PyYAML BEST PRACTICES

PyYAML BEST PRACTICES

Start with importing theyamlpackage.

```
>>> import yaml
```

Warning: It is not safe to callyaml.loadwith any data received from an untrusted source!YAML.loadis as powerful aspickle.loadand so may call any Python function.

Yaml implementation:

1.The functionyaml.loadconverts a YAML document to a Python object.

```
>>> yaml.load("""
... - Hesperiidae
... - Papilionidae
... - Apatelodidae
... - Epiplemidae
... """)


OUTPUT: ['Hesperiidae', 'Papilionidae', 'Apatelodidae', 'Epiplemidae']
```

2.yaml.loadaccepts a byte string, a Unicode string, an open binary file object, or an open text file object. A byte string or a file must be encoded with*utf-8*,*utf-16-be*or*utf-16-le*encoding.yaml.loaddetects the encoding by checking the*BOM*(byte order mark) sequence at the beginning of the string/file. If no*BOM*is present, the*utf-8*encoding is assumed.

yaml.loadreturns a Python object.

```
>>> yaml.load(u"""
... hello: !
... """)    # In Python 3, do not use the 'u' prefix


OUTPUT: {'hello': u'\u041f\u0440\u0438\u0432\u0435\u0442!'}


>>> stream = file('document.yaml', 'r')    # 'document.yaml' contains a single YAML document.


>>> yaml.load(stream)
[...]    # A Python object corresponding to the document.
```

3.If a string or a file contains several documents, you may load them all with theyaml.load_allfunction.

```
>>> documents = """
... ---
... name: The Set of Gauntlets 'Pauraegen'
... description: >
...     A set of handgear with sparks that crackle
...     across its knuckleguards.
... ---
... name: The Set of Gauntlets 'Paurnen'
... description: >
...   A set of gauntlets that gives off a foul,
```

```
...   acrid odour yet remains untarnished.
... ---
... name: The Set of Gauntlets 'Paurnimmen'
... description: >
...   A set of handgear, freezing with unnatural cold.
... """

>>> for data in yaml.load_all(documents):
...     print data

OUTPUT: {'description': 'A set of handgear with sparks that crackle across its knuckleguards.\n',
'name': "The Set of Gauntlets 'Pauraegen'"}
{'description': 'A set of gauntlets that gives off a foul, acrid odour yet remains untarnished.\n',
'name': "The Set of Gauntlets 'Paurnen'"}
{'description': 'A set of handgear, freezing with unnatural cold.\n',
'name': "The Set of Gauntlets 'Paurnimmen'"}
```

4.PyYAML allows you to construct a Python object of any type.

```
>>> yaml.load("""
... none: [~, null]
... bool: [true, false, on, off]
... int: 42
... float: 3.14159
... list: [LITE, RES_ACID, SUS_DEXT]
... dict: {hp: 13, sp: 5}
... """)

OUTPUT: {'none': [None, None], 'int': 42, 'float': 3.1415899999999999,
'list': ['LITE', 'RES_ACID', 'SUS_DEXT'], 'dict': {'hp': 13, 'sp': 5},
'bool': [True, False, True, False]}
```

5.Even instances of Python classes can be constructed using the!!python/objecttag.

```
>>> class Hero:
...     def __init__(self, name, hp, sp):
...         self.name = name
...         self.hp = hp
...         self.sp = sp
...     def __repr__(self):
...         return "%s(name=%r, hp=%r, sp=%r)" % (
...             self.__class__.__name__, self.name, self.hp, self.sp)

>>> yaml.load("""
... !!python/object:__main__.Hero
```

... name: Welthyr Syxgon

... hp: 1200

... sp: 0

... """)

Hero(name='Welthyr Syxgon', hp=1200, sp=0)

Note: The ability to construct an arbitrary Python object may be dangerous if you receive a YAML document from an untrusted source such as the Internet. The functionyaml.safe_loadlimits this ability to simple Python objects like integers or lists.

A python object can be marked as safe and thus be recognized byyaml.safe_load. To do this, derive it fromyaml.YAMLObject(as explained in section*Constructors, representers, resolvers*) and explicitly set its class propertyyaml_loadertoyaml.SafeLoader.

- yaml.load(stream, SafeLoader)

Recommended for untrusted input. Limitation: Loads a subset of the YAML language.

- yaml.load(stream, FullLoader)

For more trusted input. Still a bit of Limitation: Avoids arbitrary code execution.

- yaml.load(stream, Loader)(UnsafeLoaderis the same asLoader)

Unsafe. But has the full power.

REFERENCES: DO REFER FOR MORE DETAILED EXPLANATION

https://pyyaml.org/wiki/PyYAMLDocumentation

https://github.com/yaml/pyyaml/issues/292

https://github.com/yaml/pyyaml/issues/265