# Grasping Advanced Iteration Concepts

**Andrejs Doronins**
Software Developer in Test

# Module Overview

for-in loop

for-await-of loop

Performance considerations

```
// since ES6 - for iterables
for(let e of arr) { ... }

// since ES1 - for object properties
for(let e in arr) { ... }
```

```
let someObj = {
  prop_1: “string”
  prop_2: false
}
```

for-in

(order not guaranteed)

```
// yes
for (let e of iterable) {}
for (let prop of Object.keys(obj)) { }
for (let [key, value] of Object.entries(obj)) { }
for (let prop in obj) { }
// no!
for (let prop in array) { }
```

# Async Looping

```
for await (let e of asyncIterable) { }
```

**Prerequisites:**

- Promises
- Async functions

**Courses:**

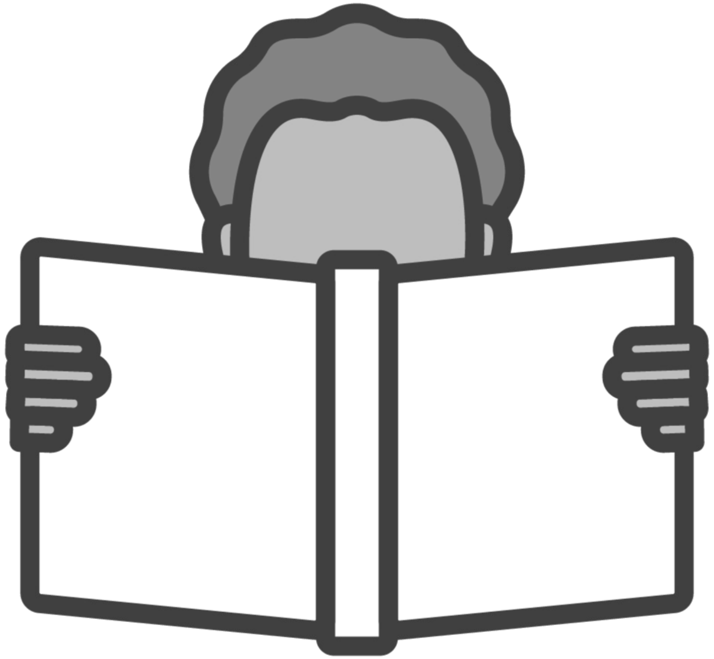
- JavaScript Promises and Async Programming
- Asynchronous Programming in JavaScript

```
for (let p of await Promise.all(promises)){
}
for (let e of items) {
   const res = await somethingAsync(e);
}
for await (let e of asyncIterable) {
    // current iteration depends on the previous iteration
}
```

How to make loops faster?

1. Optimize work done per iteration
2. Reduce the number of iterations

# Advice on Loops



Book: “High Performance JavaScript”

- It’s from 2010!

In 2022: no convincing evidence that caching makes any significant difference

End the search early if appropriate

Don't waste time with premature optimization

# Summary

for-in loop

for-await-of

Performance: don’t optimize blindly