

# Exercise 27

## FUNCTIONS CALLING FUNCTIONS

Sounds odd, right? Functions calling functions? But it's really not, and this idea of a function calling a function is very important for doing more sophisticated work with JavaScript, so let's dig into it.

The goal is to place in `<p id="top-average">` the top batting average from an object that contains batters and their averages. Look at the pattern for JavaScript objects again:

```
{  
  someKey: someValue,  
  someOtherKey: someOtherValue  
}
```

- Your first task is to create a JavaScript object called `battingAverages` with key:value pairs for the following batters:

Batter	Average
gwynn	0.338
dimaggio	0.325
williams	0.344
robinson	0.311
puckett	0.318

- Next, create a function, `compare`, that accepts two numbers and returns the larger of the two. If `compare` is passed 0.5 and 0.7, it should return 0.7. If the two numbers are the same, it should return the first argument.
- Now, create another function, `topBattingAverages`, that accepts the `battingAverages` object and the `compare` function (in that order).

- ❑ In the function body of `topBattingAverage`, set a variable, `topAverage`, set to `0`. You'll see why we use this shortly.
- ❑ After setting `topAverage`, go through each of the batters in the `battingAverages` object, calling `compare` and passing the `topAverage` first and then one of the items in the `battingAverages` object. From `compare`, you're going to get the greater number of the two. Set the value of `topAverage` to what you get back from `compare`. Go through each of the batters and, as a final thing, return `topAverage`.
- ❑ Now call `topBattingAverage`, passing in the object of batting averages, and the function that does that comparing. Take that value and place it in the `<p id="top-average">` element. That value should be `0.344`.

Remember that once you have created a variable with "let", you'll change the value of the variable without "let".

Example:

`let myCity = "Dallas"`

`myCity = "Las Vegas"`

Is this the hardest exercise so far? I certainly think so — and I'll be surprised (and delighted) if you can get through it without any help. So, I've provided that help in the form of `answer.js`. Don't feel bad if you need to look at it, but do this: once you've seen my solution, don't copy it; instead get the idea of each part of the code, then go back to `index.js` and try to reproduce it based on your new understanding. Need to refer to `answer.js` repeatedly? That's OK. Just don't copy/paste it.