

# Pattern 1: Preprocessing

## Filtering some events

This is one of the most and simple operation. Filtering in Odysseus can be done with the [Select operator](#).

See Examples for Select [here](#).

## Reshaping a stream by removing, renaming, or adding new attributes to events in the stream

This can be done in different ways. If you just want to remove Attributes, you can use the [Project operator](#).

If you want to rename attributes, use the [Rename operator](#).

If you want to add attributes, use the [Map operator](#). With this operator it is also possible to remove and rename attributes. For just removing or renaming you should use one of the above operators, because they provide a much lower footprint.

Examples: [Selection](#), [Projection](#) and [Map](#)

## Splitting and combining attributes in a stream

Splitting and combining of attributes in a single stream can be done with the [Map operator](#). This operator allows many different [mathematical expressions](#) over all attributes inside the input stream.

```

out = MAP({
    expressions = [
        ['bid','renamed_bid'] /// Renaming of attribute
        /// Remark: The output only contains attributes/expressions that are given
        here, so removing an attribute is the by not using it here
    ]
    ,nexmark:person)

/// Simple split name in forename and lastname by looking for the first blanc
out2 = MAP({
    expressions = [
        ['Substring(name,0,indexOf(name," "))','forename'],
        ['Substring(name,indexOf(name,"")+1)','lastname'],
    ]
    },nexmark:person
)

/// Splitting of attribute into list
/// Split string into substrings by " "
presplitted = MAP({
    expressions = [
        ['split(name," ")','splittedName']
    ]
    },nexmark:person
)

/// access first and last element of list
out3 = MAP({
    EXPRESSIONS = [
        ['splittedName[0]','forename'],
        ['elementAt(splittedName,size(splittedName)-1)','lastname']
    ]
    }, presplitted
)

/// access first and last element of list with special function
out4 = MAP({
    EXPRESSIONS = [
        ['first(splittedName)','forename'],
        ['last(splittedName)','lastname']
    ]
    }, presplitted
)

/// List functions
out5 = MAP({
    EXPRESSIONS = [
        ['sublist(splittedName,0,1)','nameAsList'],
        ['sublist(splittedName,1)','nameAsList2'],
        ['rest(splittedName)','allLastElements']
    ]
    }, presplitted
)

```

Remark: In some cases in PQL there are ' and " needed. Inside ' you can use " and vice versa.

## Transforming attributes

Transformation can also be done with the [Map operator](#) by applying mathematical functions (see [MEP: Functions and Operators](#)).