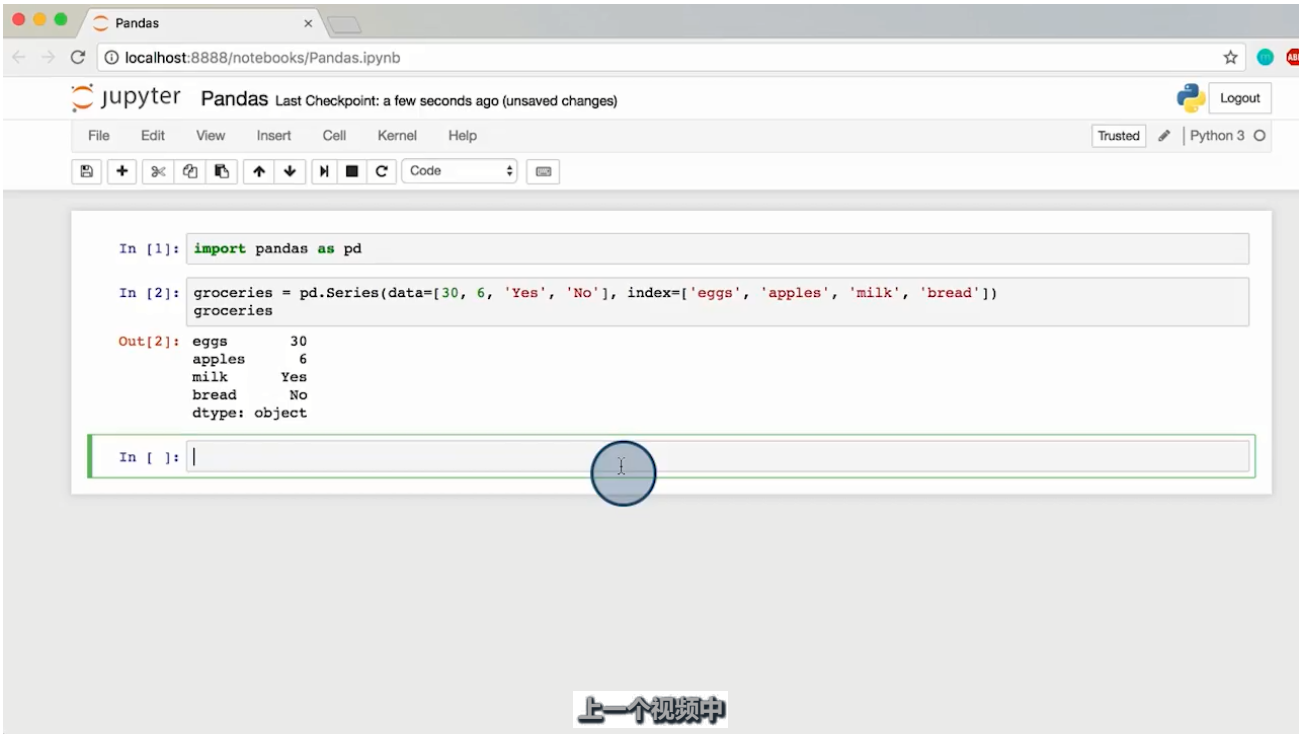
访问和删除 Pandas Series 中的元素



00:00 / 02:34 1x CC

现在我们来了解如何访问或修改 Pandas Series 中的元素。Pandas Series 的一大优势是我们能够以很多不同的方式访问数据。我们可以通过在方括号 [ ] 内添加索引标签或数字索引访问元素，就像访问 NumPy ndarray 中的元素一样。因为我们可以使用数字索引，因此可以使用正整数从 Series 的开头访问数据，或使用负整数从末尾访问。因为我们可以通过多种方式访问元素，为了清晰地表明我们指代的是索引标签还是数字索引，Pandas Series 提供了两个属性 `.loc` 和 `.iloc`，帮助我们清晰地表明指代哪种情况。属性 `.loc` 表示*位置*，用于明确表明我们使用的是标签索引。同样，属性 `.iloc` 表示*整型位置*，用于明确表明我们使用的是数字索引。我们来看一些示例：

```
# We access elements in Groceries using index labels:

# We use a single index label
print('How many eggs do we need to buy:', groceries['eggs'])
print()

# we can access multiple index labels
```

```
# we use loc to access multiple index labels
print('How many eggs and apples do we need to buy:\n', groceries.
print()

# We access elements in Groceries using numerical indices:

# we use multiple numerical indices
print('How many eggs and apples do we need to buy:\n',  groceries
print()

# We use a negative numerical index
print('Do we need bread:\n', groceries[[-1]])
print()

# We use a single numerical index
print('How many eggs do we need to buy:', groceries[0])
print()
# we use iloc to access multiple numerical indices
print('Do we need milk and bread:\n', groceries.iloc[[2, 3]])
```

> *How many eggs do we need to buy: 30*

> *Do we need milk and bread:*
> *milk Yes*
> *bread No*
> *dtype: object*

> *How many eggs and apples do we need to buy:*
> *eggs 30*
> *apples 6*
> *dtype: object*

```
apples    6
dtype: object
```

```
Do we need bread:
bread    No
dtype: object
```

```
How many eggs do we need to buy: 30
```

```
Do we need milk and bread:
milk      Yes
bread     No
dtype: object
```

和 NumPy ndarray 一样，Pandas Series 也是可变的，也就是说，创建好 Pandas Series 后，我们可以更改其中的元素。例如，我们更改下购物清单中的鸡蛋购买数量

```
# We display the original grocery list
print('Original Grocery List:\n', groceries)

# We change the number of eggs to 2
groceries['eggs'] = 2

# We display the changed grocery list
print()
print('Modified Grocery List:\n', groceries)
```

```
Original Grocery List:
eggs       30
apples      6
milk      Yes
bread      No
dtype: object
```

```
apples      6
milk      Yes
bread      No
dtype: object
```

我们还可以使用 `.drop()` 方法删除 Pandas Series 中的条目。`Series.drop(label)` 方法会从给定 `Series` 中删除给定的 `label`。请注意，`Series.drop(label)` 方法不在原地地从 Series 中删除元素，即不会更改被修改的原始 Series。我们来看看代码编写方式

```
# We display the original grocery list
print('Original Grocery List:\n', groceries)

# We remove apples from our grocery list. The drop function remov
print()
print('We remove apples (out of place):\n', groceries.drop('apple

# When we remove elements out of place the original Series remain
# we display our grocery list again
print()
print('Grocery List after removing apples out of place:\n', groce
```

```
Original Grocery List:
eggs      30
apples     6
milk      Yes
bread      No
dtype: object
```

```
We remove apples (out of place):
eggs      30
milk      Yes
```

```
Grocery List after removing apples out of place:
eggs      30
apples     6
milk     Yes
bread     No
dtype: object
```

我们可以通过在 `.drop()` 方法中将关键字 `inplace` 设为 `True`，原地地从 Pandas Series 中删除条目。我们来看一个示例：

```
# We display the original grocery list
print('Original Grocery List:\n', groceries)

# We remove apples from our grocery list in place by setting the
groceries.drop('apples', inplace = True)

# When we remove elements in place the original Series its modifi
# we display our grocery list again
print()
print('Grocery List after removing apples in place:\n', groceries)
```

```
Original Grocery List:
eggs      30
apples     6
milk     Yes
bread     No
dtype: object
```

```
Grocery List after removing apples in place:
eggs      30
milk     Yes
```

Search or ask questions in Knowledge.

Ask peers or mentors for help in Student Hub.

下一项