

Items

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 7)

Unknown directive type "module".

```
.. module:: scrapy.item
   :synopsis: Item and Field classes
```

The main goal in scraping is to extract structured data from unstructured sources, typically, web pages. [ref: Spiders <topics-spiders>](#) may return the extracted data as *items*, Python objects that define key-value pairs.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 10); [backlink](#)

Unknown interpreted text role "ref".

Scrapy supports [ref: multiple types of items <item-types>](#). When you create an item, you may use whichever type of item you want. When you write code that receives an item, your code should [ref: work for any item type <supporting-item-types>](#).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 14); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 14); [backlink](#)

Unknown interpreted text role "ref".

Item Types

Scrapy supports the following types of items, via the [itemadapter](#) library: [ref: dictionaries <dict-items>](#), [ref: Item objects <item-objects>](#), [ref: dataclass objects <dataclass-items>](#), and [ref: attrs objects <attrs-items>](#).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 24); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 24); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 24); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 24); [backlink](#)

Unknown interpreted text role "ref".

Dictionaries

As an item type, `:class:`dict`` is convenient and familiar.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 35); [backlink](#)

Unknown interpreted text role "class".

Item objects

`:class:`Item`` provides a `:class:`dict``-like API plus additional features that make it the most feature-complete item type:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 42); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 42); [backlink](#)

Unknown interpreted text role "class".

`:class:`Item`` objects replicate the standard `:class:`dict`` API, including its `__init__` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 48); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 48); [backlink](#)

Unknown interpreted text role "class".

`:class:`Item`` allows defining field names, so that:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 51); [backlink](#)

Unknown interpreted text role "class".

- `:class:`KeyError`` is raised when using undefined field names (i.e. prevents typos going unnoticed)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 53); [backlink](#)

Unknown interpreted text role "class".

- `:ref:`Item exporters <topics-exporters>`` can export all fields by default even if the first scraped object does not have values for all of them

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 56); [backlink](#)

Unknown interpreted text role "ref".

`:class:`Item`` also allows defining field metadata, which can be used to `:ref:`customize serialization <topics-exporters-field-serialization>``.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 60); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 60); [backlink](#)

Unknown interpreted text role "ref".

`:mod:`trackref`` tracks `:class:`Item`` objects to help find memory leaks (see `:ref:`topics-leaks-trackrefs``).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 63); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 63); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 63); [backlink](#)

Unknown interpreted text role "ref".

`:class:`Item`` objects also provide the following additional API members:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 66); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 68)

Unknown directive type "automethod".

```
.. automethod:: copy
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 70)

Unknown directive type "automethod".

```
.. automethod:: deepcopy
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 72)

Unknown directive type "attribute".

```
.. attribute:: fields
```

```
A dictionary containing *all declared fields* for this Item, not only
those populated. The keys are the field names and the values are the
:class:`Field` objects used in the :ref:`Item declaration
<topics-items-declaring>`.
```

Example:

```
from scrapy.item import Item, Field

class CustomItem(Item):
    one_field = Field()
    another_field = Field()
```

Dataclass objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 92)

Unknown directive type "versionadded".

```
.. versionadded:: 2.2
```

`:func:`~dataclasses.dataclass`` allows defining item classes with field names, so that `:ref:`item exporters <topics-exporters>`` can export all fields by default even if the first scraped object does not have values for all of them

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 94); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 94); [backlink](#)

Unknown interpreted text role "ref".

Additionally, `dataclass` items also allow to:

- define the type and default value of each defined field.
- define custom field metadata through `:func:`dataclasses.field``, which can be used to `:ref:`customize serialization <topics-exporters-field-serialization>``.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 102); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 102); [backlink](#)

Unknown interpreted text role "ref".

They work natively in Python 3.7 or later, or using the [dataclasses backport](#) in Python 3.6.

Example:

```
from dataclasses import dataclass

@dataclass
class CustomItem:
    one_field: str
    another_field: int
```

Note

Field types are not enforced at run time.

attr.s objects

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 126)

Unknown directive type "versionadded".

```
.. versionadded:: 2.2
```

`:func:`attr.s`` allows defining item classes with field names, so that `:ref:`item exporters <topics-exporters>`` can export all fields by default even if the first scraped object does not have values for all of them.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 128); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 128); [backlink](#)

Unknown interpreted text role "ref".

Additionally, `attr.s` items also allow to:

- define the type and default value of each defined field.
- define custom field `:ref:metadata <attrs.metadata>`, which can be used to `:ref:customize serialization <topics-exporters-field-serialization>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 136); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 136); [backlink](#)

Unknown interpreted text role "ref".

In order to use this type, the `:doc:attrs package <attrs.index>` needs to be installed.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 139); [backlink](#)

Unknown interpreted text role "doc".

Example:

```
import attr

@attr.s
class CustomItem:
    one_field = attr.ib()
    another_field = attr.ib()
```

Working with Item objects

Declaring Item subclasses

Item subclasses are declared using a simple class definition syntax and `:class:Field` objects. Here is an example:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 159); [backlink](#)

Unknown interpreted text role "class".

```
import scrapy

class Product(scrapy.Item):
    name = scrapy.Field()
    price = scrapy.Field()
    stock = scrapy.Field()
    tags = scrapy.Field()
    last_updated = scrapy.Field(serializer=str)
```

Note

Those familiar with [Django](#) will notice that Scrapy Items are declared similar to [Django Models](#), except that Scrapy Items are much simpler as there is no concept of different field types.

Declaring fields

`:class:Field` objects are used to specify metadata for each field. For example, the serializer function for the `last_updated` field illustrated in the example above.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 184); [backlink](#)

Unknown interpreted text role "class".

You can specify any kind of metadata for each field. There is no restriction on the values accepted by `:class:'Field'` objects. For this same reason, there is no reference list of all available metadata keys. Each key defined in `:class:'Field'` objects could be used by a different component, and only those components know about it. You can also define and use any other `:class:'Field'` key in your project too, for your own needs. The main goal of `:class:'Field'` objects is to provide a way to define all field metadata in one place. Typically, those components whose behaviour depends on each field use certain field keys to configure that behaviour. You must refer to their documentation to see which metadata keys are used by each component.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 188); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 188); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 188); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 188); [backlink](#)

Unknown interpreted text role "class".

It's important to note that the `:class:'Field'` objects used to declare the item do not stay assigned as class attributes. Instead, they can be accessed through the `:attr:'Item.fields'` attribute.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 199); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 199); [backlink](#)

Unknown interpreted text role "attr".

The `:class:'Field'` class is just an alias to the built-in `:class:'dict'` class and doesn't provide any extra functionality or attributes. In other words, `:class:'Field'` objects are plain-old Python dicts. A separate class is used to support the `:ref:'item declaration syntax <topics-items-declaring>'` based on class attributes.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 206); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 206); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 206); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 206); [backlink](#)

Unknown interpreted text role "ref".

Note

Field metadata can also be declared for `dataclass` and `attrs` items. Please refer to the documentation for [dataclasses.field](#) and [attr.ib](#) for additional information.

Working with Item objects

Here are some examples of common tasks performed with items, using the `Product` item `:ref:` declared above `<topics-items-declaring>`. You will notice the API is very similar to the `:class:`dict`` API.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 223); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 223); [backlink](#)

Unknown interpreted text role "class".

Creating items

```
>>> product = Product(name='Desktop PC', price=1000)
>>> print(product)
Product(name='Desktop PC', price=1000)
```

Getting field values

```
>>> product['name']
Desktop PC
>>> product.get('name')
Desktop PC
```

```
>>> product['price']
1000
```

```
>>> product['last_updated']
Traceback (most recent call last):
...
KeyError: 'last_updated'
```

```
>>> product.get('last_updated', 'not set')
not set
```

```
>>> product['lala'] # getting unknown field
Traceback (most recent call last):
...
KeyError: 'lala'
```

```
>>> product.get('lala', 'unknown field')
'unknown field'
```

```
>>> 'name' in product # is name field populated?
True
```

```
>>> 'last_updated' in product # is last_updated populated?
False
```

```
>>> 'last_updated' in product.fields # is last_updated a declared field?
True
```

```
>>> 'lala' in product.fields # is lala a declared field?
False
```

Setting field values

```
>>> product['last_updated'] = 'today'
>>> product['last_updated']
today
```

```
>>> product['lala'] = 'test' # setting unknown field
Traceback (most recent call last):
...
KeyError: 'Product does not support field: lala'
```

Accessing all populated values

To access all populated values, just use the typical `class:`dict`` API:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 291); [backlink](#)

Unknown interpreted text role "class".

```
>>> product.keys()
['price', 'name']

>>> product.items()
[('price', 1000), ('name', 'Desktop PC')]
```

Copying items

To copy an item, you must first decide whether you want a shallow copy or a deep copy.

If your item contains `term:`mutable`` values like lists or dictionaries, a shallow copy will keep references to the same mutable values across all different copies.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 308); [backlink](#)

Unknown interpreted text role "term".

For example, if you have an item with a list of tags, and you create a shallow copy of that item, both the original item and the copy have the same list of tags. Adding a tag to the list of one of the items will add the tag to the other item as well.

If that is not the desired behavior, use a deep copy instead.

See `mod:`copy`` for more information.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 319); [backlink](#)

Unknown interpreted text role "mod".

To create a shallow copy of an item, you can either call `meth:`~scrapy.Item.copy`` on an existing item (`product2 = product.copy()`) or instantiate your item class from an existing item (`product2 = Product(product)`).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 321); [backlink](#)

Unknown interpreted text role "meth".

To create a deep copy, call `meth:`~scrapy.Item.deepcopy`` instead (`product2 = product.deepcopy()`).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 326); [backlink](#)

Unknown interpreted text role "meth".

Other common tasks

Creating dicts from items:

```
>>> dict(product) # create a dict from all populated values
{'price': 1000, 'name': 'Desktop PC'}
```

Creating items from dicts:

```
>>> Product({'name': 'Laptop PC', 'price': 1500})
Product(price=1500, name='Laptop PC')

>>> Product({'name': 'Laptop PC', 'lala': 1500}) # warning: unknown field in dict
Traceback (most recent call last):
...
KeyError: 'Product does not support field: lala'
```

Extending Item subclasses

You can extend Items (to add more fields or to change some metadata for some fields) by declaring a subclass of your original Item. For example:

```
class DiscountedProduct(Product):
    discount_percent = scrapy.Field(serializer=str)
    discount_expiration_date = scrapy.Field()
```

You can also extend field metadata by using the previous field metadata and appending more values, or changing existing values, like this:

```
class SpecificProduct(Product):
    name = scrapy.Field(Product.fields['name'], serializer=my_serializer)
```

That adds (or replaces) the `serializer` metadata key for the `name` field, keeping all the previously existing metadata values.

Supporting All Item Types

In code that receives an item, such as methods of `ref: item pipelines <topics-item-pipeline>` or `ref: spider middlewares <topics-spider-middleware>`, it is a good practice to use the `class: ~itemadapter.ItemAdapter` class and the `func: ~itemadapter.is_item` function to write code that works for any `ref: supported item type <item-types>`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 376); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 376); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 376); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 376); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 376); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 383)

Unknown directive type "autoclass".

```
.. autoclass:: itemadapter.ItemAdapter
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 385)

Unknown directive type "autofunction".

```
.. autofunction:: itemadapter.is_item
```

Other classes related to items

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scrapy-master\docs\topics\ (scrapy-master) (docs) (topics) items.rst, line 391)

Unknown directive type "autoclass".

```
.. autoclass:: ItemMeta
```