

2.4. 文档实例

要创建新文档对象，请创建相关文档类的实例，为其字段提供值作为构造函数关键字参数。您可以为文档中的任何字段提供值：

```
>>> page = Page(title="Test Page")
>>> page.title
'Test Page'
```

您还可以使用标准对象属性语法为文档的字段赋值：

```
>>> page.title = "Example Page"
>>> page.title
'Example Page'
```

2.4.1. 保存和删除文件

MongoEngine 跟踪文档的更改以提供高效的保存。要将文档保存到数据库，请调用该 `save()` 方法。如果数据库中不存在该文档，则会创建它。如果它已经存在，那么任何更改都将自动更新。例如：

```
>>> page = Page(title="Test Page")
>>> page.save() # Performs an insert
>>> page.title = "My Page"
>>> page.save() # Performs an atomic set on the title field.
```

📌 笔记

跟踪文档的更改并在整体上执行 `set` 操作。

`list_field.push(0)` — 设置结果列表

`del(list_field)` — 取消设置整个列表

最好使用列表，`Doc.update(push__list_field=0)` 因为这会停止更新整个列表——停止任何竞争条件。

📌 也可以看看

2.4.1.1. 级联保存

如果您的文档包含 `ReferenceField` 或 `GenericReferenceField` 对象，则默认情况下该 `save()` 方法不会保存对这些对象的任何更改。如果您还希望保存所有引用，请注意每次保存都是一个单独的查询，然后将 `cascade=True` 传递给保存方法将级联任何保存。

2.4.1.2. 删除文件

要删除文档，请调用该 `delete()` 方法。请注意，这仅在文档存在于数据库中并且具有有效的 `id`。

2.4.2. 文档 ID

数据库中的每个文档都有一个唯一的 ID。`id` 这可以通过对象的属性来访问 `Document`。通常，`id` 将在对象保存时由数据库服务器自动生成，这意味着您只能 `id` 在文档保存后访问该字段：

```
>>> page = Page(title="Test Page")
>>> page.id
>>> page.save()
>>> page.id
ObjectId('123456789abcdef0000000000')
```

或者，您可以将自己的字段之一定义为文档的“主键”，方法是将其 `primary_key=True` 作为关键字参数提供给字段的构造函数。在幕后，MongoEngine 将使用此字段作为 `id`；事实上实际上 `id` 是您的主键字段的别名，因此您仍然可以根据 `id` 需要使用它来访问主键：

```
>>> class User(Document):
...     email = StringField(primary_key=True)
...     name = StringField()
...
>>> bob = User(email='bob@example.com', name='Bob')
>>> bob.save()
>>> bob.id == bob.email == 'bob@example.com'
True
```

您还可以使用该字段访问文档的“主键” `pk`，它是以下内容的别名 `id`：

```
>>> page = Page(title="Another Test Page")
>>> page.save()
>>> page.id == page.pk
True
```

❗ 笔记

如果您定义自己的主键字段，该字段将隐式变为必需字段，因此 `ValidationError` 如果您不提供它，将抛出一个。