

---

**project**

**horoiwa**

**2022 年 09 月 19 日**

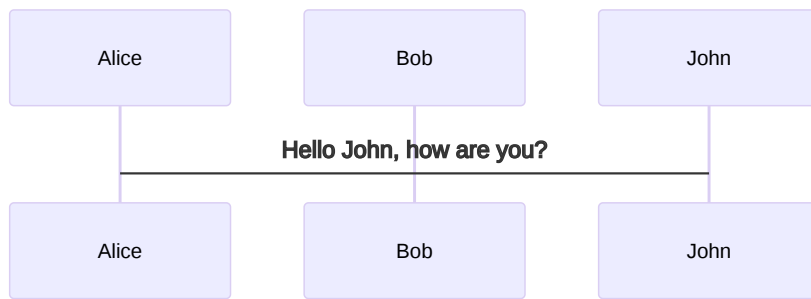


# 目次

第 1 章	Mermaid integration	1
第 2 章	Jupyter integration	3
2.1	Example notebook . . . . .	3
第 3 章	Use markdown	5
3.1	Markdown Basic Syntax . . . . .	5
第 4 章	Modules	7
4.1	package . . . . .	7
Python モジュール索引		13
索引		15



## 第1章 Mermaid integration



## 第2章 Jupyter integration

### 2.1 Example notebook

```
[2]: import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df
```

```
[2]:      sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0              5.1             3.5             1.4             0.2
1              4.9             3.0             1.4             0.2
2              4.7             3.2             1.3             0.2
3              4.6             3.1             1.5             0.2
4              5.0             3.6             1.4             0.2
..              ...              ...              ...              ...
145             6.7             3.0             5.2             2.3
146             6.3             2.5             5.0             1.9
147             6.5             3.0             5.2             2.0
148             6.2             3.4             5.4             2.3
149             5.9             3.0             5.1             1.8

[150 rows x 4 columns]
```





## 第3章 Use markdown

### 3.1 Markdown Basic Syntax

#### 3.1.1 Headings

To create a heading, add number signs (#) in front of a word or phrase. The number of number signs you use should correspond to the heading level. For example, to create a heading level three...



## 第4章 Modules

### 4.1 package

#### 4.1.1 package package

##### Subpackages

`package.common package`

##### Submodules

`package.common.cache_tools module`

`package.common.cache_tools.daily_cache(func)`

`package.common.cache_tools.monthly_cache(func)`

`package.common.cache_tools.serialized_cache(salt: str, maxsize: int = 64)`

`package.common.cache_tools.weekly_cache(func)`

`package.common.modelCV module`

```
class package.common.modelCV.BaseModelCV(n_splits: int = 3, n_repeats: int = 1, metric_func:
                                          Optional[Callable] = None, direction: Literal['minimize',
                                          'maximize'] = 'maximize')
```

ベースクラス: ABC

```
abstract define_search_space(trial)
```

```
fit(X: DataFrame, Y: Series, n_trials: int = 150)
```

```
abstract property model_class
```

```
class package.common.modelCV.LGBCV(n_splits: int = 3, n_repeats: int = 1, metric_func:
                                     Optional[Callable] = None, direction: Literal['minimize',
                                     'maximize'] = 'maximize')
```

ベースクラス: `BaseModelCV`

```
define_search_space(trial)
```

```
property model_class
```

```
class package.common.modelCV.LGBRCV(n_splits: int = 3, n_repeats: int = 1, metric_func:
    Optional[Callable] = None, direction: Literal['minimize',
    'maximize'] = 'maximize')
```

```
ベースクラス: BaseModelCV
```

```
define_search_space(trial)
```

```
property model_class
```

```
class package.common.modelCV.RidgeCV(n_splits: int = 3, n_repeats: int = 1, metric_func:
    Optional[Callable] = None, direction: Literal['minimize',
    'maximize'] = 'maximize')
```

```
ベースクラス: BaseModelCV
```

```
define_search_space(trial)
```

```
property model_class
```

```
class package.common.modelCV.SVCCV(n_splits: int = 3, n_repeats: int = 1, metric_func:
    Optional[Callable] = None, direction: Literal['minimize',
    'maximize'] = 'maximize')
```

```
ベースクラス: BaseModelCV
```

```
define_search_space(trial)
```

```
property model_class
```

```
class package.common.modelCV.SVRCV(n_splits: int = 3, n_repeats: int = 1, metric_func:
    Optional[Callable] = None, direction: Literal['minimize',
    'maximize'] = 'maximize')
```

```
ベースクラス: BaseModelCV
```

```
define_search_space(trial)
```

```
property model_class
```

## **package.common.timer module**

```
class package.common.timer.Timer(tag: str)
```

```
ベースクラス: object
```

```
Example with Timer(tag="proc_a"):
```

```
    proc_a()
```

## Module contents

`package.module1 package`

## Subpackages

`package.module1.tests package`

## Submodules

`package.module1.tests.test_api module`

## Module contents

## Submodules

`package.module1.api module`

`package.module1.api.say_spam(n: int)`

sample function

パラメータ

`n(int) -- _description_`

例外

`TypeError -- _description_`

戻り値

`_description_`

戻り値の型

`_type_`

`package.module1.config module`

`class package.module1.config.MD1Config(alpha: float, beta: int)`

ベースクラス: `object`

`alpha: float`

`beta: int`

package.module1.main module

package.module1.models module

package.module1.postprocess module

package.module1.preprocess module

## Module contents

### Submodules

package.constants module

path などの定数や aws/gcp クライアントのようなグローバルにひとつ存在すればよい stateless な変数やインスタンスを配置

**class** package.constants.ColoredStreamHandler(*stream=None*)

ベースクラス: StreamHandler

```
cmap = {'ALERT': '\x1b[0;37;41mALERT\x1b[0m', 'CRITICAL':
'\x1b[0;37;41mCRITICAL\x1b[0m', 'DEBUG': '\x1b[0;36mDEBUG\x1b[0m', 'ERROR':
'\x1b[0;31mERROR\x1b[0m', 'INFO': '\x1b[0;32mINFO\x1b[0m', 'TRACE': '[TRACE]',
'WARN': '\x1b[0;33mWARN\x1b[0m', 'WARNING': '\x1b[0;33mWARN\x1b[0m']}
```

**emit**(*record: LogRecord*) → None

Emit a record.

If a formatter is specified, it is used to format the record. The record is then written to the stream with a trailing newline. If exception information is present, it is formatted using `traceback.print_exception` and appended to the stream. If the stream has an 'encoding' attribute, it is used to determine how to do the output to the stream.

**class** package.constants.Config(*strategy: Literal['GP-EI', 'TPE', 'CMA-ES'], date\_start: datetime.date, MD1: package.module1.config.MD1Config*)

ベースクラス: object

**MD1:** *MD1Config*

**date\_start:** date

**strategy:** *Literal['GP-EI', 'TPE', 'CMA-ES']*

package.constants.getChildLogger(*name*)

package.constants.load\_config()

package.constants.print(*message: str, level: str = 'info'*)

## **package.exceptions module**

**exception** package.exceptions.UserDefinedException

ベースクラス: Exception

Example user defined exception

メモ

状況：どのような状況で発生するか？ 解決：どうすれば解決できるか？

## **package.main module**

package.main.**main**(msg: str)

## **package.sub module**

package.sub.**sub**()

## **Module contents**





# Python モジュール索引

## p

`package`, 11

`package.common`, 9

`package.common.cache_tools`, 7

`package.common.modelCV`, 7

`package.common.timer`, 8

`package.constants`, 10

`package.exceptions`, 11

`package.main`, 11

`package.module1`, 10

`package.module1.api`, 9

`package.module1.config`, 9

`package.module1.main`, 10

`package.module1.models`, 10

`package.module1.postprocess`, 10

`package.module1.preprocess`, 10

`package.module1.tests`, 9

`package.sub`, 11



# 索引

- alpha (*package.module1.config.MD1Config* の属性), 9
- BaseModelCV (*package.common.modelCV* のクラス), 7
- beta (*package.module1.config.MD1Config* の属性), 9
- cmap (*package.constants.ColoredStreamHandler* の属性), 10
- ColoredStreamHandler (*package.constants* のクラス), 10
- Config (*package.constants* のクラス), 10
- daily\_cache() (*package.common.cache\_tools* モジュール), 7
- date\_start (*package.constants.Config* の属性), 10
- define\_search\_space() (*package.common.modelCV.BaseModelCV* のメソッド), 7
- define\_search\_space() (*package.common.modelCV.LGBCV* のメソッド), 7
- define\_search\_space() (*package.common.modelCV.LGBRCV* のメソッド), 8
- define\_search\_space() (*package.common.modelCV.RidgeCV* のメソッド), 8
- define\_search\_space() (*package.common.modelCV.SVCCV* のメソッド), 8
- define\_search\_space() (*package.common.modelCV.SVRCV* のメソッド), 8
- emit() (*package.constants.ColoredStreamHandler* のメソッド), 10
- fit() (*package.common.modelCV.BaseModelCV* のメソッド), 7
- getChildLogger() (*package.constants* モジュール), 10
- LGBCV (*package.common.modelCV* のクラス), 7
- LGBRCV (*package.common.modelCV* のクラス), 8
- load\_config() (*package.constants* モジュール), 10
- main() (*package.main* モジュール), 11
- MD1 (*package.constants.Config* の属性), 10
- MD1Config (*package.module1.config* のクラス), 9
- model\_class (*package.common.modelCV.BaseModelCV* のプロパティ), 7
- model\_class (*package.common.modelCV.LGBCV* のプロパティ), 8
- model\_class (*package.common.modelCV.LGBRCV* のプロパティ), 8
- model\_class (*package.common.modelCV.RidgeCV* のプロパティ), 8
- model\_class (*package.common.modelCV.SVCCV* のプロパティ), 8
- model\_class (*package.common.modelCV.SVRCV* のプロパティ), 8
- monthly\_cache() (*package.common.cache\_tools* モジュール), 7
- package
  - モジュール, 11
- package.common
  - モジュール, 9
- package.common.cache\_tools
  - モジュール, 7
- package.common.modelCV
  - モジュール, 7
- package.common.timer
  - モジュール, 8
- package.constants
  - モジュール, 10
- package.exceptions
  - モジュール, 11
- package.main
  - モジュール, 11
- package.module1
  - モジュール, 10
- package.module1.api

モジュール, 9  
package.module1.config  
    モジュール, 9  
package.module1.main  
    モジュール, 10  
package.module1.models  
    モジュール, 10  
package.module1.postprocess  
    モジュール, 10  
package.module1.preprocess  
    モジュール, 10  
package.module1.tests  
    モジュール, 9  
package.sub  
    モジュール, 11  
print() (*package.constants* モジュール), 10  
  
RidgeCV (*package.common.modelCV* のクラス), 8  
  
say\_spam() (*package.module1.api* モジュール), 9  
serialized\_cache() (*package.common.cache\_tools*  
    モジュール), 7  
strategy (*package.constants.Config* の属性), 10  
sub() (*package.sub* モジュール), 11  
SVCCV (*package.common.modelCV* のクラス), 8  
SVRCV (*package.common.modelCV* のクラス), 8  
  
Timer (*package.common.timer* のクラス), 8  
  
UserDefinedException, 11  
  
weekly\_cache() (*package.common.cache\_tools* モジ  
    ュール), 7  
  
モジュール  
    package, 11  
    package.common, 9  
    package.common.cache\_tools, 7  
    package.common.modelCV, 7  
    package.common.timer, 8  
    package.constants, 10  
    package.exceptions, 11  
    package.main, 11  
    package.module1, 10  
    package.module1.api, 9  
    package.module1.config, 9  
    package.module1.main, 10  
    package.module1.models, 10  
    package.module1.postprocess, 10  
    package.module1.preprocess, 10  
    package.module1.tests, 9  
    package.sub, 11