

## Readability

In terms of readability, Python's code is much cleaner and provides a more intuitive and straightforward syntax that even non-programmers can easily understand. It also looks more visually appealing due to the absence of explicit type declarations. For example, when finding the length of an array, python's `len()` function is easy to figure out what it means vs perl's `scalar()` function does not have an obvious relationship to the length of the array. However, in terms of organization, Perl's indentation block structure seems neater as it provides a clever starting and ending point of each block using '{' and '}'. (Please ignore the comments here.)

```
# Function for insertion sort
def insertionSort(arrNum):
    for number in range(1, len(arrNum)):
        current = arrNum[number]
        counter = number - 1 # counter variable stores the
        while counter >= 0 and current < arrNum[counter]:
            # current is smaller than the value before it, and start swapping them until the
            arrNum[counter + 1] = arrNum[counter] # The current value is moved to the position of current)
            counter -= 1
        arrNum[counter + 1] = current # The current variable is placed at the position of counter)
```

```
sub insertionSort{
    my $arrNum = @_; # variable to store the passed array
    my $arrLength = scalar(@$arrNum); #length of the array

    for my $number (1 .. $arrLength){ #loop thorough every number in the array
        my $current = $arrNum->[$number]; #current variable is the value at the current index
        my $counter = $number - 1; #counter variable is the value before the current index
        while ($counter >= 0 && $current < $arrNum->[$counter]){
            # current is smaller than the value before it and start swapping untill they get to the position of current
            $arrNum->[$counter + 1] = $arrNum->[$counter]; #A
            $counter -= 1; #Decrease the counter variable by 1
        } #end while
        $arrNum->[$counter + 1] = $current;
    } #end for
} #end insertionSort
```

## Writability

In terms of writability, I found it easier to write the Python code compared to Perl. The frequent uses of `$`, `->` and `@` slow down the speed of typing. Therefore, I would say that Python provides a more intuitive and concise syntax again in the favor of the programmer due to its simplicity.

## Expressiveness

In terms of expressiveness, I think they are pretty similar as they use the same logic to complete the same implementation in an approximately identical number of lines of code. However, Python seems to be more expressive because its high level of readability and clean syntax make it easier for humans to understand. Perl does not seem to be too expressive compared to python because its syntax are less intuitive and are not obvious to understand its conventions. In addition, its repeated uses of \$ and @ can make the code look messy at a glance.