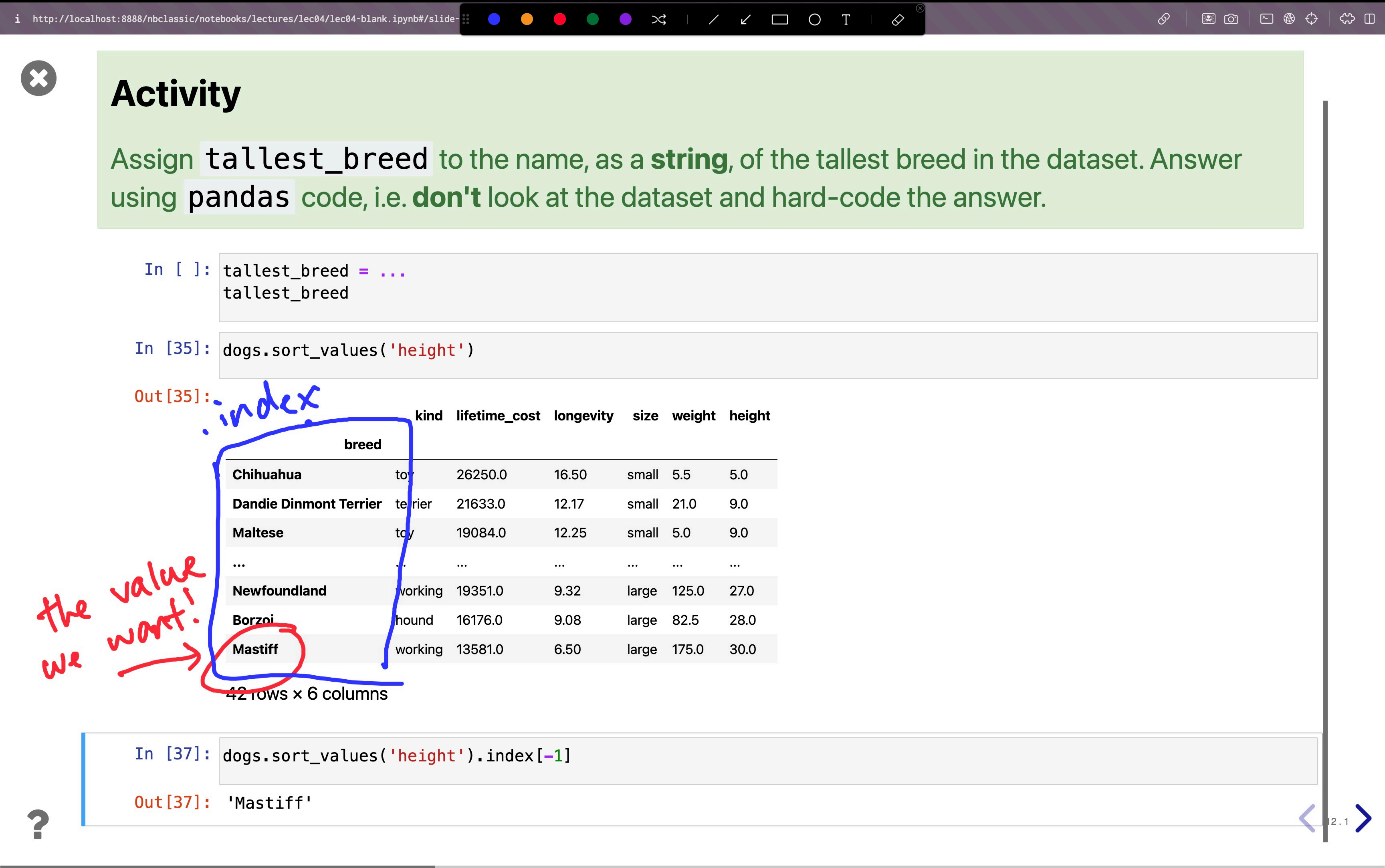


Activity

Assign `tallest_breed` to the name, as a **string**, of the tallest breed in the dataset. Answer using `pandas` code, i.e. **don't** look at the dataset and hard-code the answer.

In [ ]:
```
tallest_breed = ...
tallest_breed
```

In [35]:
```
dogs.sort_values('height')
```

Out[35]: .index

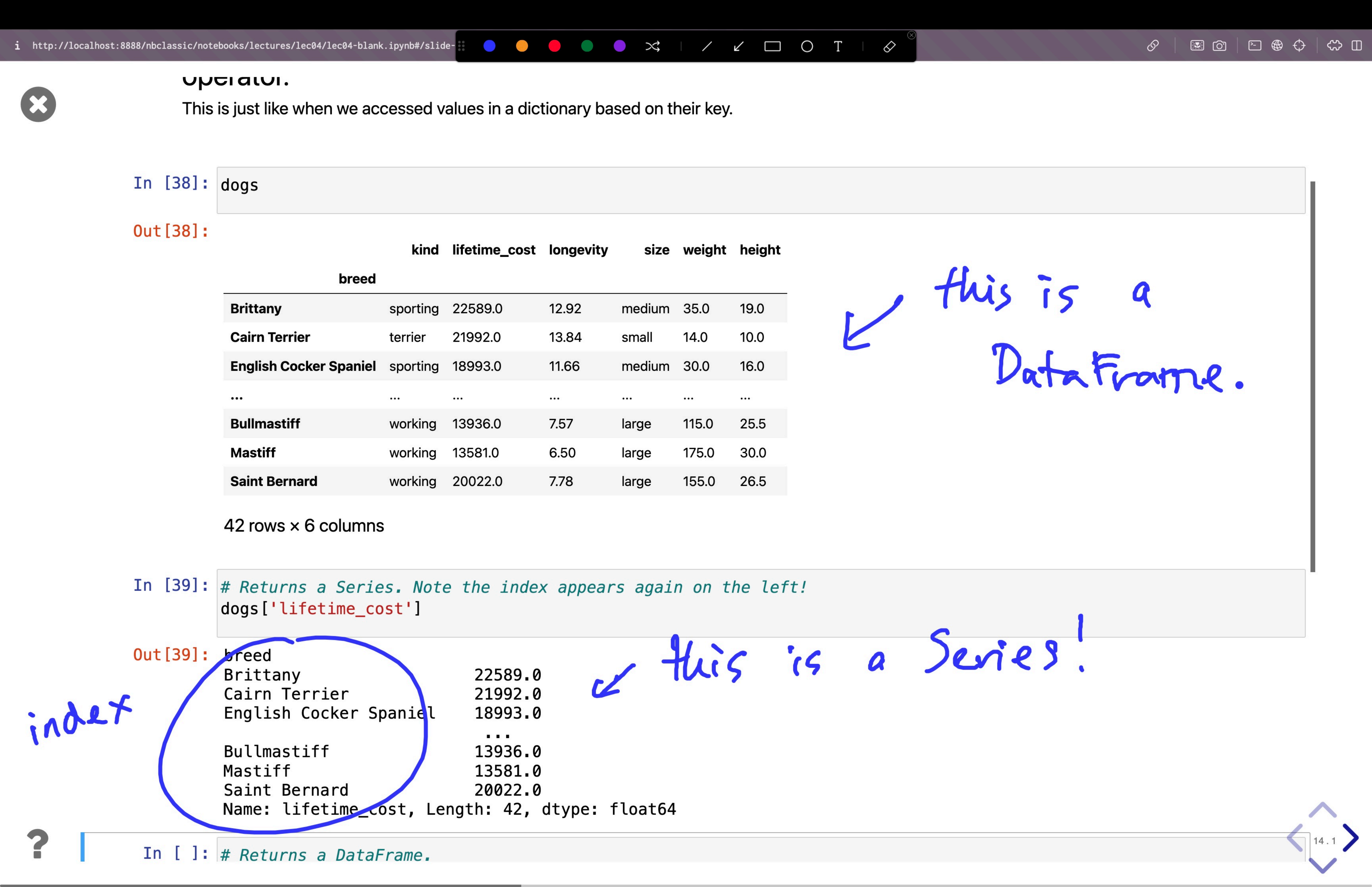
breed	kind	lifetime_cost	longevity	size	weight	height
Chihuahua	toy	26250.0	16.50	small	5.5	5.0
Dandie Dinmont Terrier	terrier	21633.0	12.17	small	21.0	9.0
Maltese	toy	19084.0	12.25	small	5.0	9.0
...	...	...	...	...	...	...
Newfoundland	working	19351.0	9.32	large	125.0	27.0
Borzoi	hound	16176.0	9.08	large	82.5	28.0
Mastiff	working	13581.0	6.50	large	175.0	30.0

42 rows × 6 columns

the value we want!

In [37]:
```
dogs.sort_values('height').index[-1]
```

Out[37]:
```
'Mastiff'
```


http://localhost:8888/nbclassic/notebooks/lectures/lec04/lec04-blank.ipynb#/slide-
14.1

operator.

This is just like when we accessed values in a dictionary based on their key.

In [38]:
```
dogs
```

Out[38]:

	kind	lifetime_cost	longevity	size	weight	height
breed						
Brittany	sporting	22589.0	12.92	medium	35.0	19.0
Cairn Terrier	terrier	21992.0	13.84	small	14.0	10.0
English Cocker Spaniel	sporting	18993.0	11.66	medium	30.0	16.0
...	...	...	...	...	...	...
Bullmastiff	working	13936.0	7.57	large	115.0	25.5
Mastiff	working	13581.0	6.50	large	175.0	30.0
Saint Bernard	working	20022.0	7.78	large	155.0	26.5

42 rows × 6 columns

this is a DataFrame.

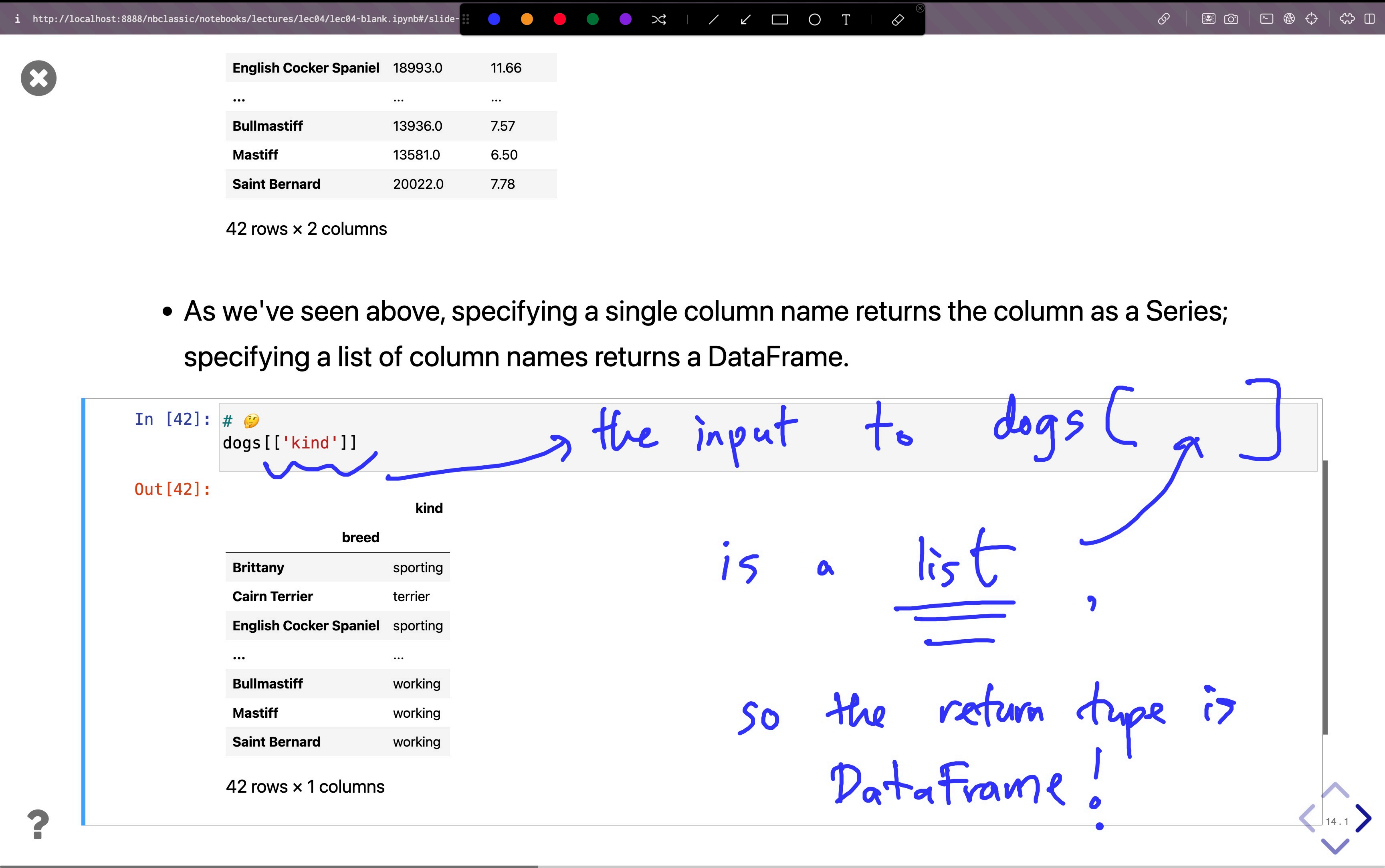
In [39]:
```
# Returns a Series. Note the index appears again on the left!
dogs['lifetime_cost']
```

Out[39]:
```
breed
Brittany                  22589.0
Cairn Terrier             21992.0
English Cocker Spaniel    18993.0
                           ...
Bullmastiff               13936.0
Mastiff                   13581.0
Saint Bernard             20022.0
Name: lifetime_cost, Length: 42, dtype: float64
```

index

this is a Series!

In [ ]:
```
# Returns a DataFrame.
```


http://localhost:8888/nbclassic/notebooks/lectures/lec04/lec04-blank.ipynb#/slide-
14.1

English Cocker Spaniel	18993.0	11.66
...	...	...
Bullmastiff	13936.0	7.57
Mastiff	13581.0	6.50
Saint Bernard	20022.0	7.78

42 rows × 2 columns

- As we've seen above, specifying a single column name returns the column as a Series; specifying a list of column names returns a DataFrame.

In [42]:
```
# 🤔
dogs[['kind']]
```

Out[42]:

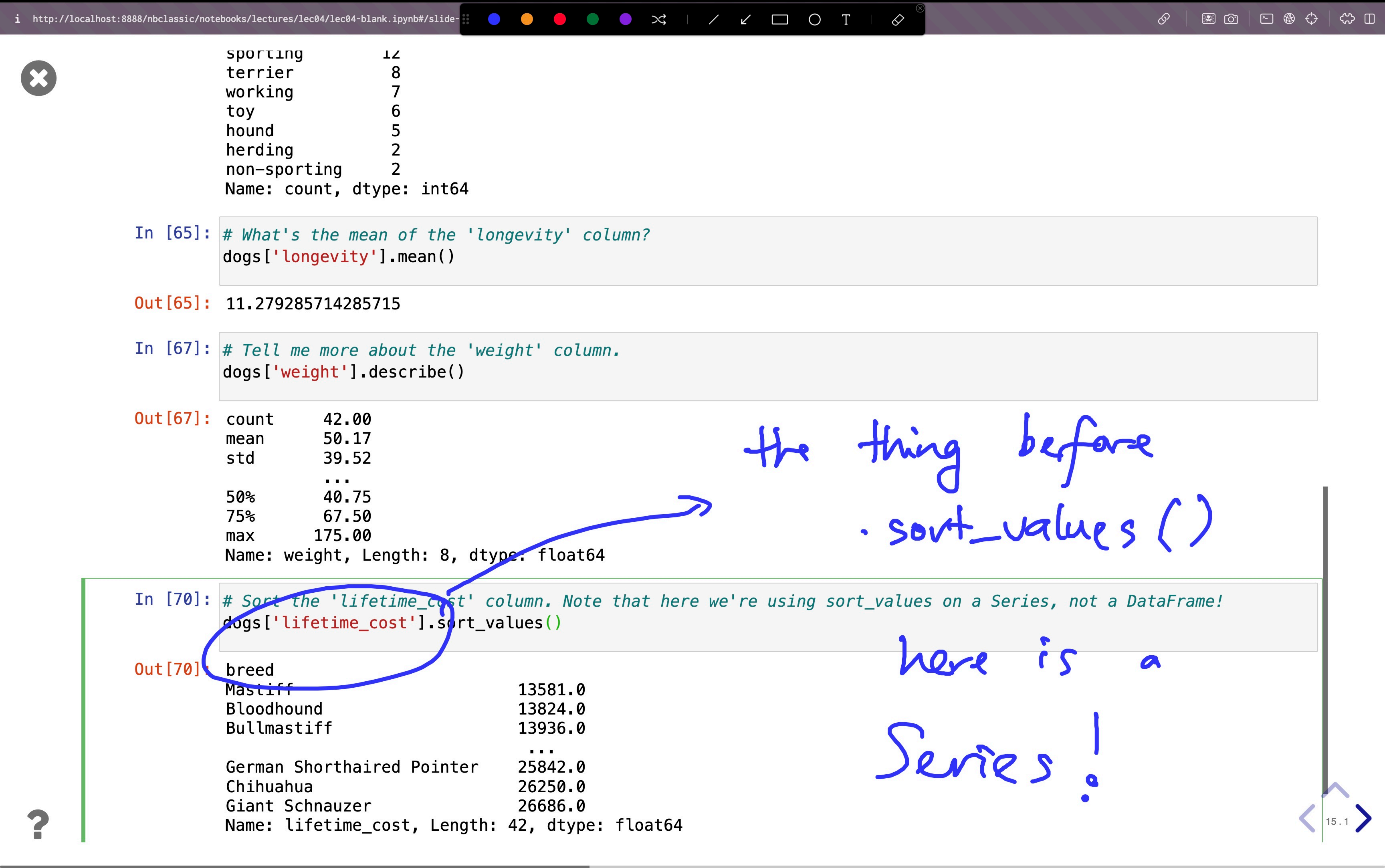
	kind
breed	
Brittany	sporting
Cairn Terrier	terrier
English Cocker Spaniel	sporting
...	...
Bullmastiff	working
Mastiff	working
Saint Bernard	working

42 rows × 1 columns

the input to dogs[ ]

is a list,

so the return type is DataFrame!


http://localhost:8888/nbclassic/notebooks/lectures/lec04/lec04-blank.ipynb#/slide-
15.1

```
sporting         12
terrier           8
working           7
toy               6
hound             5
herding           2
non-sporting      2
Name: count, dtype: int64
```

In [65]:
```
# What's the mean of the 'longevity' column?
dogs['longevity'].mean()
```

Out[65]:
```
11.279285714285715
```

In [67]:
```
# Tell me more about the 'weight' column.
dogs['weight'].describe()
```

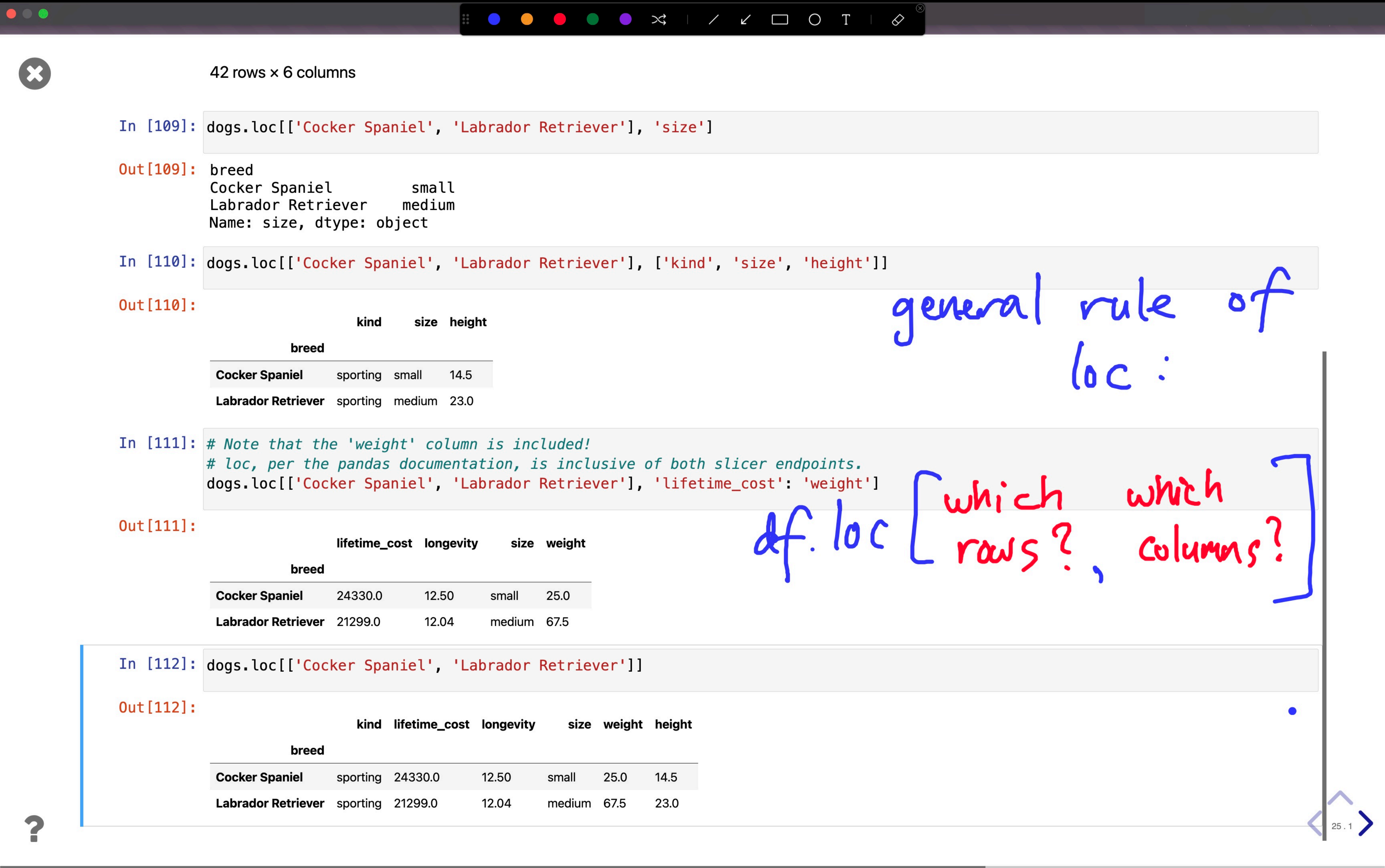
Out[67]:
```
count     42.00
mean      50.17
std       39.52
          ...
50%       40.75
75%       67.50
max      175.00
Name: weight, Length: 8, dtype: float64
```

In [70]:
```
# Sort the 'lifetime_cost' column. Note that here we're using sort_values on a Series, not a DataFrame!
dogs['lifetime_cost'].sort_values()
```

Out[70]:
```
breed
Mastiff                       13581.0
Bloodhound                    13824.0
Bullmastiff                   13936.0
                                ...
German Shorthaired Pointer    25842.0
Chihuahua                     26250.0
Giant Schnauzer               26686.0
Name: lifetime_cost, Length: 42, dtype: float64
```

the thing before .sort_values()

here is a Series!



42 rows × 6 columns

In [109]:
```python
dogs.loc[['Cocker Spaniel', 'Labrador Retriever'], 'size']
```

Out[109]:
```
breed
Cocker Spaniel         small
Labrador Retriever    medium
Name: size, dtype: object
```

In [110]:
```python
dogs.loc[['Cocker Spaniel', 'Labrador Retriever'], ['kind', 'size', 'height']]
```

Out[110]:

	kind	size	height
breed			
Cocker Spaniel	sporting	small	14.5
Labrador Retriever	sporting	medium	23.0

general rule of loc :

df.loc [which rows? , which columns?]

In [111]:
```python
# Note that the 'weight' column is included!
# loc, per the pandas documentation, is inclusive of both slicer endpoints.
dogs.loc[['Cocker Spaniel', 'Labrador Retriever'], 'lifetime_cost': 'weight']
```

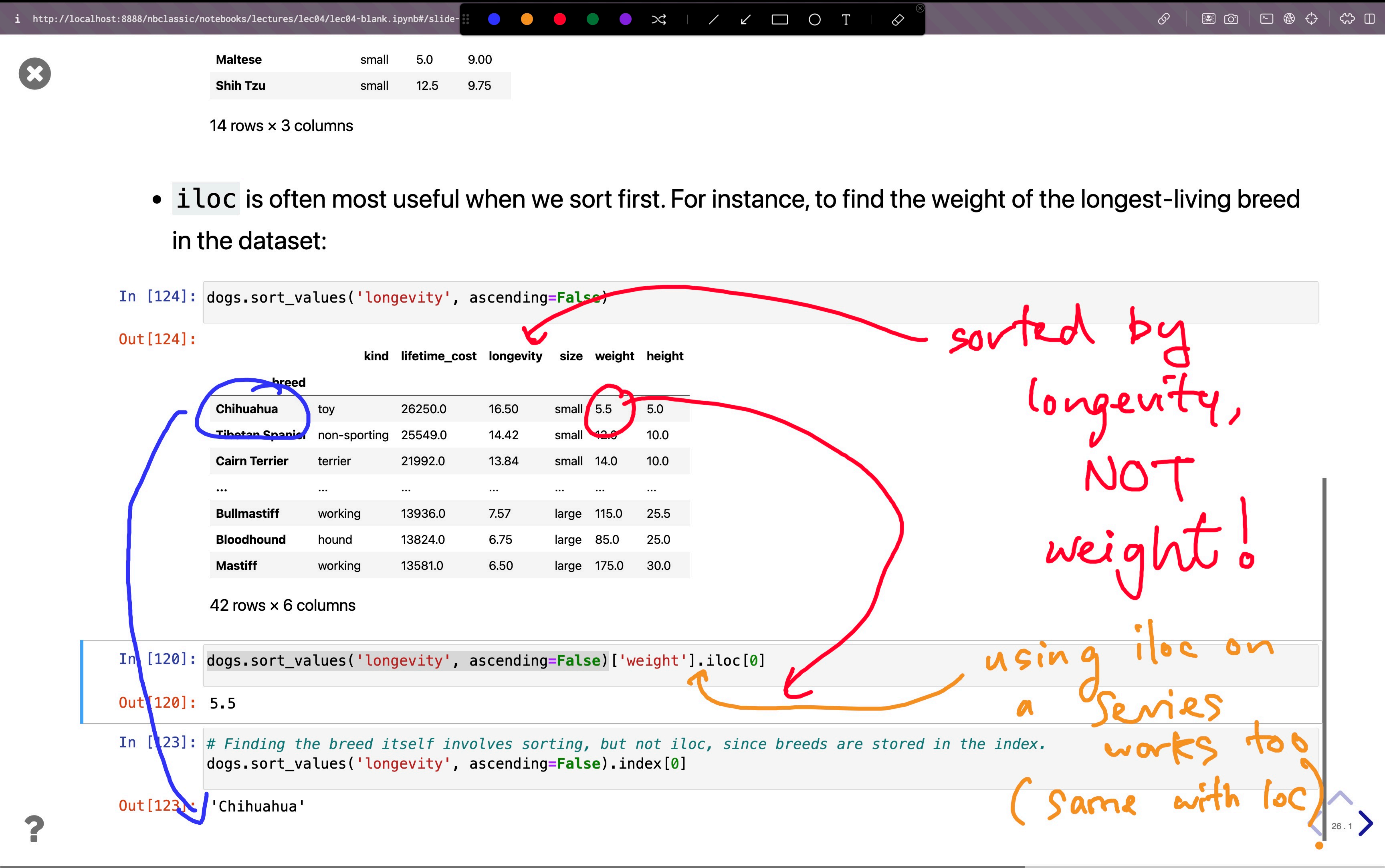
Out[111]:

	lifetime_cost	longevity	size	weight
breed				
Cocker Spaniel	24330.0	12.50	small	25.0
Labrador Retriever	21299.0	12.04	medium	67.5

In [112]:
```python
dogs.loc[['Cocker Spaniel', 'Labrador Retriever']]
```

Out[112]:

	kind	lifetime_cost	longevity	size	weight	height
breed						
Cocker Spaniel	sporting	24330.0	12.50	small	25.0	14.5
Labrador Retriever	sporting	21299.0	12.04	medium	67.5	23.0


http://localhost:8888/nbclassic/notebooks/lectures/lec04/lec04-blank.ipynb#/slide-
26 . 1

Maltese	small	5.0	9.00
Shih Tzu	small	12.5	9.75

14 rows × 3 columns

- `iloc` is often most useful when we sort first. For instance, to find the weight of the longest-living breed in the dataset:

```
In [124]: dogs.sort_values('longevity', ascending=False)
```

Out[124]:

breed	kind	lifetime_cost	longevity	size	weight	height
Chihuahua	toy	26250.0	16.50	small	5.5	5.0
Tibetan Spaniel	non-sporting	25549.0	14.42	small	12.0	10.0
Cairn Terrier	terrier	21992.0	13.84	small	14.0	10.0
...	...	...	...	...	...	...
Bullmastiff	working	13936.0	7.57	large	115.0	25.5
Bloodhound	hound	13824.0	6.75	large	85.0	25.0
Mastiff	working	13581.0	6.50	large	175.0	30.0

42 rows × 6 columns

sorted by longevity, NOT weight!

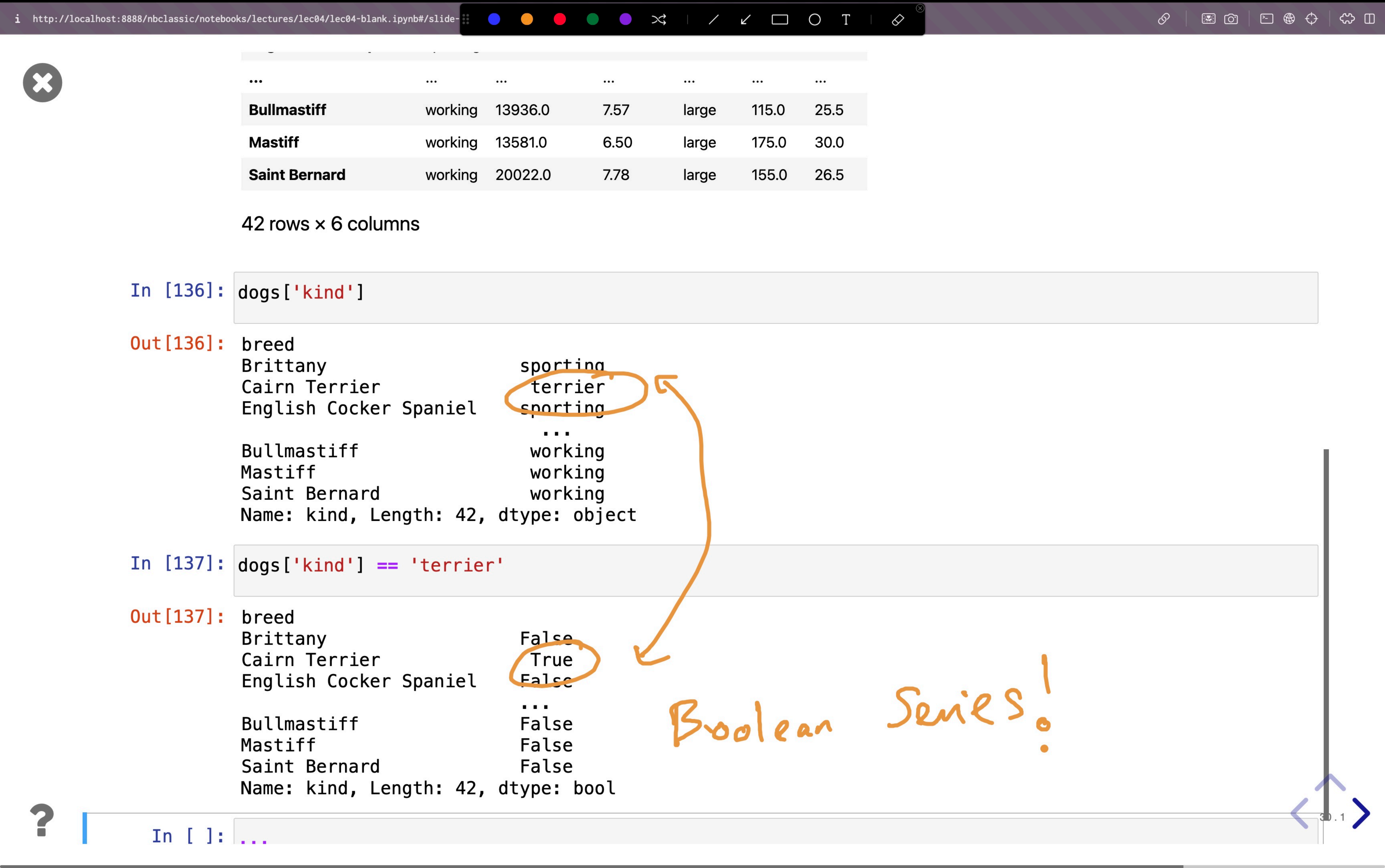
```
In [120]: dogs.sort_values('longevity', ascending=False)['weight'].iloc[0]

Out[120]: 5.5
```

using iloc on a Series works too (same with loc)

```
In [123]: # Finding the breed itself involves sorting, but not iloc, since breeds are stored in the index.
          dogs.sort_values('longevity', ascending=False).index[0]

Out[123]: 'Chihuahua'
```



http://localhost:8888/nbclassic/notebooks/lectures/lec04/lec04-blank.ipynb#/slide-

...	...	...	...	...	...	...
Bullmastiff	working	13936.0	7.57	large	115.0	25.5
Mastiff	working	13581.0	6.50	large	175.0	30.0
Saint Bernard	working	20022.0	7.78	large	155.0	26.5

42 rows × 6 columns

```
In [136]: dogs['kind']
```

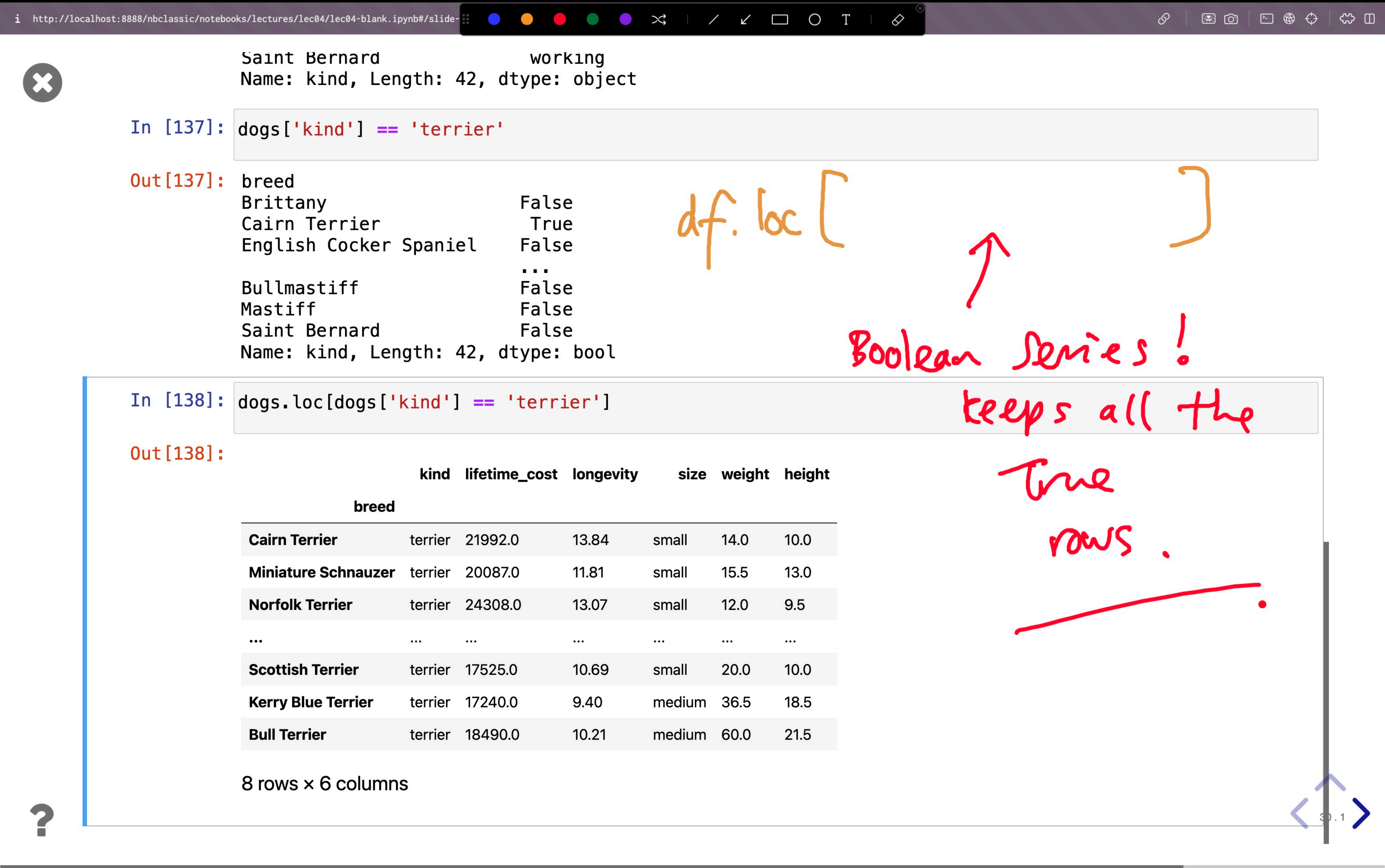
```
Out[136]: breed
          Brittany                       sporting
          Cairn Terrier                   terrier
          English Cocker Spaniel         sporting
                                            ...
          Bullmastiff                     working
          Mastiff                         working
          Saint Bernard                   working
          Name: kind, Length: 42, dtype: object
```

```
In [137]: dogs['kind'] == 'terrier'
```

```
Out[137]: breed
          Brittany                       False
          Cairn Terrier                   True
          English Cocker Spaniel         False
                                          ...
          Bullmastiff                    False
          Mastiff                        False
          Saint Bernard                  False
          Name: kind, Length: 42, dtype: bool
```

Boolean Series!

```
In [ ]: ...
```


http://localhost:8888/nbclassic/notebooks/lectures/lec04/lec04-blank.ipynb#/slide-
30 . 1

```
Saint Bernard                 working
Name: kind, Length: 42, dtype: object
```

In [137]:
```
dogs['kind'] == 'terrier'
```

Out[137]:
```
breed
Brittany                     False
Cairn Terrier                 True
English Cocker Spaniel       False
                             ...
Bullmastiff                  False
Mastiff                      False
Saint Bernard                False
Name: kind, Length: 42, dtype: bool
```

df.loc[ ]

Boolean Series!

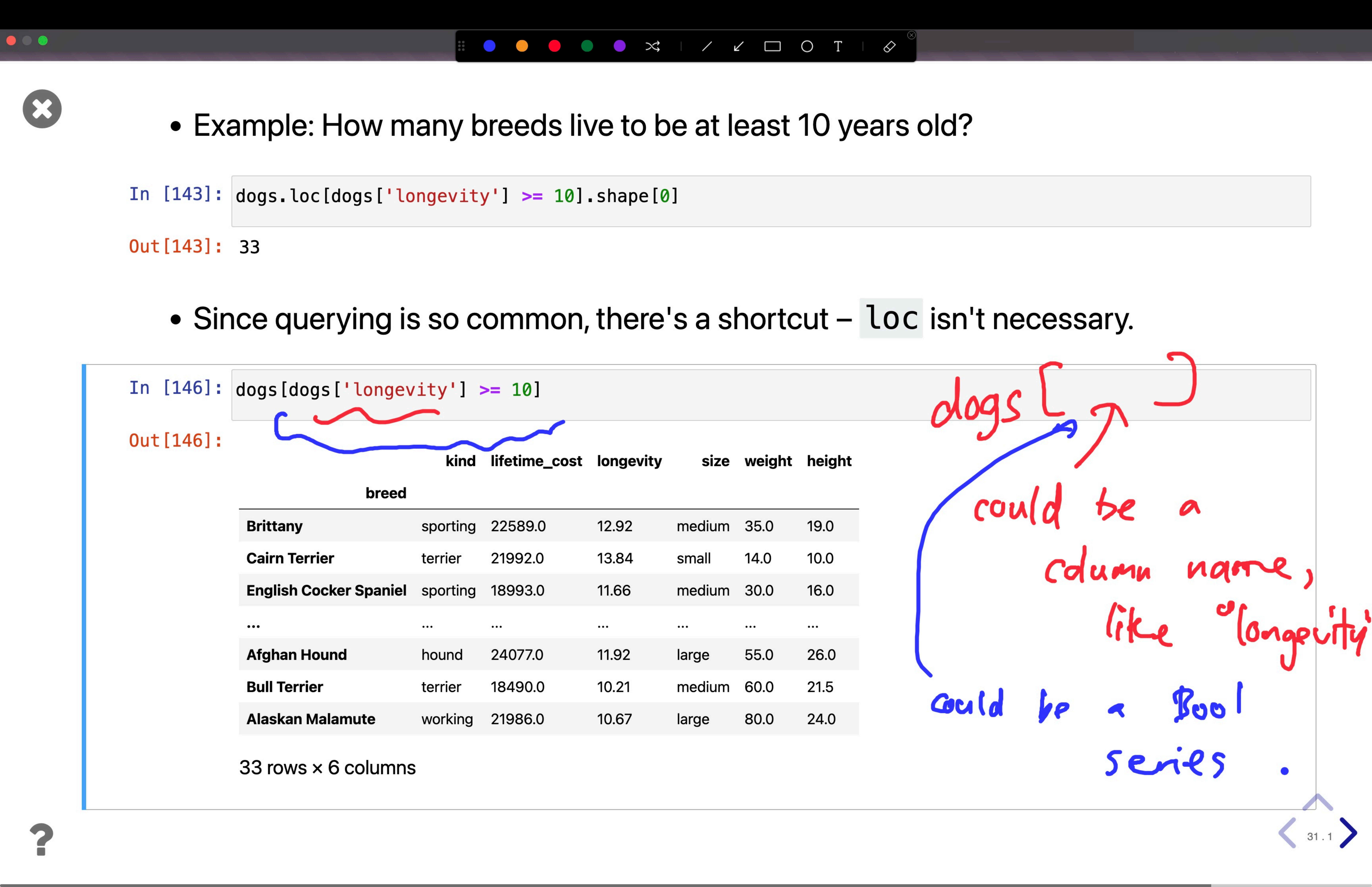
In [138]:
```
dogs.loc[dogs['kind'] == 'terrier']
```

keeps all the True rows.

Out[138]:

	kind	lifetime_cost	longevity	size	weight	height
breed						
Cairn Terrier	terrier	21992.0	13.84	small	14.0	10.0
Miniature Schnauzer	terrier	20087.0	11.81	small	15.5	13.0
Norfolk Terrier	terrier	24308.0	13.07	small	12.0	9.5
...	...	...	...	...	...	...
Scottish Terrier	terrier	17525.0	10.69	small	20.0	10.0
Kerry Blue Terrier	terrier	17240.0	9.40	medium	36.5	18.5
Bull Terrier	terrier	18490.0	10.21	medium	60.0	21.5

8 rows × 6 columns



- Example: How many breeds live to be at least 10 years old?

```
In [143]: dogs.loc[dogs['longevity'] >= 10].shape[0]

Out[143]: 33
```

- Since querying is so common, there's a shortcut – `loc` isn't necessary.

```
In [146]: dogs[dogs['longevity'] >= 10]

Out[146]:
                              kind  lifetime_cost  longevity    size  weight  height
                      breed
Brittany                  sporting        22589.0      12.92  medium    35.0    19.0
Cairn Terrier              terrier        21992.0      13.84   small    14.0    10.0
English Cocker Spaniel    sporting        18993.0      11.66  medium    30.0    16.0
...                            ...            ...        ...     ...     ...     ...
Afghan Hound                 hound        24077.0      11.92   large    55.0    26.0
Bull Terrier               terrier        18490.0      10.21  medium    60.0    21.5
Alaskan Malamute           working        21986.0      10.67   large    80.0    24.0

33 rows × 6 columns
```

dogs[ ]

could be a column name, like "longevity"

could be a Bool series.

- Example: Show me all of the rows for `'sporting'` or `'working'` breeds.

If using multiple conditions, you need parentheses around each condition!
Also, you must use the bitwise operators `&` and `|` instead of the standard `and` and `or` keywords, as we saw in Lecture 3.

bitwise OR

```
In [161]: dogs[(dogs['kind'] == 'sporting') | (dogs['kind'] == 'working')]
```

Out[161]:

the parentheses are needed!

	kind	lifetime_cost	longevity	size	weight	height
breed						
Brittany	sporting	22589.0	12.92	medium	35.0	19.0
English Cocker Spaniel	sporting	18993.0	11.66	medium	30.0	16.0
Cocker Spaniel	sporting	24330.0	12.50	small	25.0	14.5
...	...	...	...	...	...	...
Bullmastiff	working	13936.0	7.57	large	115.0	25.5
Mastiff	working	13581.0	6.50	large	175.0	30.0
Saint Bernard	working	20022.0	7.78	large	155.0	26.5

19 rows × 6 columns

```
In [ ]: # Equivalent to the above!
        ...
```