Step 1: Runtime Analysis

Results for the extraLargeArray:
"doublerAppend" function (.push()): 3.964315 ms
"doublerInsert" function (.unshift()): 823.413875 ms

Results of doublerAppend and doublerInsert applied to all 5 arrays:

	"doublerAppend" function, .push() method	"doublerInsert" function, .unshift() method
tinyArray, length = 10	96 µs	126 µs
smallArray, length = 100	14 µs	24 µs
mediumArray, length = 1,000	50 µs	168 µs
largeArray, length = 10,000	476 µs	7,000 µs
extraLargeArray, length = 100,000	4,000 µs	845,000 µs

Read over the results, and write a paragraph that explains the pattern you see. How does each function "scale"? Which of the two functions scales better? How can you tell?
The runtime differences for both functions from tinyArray to smallArray to mediumArray are surprising. I am not sure why it takes more time to process an array that has a length of 10, than an array that has a length of 100. The function that uses .push() also takes less time to process mediumArray (length of 1,000) than it takes to process the array with a length of 10.

Looking past that, the runtime differences for both functions from mediumArray to largeArray to extraLargeArray are closer to what I would have expected. While there is not a large runtime difference between the .push() method and the .unshift() method while the array length is 1,000 and below, we start to see an exponential increase in the difference in runtime when the array length is 10,000 and 100,000. We can imagine that this difference would be even larger is the array length were in the millions. The "doublerAppend" function that uses .push() scales much better.

For extra credit, do some review/research on why the slower function is so slow and summarize the reasoning for this.
The .push() method adds an element to the end of an array. The .unshift() method shifts all existing elements by one and adds an element at index 0. This means that .unshift() requires reallocating memory and copying data over, while .push() does not. This means that .unshift() is not as time efficient as .push()

2nd attempt at checking runtime:
Append tiny: 190.431 µs

Append small: 28.565 µs
Append medium: 86.427 µs
Append large: 1.000183 ms
Append extra large: 6.66943 ms

Insert tiny: 176.297 µs
Insert small: 37.658 µs
Insert medium: 262.967 µs
Insert large: 12.845556 ms
Insert extra large 850.678705 ms