

# Issue #3 - Gabe, Howard, Mir

## DOCUMENT

This document describes the implementation details for fixing issue **#19708** in pandas library.

### Team Members

The following team members were involved in fixing the issue:

**Implementation**- Howard, Mir, Gabriel

**Testing** - Howard, Mir

**Documentation** - Gabriel

### Description

The `diff()` method is not available in for the **Index** class because it is not a valid operation for the codebase, whereas the `diff()` method exists for **Series** and **DataFrame** objects. Considering that all 3 of the mentioned classes are all data structures used to store and manipulate data, it doesn't make sense that **Series** and **DataFrame** have the implemented method, but not **Index**. Thus, our goal is to simply extend the functionality of the **Index** class by defining a `diff()` method.

Traditionally, the `diff()` method is used to calculate the difference between the values of the rows of a Series or DataFrame object. Here is a small example of how it traditionally works for the **DataFrame** class.

```
>>> df = pd.DataFrame({'a': [1, 2, 3, 4, 5, 6],
...                    'b': [1, 1, 2, 3, 5, 8],
...                    'c': [1, 4, 9, 16, 25, 36]})
>>> df
   a  b  c
0  1  1  1
1  2  1  4
2  3  2  9
3  4  3 16
4  5  5 25
5  6  8 36

>>> df.diff()
   a  b  c
0 NaN NaN NaN
1 1.0 0.0 3.0
2 1.0 1.0 5.0
3 1.0 1.0 7.0
4 1.0 2.0 9.0
5 1.0 3.0 11.0
```

We want to achieve a similar effect on **Index**. Without having to make a function, There is currently a workaround:

```
>>> idx1 = pd.Index([2, 1, 3, 4])
>>> idx2 = pd.Index([3, 4, 5, 6])
>>> idx1.difference(idx2)

Int64Index([1, 2], dtype='int64')
```

Not only is the formatting of the output unintuitive, but having to encapsulate such a simple computation within a method makes the process a lot more convenient when it comes to utilizing **Index**

## Implementation

To address the issue, the following changes were made to the codebase:

Location: `pandas/pandas/core/indexes/base.py`

- First we convert the calling object to a DataFrame. This way we can utilize the `diff()` method that was already defined for DataFrame objects
- The `diff()` function is called on a pandas DataFrame object, which returns a pandas Series object. The `to_series()` function is called on the calling object, which is an instance of the `Index` class.
- The diff method also provides an optional argument, the “period” is how far back we should look when we are calculating the difference with the current element. By default the period is 1 so each difference is calculated by taking the difference between the ith element and the (i-1)-th element. The diff method in the Series method also takes a “period” optional parameter which is why it was included here.
- Finally, the `Index()` function is called on the newly converted Series object. This returns a new instance of the `Index` class, which contains the difference between the values for each row and, by default, the previous row.
- Added test cases in `pandas/pandas/tests/indexes/test_base.py`

Using this method, we reused the `diff()` logic from the **Series** class when creating our `diff()` method. This reduces code duplication and improves maintainability and modularity, as we can now

In our implementation, we constructed a `diff()` method for the **Index** class in the pandas codebase by reusing the logic from the **Series** `diff()` method. This approach involved adapting existing code to work with a new class, which leverages the principles of code reuse and modular design. By utilizing existing, tested code rather than writing new code from scratch, we have ultimately reduced code duplication and testing, as well as improved maintainability.

## Changes Made

The following files were modified as a result of the implementation:

- `pandas/pandas/core/indexes/base.py`

- `pandas/pandas/tests/indexes/test_base.py`

## Files Modified

`pandas/pandas/core/indexes/base.py`

- Implemented the `diff()` function:

```
###implementation of diff()

def diff(self, periods: int = 1) -> Index:
    """
    Calculates the difference of an Index element compared with another element
    in the Index.
    >>> index = Index([1, 2, 3, 4])
    >>> result = index.diff()
    >>> result

    Index([nan, 1.0, 1.0, 1.0], dtype='float64')
    """
    series = self.to_series()
    diff = series.diff(periods=periods)
    return Index(diff)
```

## Testing Suite

`pandas/pandas/tests/indexes/test_base.py`

- Added the following test case to test the behavior of `diff()`:

```
def test_diff(self):
    # GH#19708
    index = Index([1, 2, 3, 4])

    result = index.diff()
    expected = Index([np.nan, 1, 1, 1])
    tm.assert_index_equal(result, expected)

    result = index.diff(periods=-1)
    expected = Index([-1, -1, -1, np.nan])
    tm.assert_index_equal(result, expected)

    result = index.diff(periods=2)
    expected = Index([np.nan, np.nan, 2, 2])
    tm.assert_index_equal(result, expected)
```

This can be run in `pandas/pandas/tests/indexes/test_base.py` with command

```
pytest pandas/pandas/tests/indexes/test_base.py -k test_diff
```