JS (Part 7)
Practice Solutions

Ans 1

```js
const arrayAverage = (arr) => {
    let total = 0;
    for (let number of arr) {
        total += number;
    }
    return total / arr.length;
};


let arr = [1, 2, 3, 4, 5, 6];
console.log(arrayAverage(arr));
```

Ans 2

```js
let num = 4;


const isEven = (num) => num % 2 == 0;
```

Ans 3

After a delay of 1 second, undefined is logged to the console.

While the setTimeout() function uses the object.logMessage as a callback, still, it invokes object.logMessage as a regular function, rather than a method.

And during a regular function invocation *this* equals the global object, which is a window in the case of the browser environment.

That's why console.log(this.message) inside logMessage method logs window.message, which is undefined.

Ans 4

'Hello, World!' is logged to the console.

object.getMessage() is a method invocation, that's why *this* inside the method equals object.

There's also a variable declaration const message = 'Hello, Earth!' inside the method. The variable doesn't influence the value of this.message.