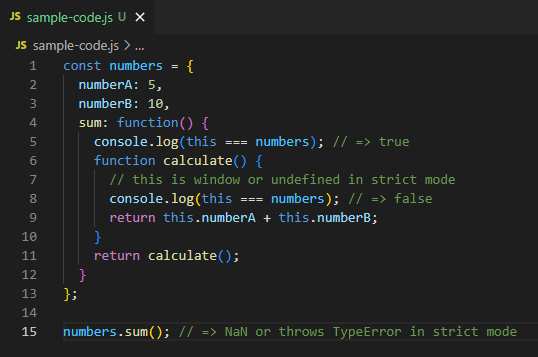
Main concept:

A common trap with the function invocation is thinking that this is the same in an inner function as in the outer function.

The context of the inner function (except arrow function) depends only on its **own invocation type**, but not on the outer function's context.



numbers.sum()

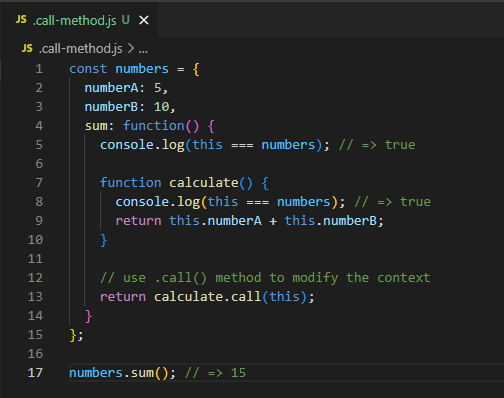
* METHOD invocation
* this === numbers (the object)

calculate()

* FUNCTION invocation NOT METHOD
* function inside sum()
* this === window (non-strict)
* this === undefined (strict)

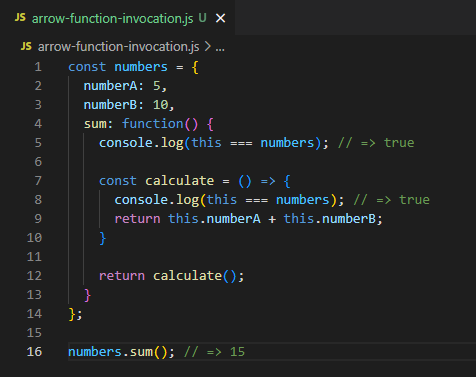
Solutions for this nested functions:

1. .call(this) method



* + calculate became a METHOD invocation
  + this.numberA + this.numberB is same as numbers.numberA + numbers.numberB

1. Arrow Function (preferred)



* + Calling calculate() will invoke an ANONYMOUS method () that is why it is basically a METHOD invocation