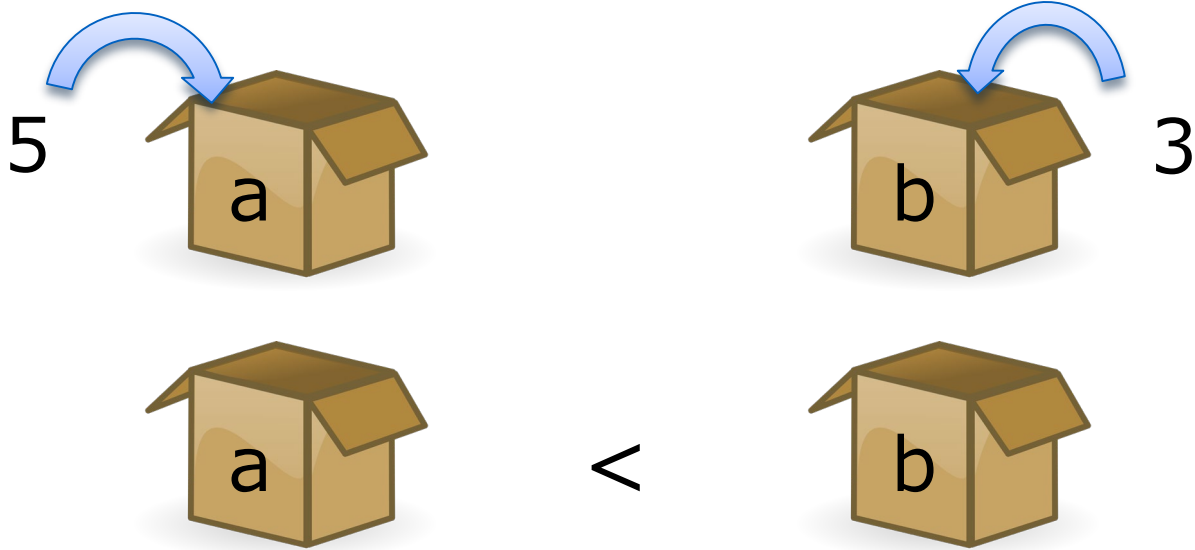
ヒント) listの初期化が必要 : BMIs=[]

listに要素の追加 : BMIs.append(xx)

- リストBMIsの1番目 (B君) の値を表示してみよう

if

# 比較 (1)



a が b より小さい？



```
In [306]: a < b  
Out[306]: False
```

True  
(正しい)

or

False  
(正しくない)

## 比較（２）

演算子	例	意味	出力
==	a==b	a と b が等しい？	True (正しい)  または  False (正しくない)
!=	a!=b	a と b が異なる？	
>	a>b	a が b より大きい？	
>=	a>=b	a が b 以上？	
<	a<b	a が b より小さい？	
<=	a<=b	a が b 以下？	

例

In [27]: 3==2  
Out[27]: False

In [28]: 4.0<4.1  
Out[28]: True

In [29]: 3==2  
Out[29]: False

In [30]: 4.0<4.0  
Out[30]: False

In [31]: 4.0<=4.0  
Out[31]: True

In [32]: 1!=2  
Out[32]: True



「a=b」は、aにbの中身を入れると  
 いう意味だったので、  
 混同しないように比較は、「==」

演算子	例		
==	a==b	a と b が等しい？	True (正しい)  または  False (正しくない)
!=	a!=b	a と b が異なる？	
>	a>b	a が b より大きい？	
>=	a>=b	a が b 以上？	
<	a<b	a が b より小さい？	
<=	a<=b	a が b 以下？	

例

In [27]: 3==2  
 Out[27]: False

In [28]: 4.0<4.1  
 Out[28]: True

In [29]: 3==2  
 Out[29]: False

In [30]: 4.0<4.0  
 Out[30]: False

In [31]: 4.0<=4.0  
 Out[31]: True

In [32]: 1!=2  
 Out[32]: True

# 「かつ」と「または」(1)

- 例：あなたの結婚の条件は？

25歳以上**かつ**身長180cm以上！

age= 30

height=1.68

age>=25 and height>=1.80

```
In [129]: age=30
```

```
In [130]: height=1.68
```

```
In [131]: age>=20 and height>=1.80
```

```
Out[131]: False
```

# 「かつ」と「または」(2)

演算子	例	意味	出力
and	a and b	a <b>かつ</b> b が 両方 正しい？	True (正しい) または False (正しくない)
or	a or b	a <b>または</b> b の どちらかが 正しい？	

# if, elif, else

- 条件文で分岐をつくる
- 例：20歳以上は“Adult”，13歳以上20歳未満なら “Young”，13未満は“Child”

```
age=30
```

```
if age>=20:
```

```
    print( "Adult" ) # もし20歳以上なら...
```

```
elif age>=13:
```

```
    print( "Young" ) # 20歳以上でない。 でももし13歳以上なら...
```

```
else:
```

```
    print( "Child" ) # 20歳以上でも13歳以上でもないなら...
```

## ところで・・・ 「コロン」と「インデント」

if age>=20:

print( "OK" )

else:

print( "NG" )

- コロン「:」で「次の行に続きますよ」
- インデント「空白 4 つ」で「続いていますよ」

## 練習 2 : if

- BMIを入力として与えて、以下のように表示するプログラムを作ろう！
  - 18.5未満の場合、「やせ型」と表示
  - 18.5以上25未満の場合、「標準」と表示
  - 25以上の場合、「肥満気味」と表示
- 以下の3人のBMIを計算し、上のプログラムで動かそう
  - A君 : 1.7m, 52kg
  - B君 : 1.6m, 75kg
  - C君 : 1.8m, 73kg

for

# for ( 1 )

- 同じ計算を繰り返す

- 例 :

```
for i in [0,1,2]:  
    print( i )
```

リストの中身がなくなるまで繰り返す



1回目は、i に0を代入して、  
print i

2回目は、i に1を代入して、  
print i

3回目は、i に2を代入して、  
print i

```
In [343]: for i in [0,1,2]:  
         ....:     print(i)  
         ....:  
         ....:
```

0

1

2



## for ( 2 )

- リストの中身を増やすと？

```
for i in [0,1,2,3,4]:  
    print( i )
```

```
In [344]: for i in [0,1,2,3,4]:  
         ....:     print(i)  
         ....:  
         ....:
```

0  
1  
2  
3  
4

100回繰り返したい場合は、  
リストの中に、100個書く？

# for ( 3 )

```
for i in range(5):  
    print( i )
```

```
for i in [0,1,2,3,4]:  
    print( i )
```

range()

整数のリストを作る関数

0～5未満の整数のリスト

In [331]: range(5)

Out[331]: [0, 1, 2, 3, 4]



同じ

## for (4)

- 同じ計算を繰り返す
- 例：xに1を3回足す

x=1

```
for i in range(3):  
    x = x + 1  
    print( x )
```

繰り返しの中で、  
毎回 print( x ) が実行

```
In [349]: x = 1  
...: for i in range(3):  
...:     x = x + 1  
...:     print(x)  
...:  
...:
```

2  
3  
4

## for (5)

- インデントによる違い
- 例：xに1を3回足す

x=1

```
for i in range(3):
```

```
    x = x + 1
```

```
print( x )
```

```
In [348]: x = 1
          ...: for i in range(3):
          ...:     x = x + 1
          ...:
          ...: print(x)
```

4

繰り返しの中で、 $x=x+1$  のみが実行され、  
`print( x )` は、最後に1回のみ実行される

## for (6)

- もちろんリストの中には数字でなくても OK

```
for i in ["A", 52, 1.7]:  
    print( i )
```

```
In [391]: for i in ["A", 52, 1.7]:  
          ...:     print(i)  
          ...:
```

```
A  
52  
1.7
```

## 練習3: for

- 練習 2 で計算したA君、B君、C君のBMIのリストをfor文を使って、順に呼び出してprint文で表示してみよう
- それぞれの身長、体重
  - A君 : 1.7m, 52kg
  - B君 : 1.6m, 75kg
  - C君 : 1.8m, 73kg

# 外部データの読み込み

CSV file

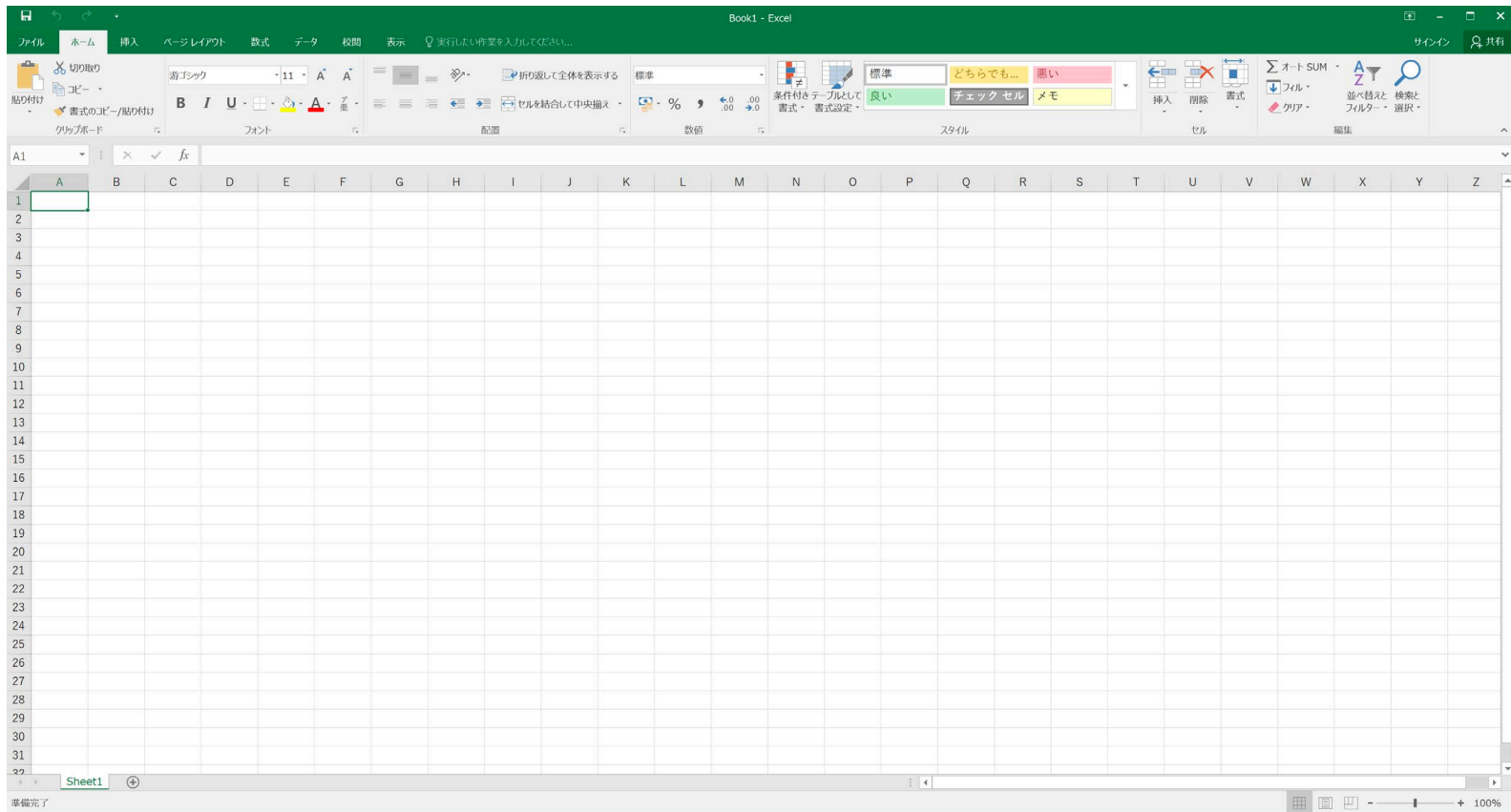
# CSV file format

- CSV = comma separated values
- File format specifications:
  - 項目ごとにコンマ “,” で区切られたデータ
  - 1行が一まとめのデータで、改行で行を識別
- Example:  
name, age, weight, height  
Bob, 40, 80, 175  
Taro, 35, 65, 170  
▪ ▪ ▪

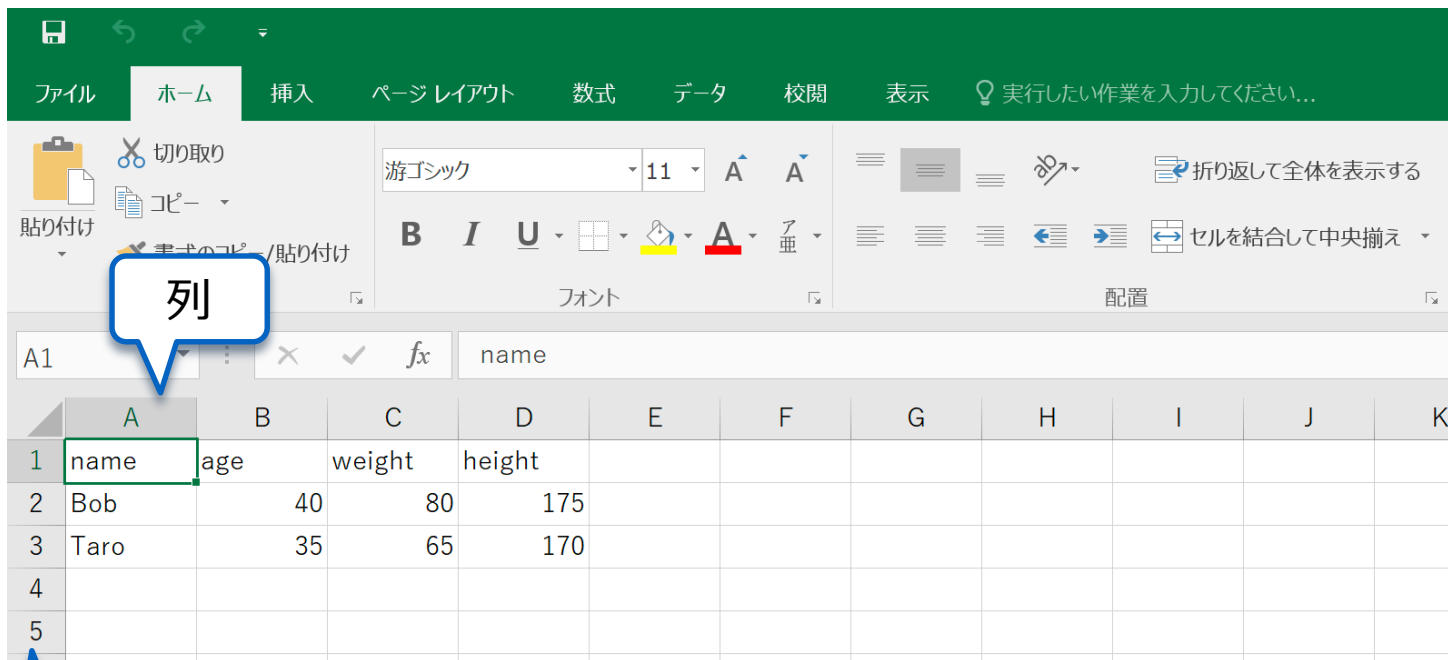


# Excel で csv ファイル

## ● Excel



# Excel で csv ファイル



The screenshot shows the Microsoft Excel interface with the 'ホーム' (Home) tab selected. The ribbon includes options for '貼り付け' (Paste), '切り取り' (Cut), 'コピー' (Copy), and '書式の貼り付け' (Paste Styles). The font settings are set to '游ゴシック' (Yu Gothic) and size 11. The ribbon also includes options for 'B' (Bold), 'I' (Italic), 'U' (Underline), and 'A' (Text Color). The ribbon also includes options for '配置' (Layout) and 'セルを結合して中央揃え' (Merge Cells and Center). The spreadsheet shows a CSV file with the following data:

	A	B	C	D	E	F	G	H	I	J	K
1	name	age	weight	height							
2	Bob	40	80	175							
3	Taro	35	65	170							
4											
5											

行

# Excel で csv ファイル

## ● CSV fileとして保存

The image shows the Excel 'File' menu on the left and the 'Save As' dialog box on the right. Annotations in blue boxes with arrows point to specific steps:

- ファイル → “名前を付けて保存”**: Points to the '名前を付けて保存' (Save As) option in the File menu.
- フォルダの選択**: Points to the '名前' (Name) field in the 'Save As' dialog box.
- ファイルフォーマットの選択**: Points to the 'ファイルの種類' (File type) dropdown menu in the 'Save As' dialog box.
- csv (カンマ区切り)**: Points to the 'CSV (カンマ区切り)' option in the file type list.

At the bottom, the text **適当に名前をつけて終了** (End after naming appropriately) is displayed.

# ExcelでCSVファイルを作ってみよう！

- Excelを開いて、次の内容を自分で書いてみよう！
- Bob, Taroに加えて、1行分、適当に自分で、データを書いて、CSVファイルとしてsaveしてみよう！

name, age, weight, height

Bob, 40, 80, 175

Taro, 35, 65, 170

(ここに好きなデータを記載)

# Fileのアップロード

Colaboratory へようこそ  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 変更を保存できませんでした

目次

- はじめに
- データサイエンス
- 機械学習
- その他のリソース
- 機械学習の例
- セクション

Colaboratory とは

Colaboratory (略称: Colab) は、ブラウザから Python

- 環境構築が不要
- GPU への無料アクセス
- 簡単に共有

Colab は、学生からデータサイエンティスト、AI リサ  
覧ください。下のリンクからすぐに使ってみることも

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

はじめに

ご覧になっているこのドキュメントは静的なウェブペ  
ラクティブな環境です。

たとえば次のコードセルには、値を計算して変数に保

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

上記のセルのコードを実行するには、セルをクリック  
ショートカット「command+return」または「Ctrl+Enter」

1 つのセルで定義した変数は、後で他のセルで使用で

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

604800

クリック

Colaboratory へようこそ  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 変更を保存できませんでした

ファイル

- bin
- boot
- content
  - drive
  - sample\_data
- datalab
- dev
- etc
- home
- lib
- lib32
- lib64
- media
- mnt
- opt
- proc
- root
- run
- sbin
- srv
- swift
- sys
- tensorflow-1.15.2
- tmp
  - tmpdyu3rlha
- tools
- usr
- var

Contentの下に  
ファイルやデータを置く

Colaboratory とは

Colaboratory (略称: Colab) は、ブラウザ

- 環境構築が不要
- GPU への無料アクセス
- 簡単に共有

Colab は、学生からデータサイエンティ  
のリンクからすぐに使って

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

はじめに

ご覧になっているこのドキュメントは静的なウェブペ  
ラクティブな環境です。

たとえば次のコードセルには、値を計算し

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

上記のセルのコードを実行するには、セル  
ショートカット「command+return」または

1 つのセルで定義した変数は、後で他のセ

# Fileのアップロード

Colaboratory へようこそ  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 変更を保存できませんでした

ファイル

- bin
- boot
- content
  - code
  - drive
  - sample\_data
- datalab
- dev
- etc
- home
- lib
- lib32
- lib64
- media
- mnt
- opt
- proc
- root
- run
- sbin
- srv
- swift
- sys
- tensorflow-1.15.2
- tmp
- tools
- usr
- var

Colaboratory とは  
Colaboratory (略称: Colab) は、ブラウザから

アップロード  
フォルダの削除  
ファイル名の変更  
新しいファイル  
新しいフォルダ  
パスをコピー  
更新

Content の下に  
フォルダを作れる

Drive is mounted at /content/drive;

```
[3] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day

86400
```

上記のセルのコードを実行するには、セルをク  
ョートカット「command+return」または「Ctr  
1つのセルで定義した変数は、後で他のセルで

Colaboratory へようこそ  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 変更を保存できませんでした

ファイル

- bin
- boot
- content
  - code
    - drive
    - sample\_data
      - README.md
      - anscombe.js
      - california\_test.csv
      - california\_train.csv

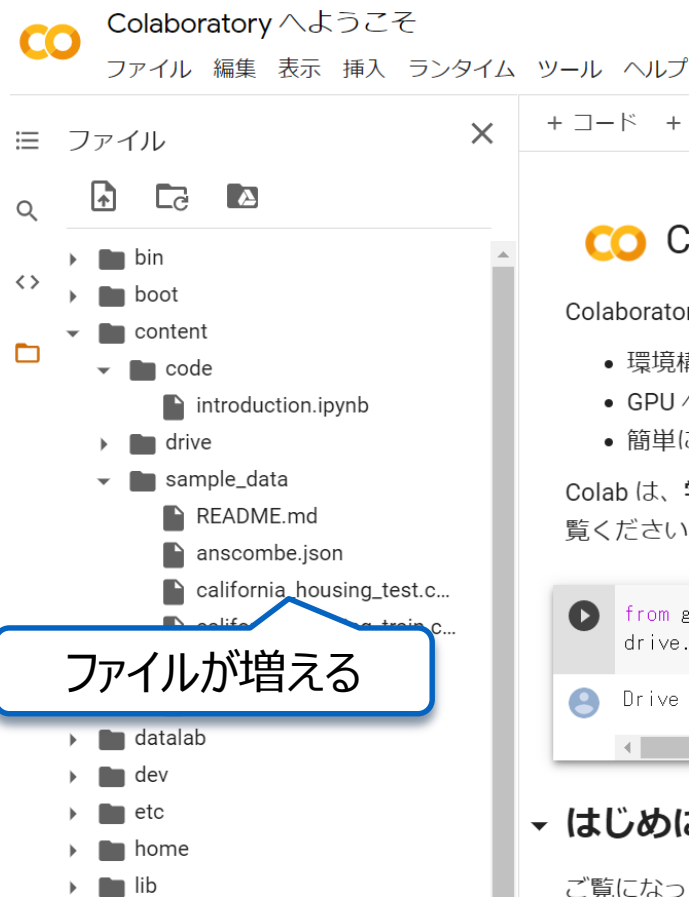
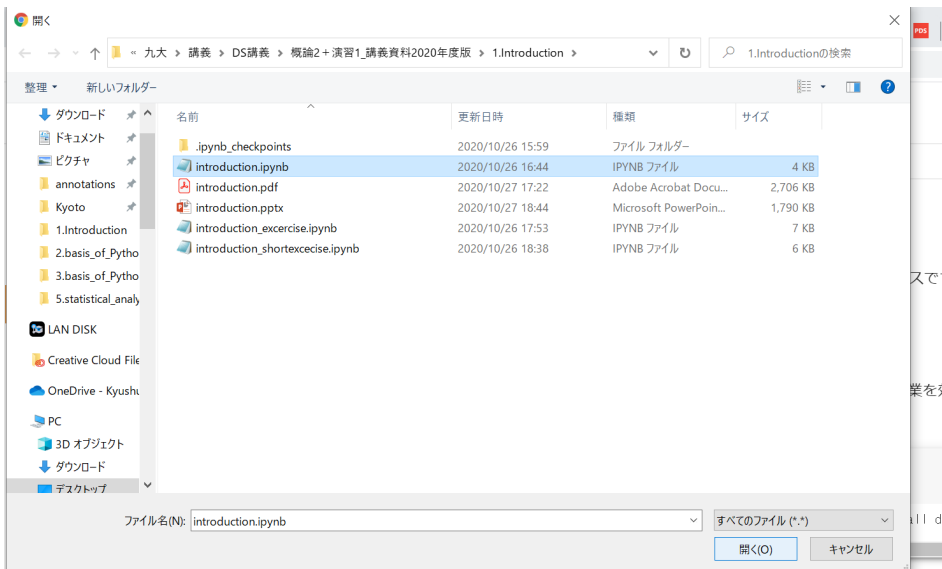
Colaboratory と  
Colaboratory (略称: Colab) は、

アップロード  
フォルダの削除  
ファイル名の変更  
新しいファイル  
新しいフォルダ  
パスをコピー  
更新

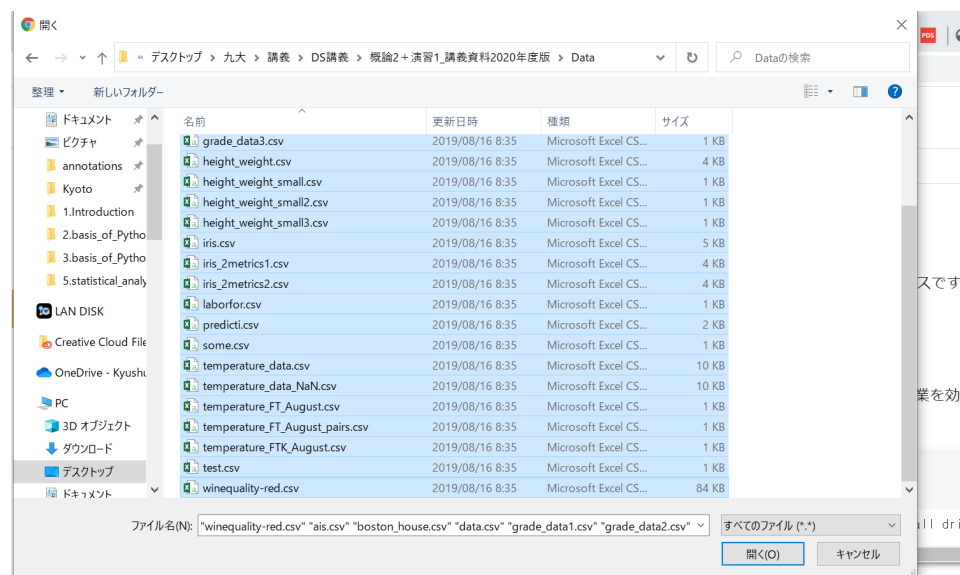
Folderの右をクリックすると  
ファイルをアップロードできる

はじめに  
ご覧になっているこのドキュメン  
ラクティブな環境です。  
たとえば次のコードセルには、値?

# Fileのアップロード



# Fileのアップロード



はじめに  
ご覧になっているこのドキュメントは静的なウェブページではなく、Colab ノートブック

複数選択すれば  
まとめてアップロード  
できる

Colaboratory へようこそ  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 変更を保存できませんでした

+ コード + テキスト ドライブにコピー

Colaboratory とは  
Colaboratory (略称: Colab) は、ブラウザから P

- 環境構築が不要
- GPU への無料アクセス
- 簡単に共有

Colab は、学生からデータサイエンティスト、A  
覧ください。下のリンクからすぐに使ってみるこ

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; t

はじめに  
ご覧になっているこのドキュメントは静的なウェ  
ラクティブな環境です。  
たとえば次のコードセルには、値を計算して変型

```
[3] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day

86400
```

上記のセルのコードを実行するには、セルをクリ  
ョートカット「command+return」または「Ctrl+

1 つのセルで定義した変数は、後で他のセルで使

```
[4] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

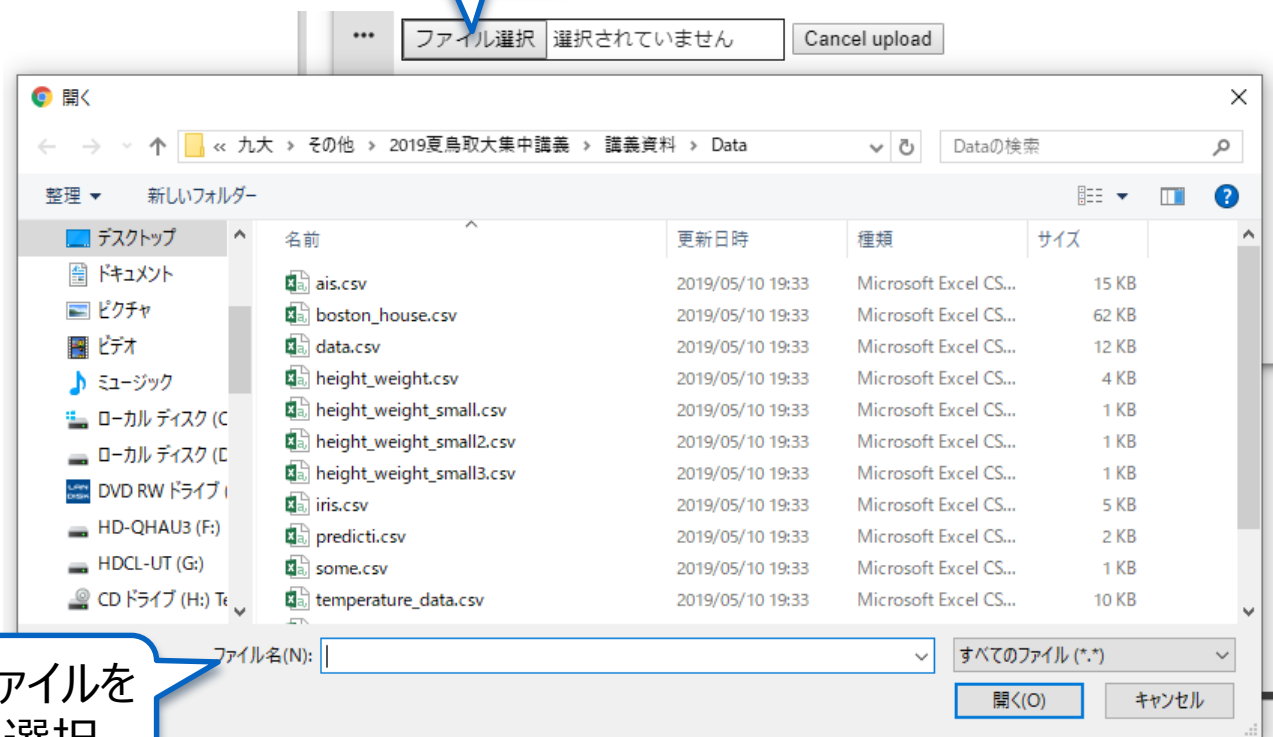
ファイルが一気に増える



# Fileのアップロード: コマンド

from google.colab import files  
files.upload()

クリック



# PythonによるCSVファイルの読み込み

- CSVファイル読み込み用の道具をPython上で呼び出す

```
import pandas as pd
```

# CSVファイルの読み込み

## ● CSVファイルを読み出すおまじない

CSVファイルの中身の  
データが格納

```
Name, age, weight, height  
Bob, 40, 80, 175  
Taro, 35, 65, 170
```

読み込むCSVファイルのパス

```
data = pd.read_csv("some.csv")
```

読み込むCSVファイルの名前

以下を事前に行うことを忘れないように！

(1) ***import pandas as pd***

(2) "some.csv" をコードと同じフォルダに置く

# データの見方1

## ●好きな行を見る

```
In [399]: print(data.iloc[0])  
name      Bob  
age        40  
weight     80  
height    175  
Name: 0, dtype: object
```

```
In [400]: print(data.iloc[0:2])  
   name  age  weight  height  
0   Bob   40     80    175  
1  Taro   35     65    170
```

↑ 0,1行目を見る

## ●好きな列を見る

```
In [401]: print(data.iloc[:,0])  
0      Bob  
1    Taro  
Name: name, dtype: object
```

```
In [402]: print(data.iloc[:,0:3])  
   name  age  weight  
0   Bob   40     80  
1  Taro   35     65
```

↑ 0,1,2列目を見る

# データの見方 2

## ● pandasデータの構造

```
In [404]: print(data.iloc[0])
```

name	Bob
age	40
weight	80
height	175

列ラベル

データ (value)

行ラベル

Name: 0, dtype: object

## ● Indexを確認

```
In [405]: print(data.columns)
Index(['name', 'age', 'weight', 'height'], dtype='object')
```

# データの見方2

## ● 項目(index)を指定してみる

↓ 項目を指定

```
In [406]: print(data.name)
0      Bob
1      Taro
Name: name, dtype: object
```

```
In [407]: print(data.height)
0      175
1      170
Name: height, dtype: int64
```

## ● 項目を指定して好きな行を見る

どちらでもO.K.

```
In [125]: print(data.iloc[0].weight)
80
```

```
In [126]: print(data.weight[1])
65
```

行を指定して項目を指定

項目を指定して行を指定

# データの見方 3

- ファイル内の文字を行ごとに読み取り
  - 行数を取得 : `len(data)`
  - 列数を取得 : `len(data.columns)`
  - 行数・列数を取得 : `data.shape`

```
In [409]: for i in range(data.shape[0]):  
          ...:     print(data.iloc[i])  
          ...:
```

列ラベル	<div>name age weight height</div>	<div>Bob 40 80 175</div>	データ (value)
------	---	--------------------------------------	-------------

行ラベル	<div>Name: 0, dtype: object</div>		
	<div>name age weight height</div>	<div>Taro 35 65 170</div>	データ (value)
	<div>Name: 1, dtype: object</div>		

# データを編集する 1

- 行を追加
  - 行ラベル,列ラベル,データを指定したSeriesを作成
  - appendで追加

データを指定

列ラベルを指定

```
In [173]: s = pd.Series(['Jiro',30,75,180],index=data.columns)
```

```
In [174]: data = data.append(s,ignore_index=True)
```

```
In [175]: data
```

```
Out[175]:
```

	name	age	weight	height
0	Bob	40	80	175
1	Taro	35	65	170
2	Jiro	30	75	180



## データを編集する 2

- 列を追加
  - 列ラベル,データを指定して追加

列ラベルを指定

データを指定

```
In [412]: data['BMI'] = [20, 30, 25]
```

```
In [413]: print(data)
```

	name	age	weight	height	BMI
0	Bob	40	80	175	20
1	Taro	35	65	170	30
2	Jiro	30	75	180	25

# データを編集する3

- 行・列を指定して値を変更

行・列を指定      データを指定

```
In [414]: data.iloc[0,4] = 40
```

```
In [415]: print(data)
```

	name	age	weight	height	BMI
0	Bob	40	80	175	40
1	Taro	35	65	170	30
2	Jiro	30	75	180	25

# CSVファイルの書き込み

- “書き込み” モードでファイルを開く

CSVファイルのデータが格納された変数

```
In [100]: data.to_csv('test.csv')
```

書き込む先のCSVファイルの名前

***import pandas*** を書くのを忘れずに！  
“test.csv”が既にあると、上書きされてしまうので注意！

## 練習 4 : CSVファイルの読み込み

- “height\_weight\_small.csv”ファイルを読み込んで、身長データを各行に順番に表示してみよう
- Height : xxx というように表示してみよう。

ヒント) `print("{}".format)` を利用しよう  
身長はHtという項目に入っています

- CSVファイルをExcelで開いてあっているか確認してみよう

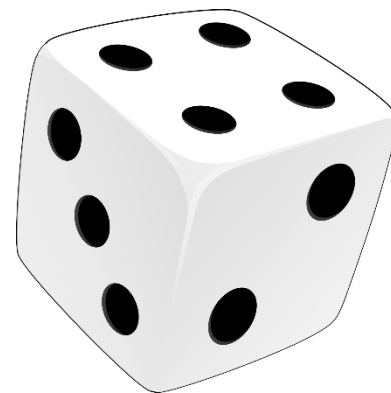
乱数

random

# 乱数（1）

- 乱数：次に何がでるかわからない数字  
例）サイコロ

1から6の整数をランダムに出す乱数



```
# 乱数を作る道具の準備  
import random
```

```
# 1から6の整数をランダムに発生  
random.randint(1,6)
```

実行するたびに違う値

```
In [4]: random.randint(1,6)  
Out[4]: 1
```

```
In [5]: random.randint(1,6)  
Out[5]: 4
```

```
In [6]: random.randint(1,6)  
Out[6]: 5
```

## 乱数（2）

- 1~20の目を出すサイコロを作ろう！



#乱数を作る道具の準備  
`import random`

#1から20の整数をランダムに発生  
`random.randint(1,20)`

```
In [9]: random.randint(1,20)
```

```
Out[9]: 9
```

```
In [10]: random.randint(1,20)
```

```
Out[10]: 20
```

```
In [11]: random.randint(1,20)
```

```
Out[11]: 5
```

# 乱数 (3)

`random.random()` # 0 以上 1 未満の実数値を返す

```
In [13]: random.random()  
Out[13]: 0.7420737840070214
```

```
In [14]: random.random()  
Out[14]: 0.15813715204502699
```

```
In [15]: random.random()  
Out[15]: 0.5928004280279405
```

`random.uniform(1.0, 10.0)` # 1 以上 10以下の実数値を返す

```
In [17]: random.uniform(1.0,10.0)  
Out[17]: 7.301516051731666
```

```
In [18]: random.uniform(1.0,10.0)  
Out[18]: 9.145636086942584
```



# break

- 繰り返しを強制的に終了させる
- 例：くじ引き（当りが出たら終わり）

当選確率は  $1/10$

```
for i in range(5):  
    number = random.randint(1, 10)  
    if number == 1:  
        print "atari !"  
        break  
    else:  
        print "hazure"
```

## 練習5: くじ引き

- 当選確率が $1/5$ のくじ引きを作ろう
- ヒント) `random.randint(1,5)`

# 関数による数値計算

Functions in Python

# 「材料」と「道具」(1)

材料 (変数)



道具 (関数)

切る



皮をむく



煮る

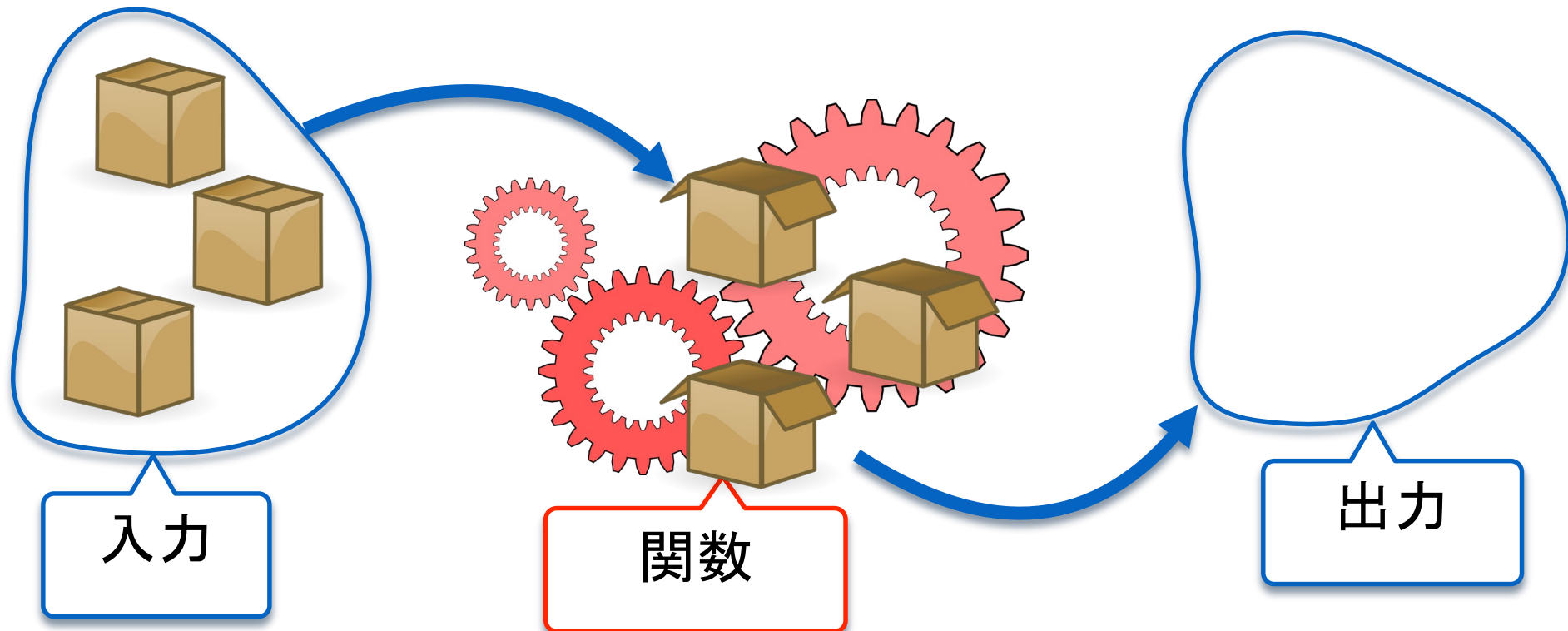


print や  
+や- も  
道具の一種



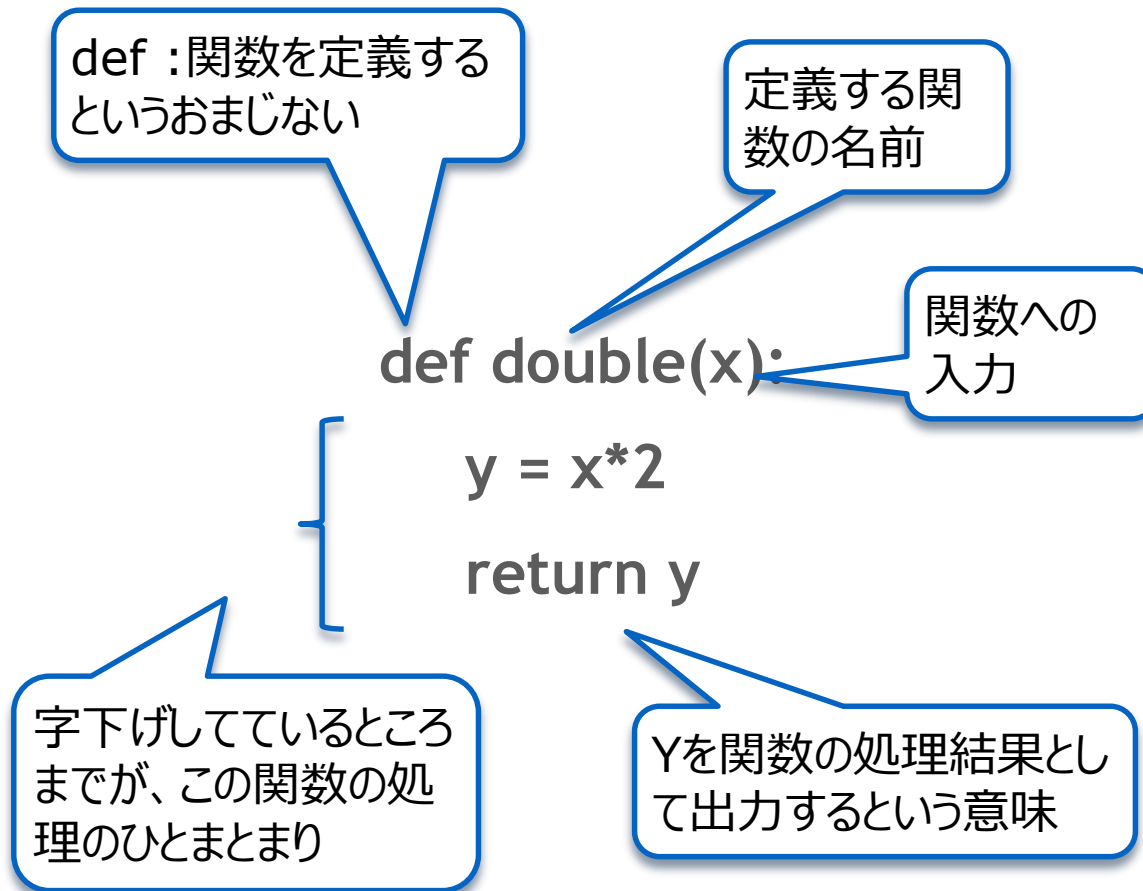
# 関数による数値計算

- 関数は、入力に対して決まった処理をする。
- 関数内部での計算結果を出力することもできる
- 用意されている関数だけでなく、自分で好きな関数を作れる



# Pythonでの関数定義

- Xを入力したらその値を2倍した数を入力する関数



# 関数の定義(コンソール)

## ●セルへ入力

```
>>> def double(x):  
...     '''Put your comments  
...     example: the function returns the double of the input'''  
...     return x*2  
... 
```

“tab” ボタンで字下げするのを忘れずに！

# 関数の利用

- 入力として数字を入れて、実行

```
>>> double(3)  
6
```

- 関数の出力を変数に格納

```
>>> a = 2  
>>> y = double(a)  
>>> print(y)  
4
```



# 関数の利用：練習

- 数字を3倍にする関数tripleを定義してみよう！

```
def triple(x):  
    '''Put your comments  
    example: the function returns the triple of the input'''  
    y = x*3  
    return y
```

```
In [43]: import DoubleFunc  
        ...: DoubleFunc.triple(5)  
        ...:  
Out[43]: 15
```



# 入力と出力



- Pythonでは, **入力** と **出力** は型をもったオブジェクト
- 数値だけでなく、文字列や他のデータ構造もO.K.
- 入力も複数の変数を入力できる
- 数値 : 0, 1, 5.5 ...
- 文字列 : "Hello World"
- データ構造 : Array, list, dictionaries ...

# 関数の利用：練習

- 2つの文字を間に空白を入れてつなげる関数

```
def connectStr(str1, str2):  
    a = str1 + ' ' + str2  
    return a
```

- 関数の呼び出し

```
In [20]: a = connectStr('Hello', 'World!')
```

```
In [21]: print(a )  
Hello World!
```

# 関数の利用：練習

- 2つのリストをつなげる関数

```
def connectList(list1, list2):  
    a = list1 + list2  
    return a
```

- 関数の呼び出し

```
In [26]: a = connectList([0,1,2],[1,2,3,4])
```

```
In [27]: print(a )  
[0, 1, 2, 1, 2, 3, 4]
```

# help 関数

- 予め関数の説明を'''xxxx'''で囲って記載

```
def double(x):  
    '''this function multiplies its input by 2  
    input: x (number)  
    output: the value that x times 2 (number)'''  
    y = x*2  
    return y
```

- help(double):

```
In [34]: help(double)  
Help on function double in module __main__:
```

```
double(x)  
    this function multiplies its input by 2  
    input: x (number)  
    output: the value that x times 2 (number)
```

この部分が  
表示される

## 定義した関数の利用(2)

- DoubleFuncの中から利用する関数だけ、使えるように呼び出す

```
>>> from DoubleFunc import double  
>>> double(3)  
6
```

- Pythonであらかじめ用意されている関数と同じ名前は避けよう！

## 練習 6 : 関数定義と呼び出し

- 入力 $x$ を10倍する関数tentimesを定義しよう
- tentimesを呼び出して、入力に5を与えて計算しよう

# 演習資料

Playing with the Body-mass index



# 演習 1 : for

- 1から10まで順に足して行くプログラムをforを使って書いてみよう。また、それぞれの計算結果をprint文で出力してみよう

ヒント)  $0+1, 1+2, 1+2+3, \dots,$

$1+2+3+4+5+6+7+8+9+10$

- 上記の問題で最後の1～10まで全部足した結果だけをprint文で出力してみよう

Answer

## 演習 2 : CSVの読み込み・書き込み + for

- “height\_weight\_small.csv”ファイルを読み込んで、各行のBMIを計算して、BMIを表示しよう。（BMIは四捨五入）
  - 行番号 : BMI : xxx
  - ヒント 1 ) 身長はcm単位になっているので、m単位に変換しよう（100で割る）。
  - ヒント 2 ) 四捨五入はround()を利用
- 進みが早い人用
  - BMIを列に追加して、CSVファイルへ書き込もう。
    - ヒント) data2=data  
data['BMI']=0 ← 新しい列を追加
  - “height\_weight\_small2.csv”というファイル名にしよう。

Answer

## 演習3: CSVの読み込み + for + if

- “height\_weight\_small.csv”ファイルを読み込んで、各行のBMIを計算して、BMIの値に応じて次のように表示するプログラムを作成しよう
  - 18.5未満の場合、「やせ型(slender)」と表示
  - 18.5以上25未満の場合、「標準(normal)」と表示
  - 25以上の場合、「肥満気味(fat)」と表示
- 進みが早い人用
  - “BodyShape”を列に追加して、CSVファイルへ書き込もう。(データ追加には、英文字を利用)

## 演習 4

- BMIを計算する関数を定義しよう！
- 定義した関数を用いて演習 3 の内容を実現するプログラムを作成しよう

# アドバンス 1 (進みが早い人用)

A simple game in Python (the random game)

## 演習5：宝くじ

- 宝くじって儲かるの！？調べてみよう！

等	当選金額	確率
1等	5億円	1/1000万
2等	1億円	2/1000万
3等	1000万円	4/1000万
4等	10万円	1/10万
5等	3000円	1/100
6等	300円	1/10

※ 実際の宝くじとは異なります



# 演習：宝くじ

- takarakuji.py を開く
- 大体いくらくらいになるか調べてみよう
- 1等が出るまで引き続けるまで引いてみよう
- 持ち金を変えてやってみよう
- 1円でも儲かったらやめる！

# アドバンス2 (進みが早い人用)

A simple game in Python (the random game)

# For で遊ぼうバトル10回

- ユーザが0と1を入力して、ランダム値を当てること  
ができれば(同じ値になったら)HPが増えるゲーム

```
hp = 0
```

```
for i in range(10): #以下を10回繰り返し
```

```
    rand = random.randint(0, 1) #0か1をランダムで生成
```

```
    print("please input 0 or 1")
```

```
    inp = int(input()) # ユーザの入力を読み込み
```

```
    if inp==rand: #ユーザの入力とランダム値が一致していたら、
```

```
        hp = hp + 1 #hpを1増やす
```

```
    print("Your HP = " + str(hp)) #現在のhpを表示
```

# 減ることもある

- ユーザが0と1を入力して、ランダムのを当てることができれば (同じ値になったら)HPが増え、間違うと減るゲーム

```
hp = 0
```

```
for i in range(10): # 10回繰り返し
```

```
    rand = random.randint(0, 1) #ランダムで0か1を生成
```

```
    print( "please input 0 or 1" )
```

```
    inp = int(input()) #ユーザの入力の読み込み
```

```
    if inp==rand: #ユーザの入力とランダム値が一致していたら、
```

```
        hp = hp + 1 #hpを1増やす
```

```
    else #間違ったら
```

```
        hp = hp - 1 #hpを1減らす
```

```
print( "Your HP = " + str(hp) ) #現在のhpを表示
```

# 増減量もランダム

- ユーザが0と1を入力して、ランダム値を当てることができたら、ランダムな量HPが増え、間違えるとランダムに減るゲーム

```
hp = 0
```

```
for i in range(10): # 10回繰り返し
```

```
    rand = random.randint(0, 1) #ランダムで0か1を生成
```

```
    print("please input 0 or 1")
```

```
    inp = int(input()) #ユーザの入力の読み込み
```

```
    if inp==rand: #ユーザの入力とランダム値が一致していたら、
```

```
        hp = hp + random.randint(0, 9) #hpを0~9ランダムに増やす  
    else #間違ったら
```

```
        hp = hp - random.randint(0, 9) #hpをランダムに減らす
```

```
    print("Your HP = " + str(hp)) #現在のhpを表示
```

# HPがマイナスになったらゲームオーバー

## ● 同じルールで、HPがマイナスになったらゲームオーバー

```
hp = 0
```

```
for i in range(10): # 10回繰り返し
```

```
    rand = random.randint(0, 1) # ランダムで0か1を生成
```

```
    print( "please input 0 or 1" )
```

```
    inp = int(input()) # ユーザの入力の読み込み
```

```
    if inp==rand: # ユーザの入力とランダム値が一致していたら、
```

```
        hp = hp + random.randint(0, 9) # hpを0~9ランダムに増やす
```

```
    else # 間違ったら
```

```
        hp = hp - random.randint(0, 9) # hpをランダムに減らす
```

```
    print( "Your HP = " + str(hp) ) # 現在のhpを表示
```

```
    if hp < 0: # HPが0以下になったら
```

```
        print( "Game Over!" ) # ゲームオーバーと表示して、
```

```
        break # 繰り返しを終了
```

## おまけ：興味ある人はバトルを作ってみよう

- HPを敵(コンピュータ)、自分と2人分設定して、
- 自分のターンでは、敵のHPをランダムに減らし、
- コンピュータのターンでは、自分のHPをランダムに減らす
- どちらかのHPがなくなるまで続ける

# おまけ：興味ある人はバトルを作ってみよう

## 応用

- 防御力、攻撃力等のパラメータを振ってその値によって、HPの減らし方を変える。
  - 例1)  $a$ =自分の攻撃力 3、 $d$ =敵の防御力 2
$$hp = hp - \text{random.randint}(0+a-b, 9+a-b)$$
$$\Rightarrow hp = hp - \text{random.randint}(1, 10)$$
  - 例2)  $a$ =自分の攻撃力5、 $d$ =敵の防御力1
$$hp = hp - \text{random.randint}(0+a-b, 9+a-b)$$
$$\Rightarrow hp = hp - \text{random.randint}(4, 13)$$
- 攻撃力が敵の防御力より高いとより多くのHPを減らしやすい