

findLast & findLastIndex for Stage 4

Daniel Rosenwasser and Wenlu Wang
Microsoft Corporation

Status

- GitHub Repo
 - <https://github.com/tc39/proposal-array-find-from-last>
- Spec Text
 - <https://tc39.es/proposal-array-find-from-last/>
- Stage 4 Criteria
 - <https://github.com/tc39/proposal-array-find-from-last/issues/49>
- Implementations
 - <https://github.com/tc39/proposal-array-find-from-last/issues/37>

Updates

- No major changes
 - One typo fix by @styfle

Array.prototype.findLast (*predicate* [, *thisArg*])

When the **findLast** method is called, the following steps are taken:

1. Let *O* be ? ToObject(**this** value).
2. Let *len* be ? LengthOfArrayLike(*O*).
3. If IsCallable(*predicate*) is **false**, throw a **TypeError** exception.
4. Let *k* be *len* - 1.
5. Repeat, while *k* ≥ 0,
 - a. Let *Pk* be ! ToString(ℱ(*k*)).
 - b. Let *kValue* be ? Get(*O*, *Pk*).
 - c. Let *testResult* be ! ToBoolean(? Call(*predicate*, *thisArg*, « *kValue*, ℱ(*k*), *O* »)).
 - d. If *testResult* is **true**, return *kValue*.
 - e. Set *k* to *k* - 1.
6. Return **undefined**.

Array.prototype.findLastIndex (*predicate* [, *thisArg*])

When the **findLastIndex** method is called, the following steps are taken:

1. Let *O* be ? ToObject(**this** value).
2. Let *len* be ? LengthOfArrayLike(*O*).
3. If IsCallable(*predicate*) is **false**, throw a **TypeError** exception.
4. Let *k* be *len* - 1.
5. Repeat, while *k* ≥ 0,
 - a. Let *Pk* be ! ToString(ℱ(*k*)).
 - b. Let *kValue* be ? Get(*O*, *Pk*).
 - c. Let *testResult* be ! ToBoolean(? Call(*predicate*, *thisArg*, « *kValue*, ℱ(*k*), *O* »)).
 - d. If *testResult* is **true**, return ℱ(*k*).
 - e. Set *k* to *k* - 1.
6. Return -1_ℱ.

%TypedArray%.prototype.findLast (*predicate* [, *thisArg*])

When the **findLast** method is called, the following steps are taken:

1. Let *O* be the **this** value.
2. Perform ? ValidateTypedArray(*O*).
3. Let *len* be *O*.[[ArrayLength]].
4. If IsCallable(*predicate*) is **false**, throw a **TypeError** exception.
5. Let *k* be *len* - 1.
6. Repeat, while *k* ≥ 0,
 - a. Let *Pk* be ! ToString(ℱ(*k*)).
 - b. Let *kValue* be ? Get(*O*, *Pk*).
 - c. Let *testResult* be ! ToBoolean(? Call(*predicate*, *thisArg*, « *kValue*, ℱ(*k*), *O* »)).
 - d. If *testResult* is **true**, return *kValue*.
 - e. Set *k* to *k* - 1.
7. Return **undefined**.

%TypedArray%.prototype.findLastIndex (*predicate* [, *thisArg*])

When the **findLastIndex** method is called, the following steps are taken:

1. Let *O* be the **this** value.
2. Perform ? ValidateTypedArray(*O*).
3. Let *len* be *O*.[[ArrayLength]]
4. If IsCallable(*predicate*) is **false**, throw a **TypeError** exception.
5. Let *k* be *len* - 1.
6. Repeat, while *k* ≥ 0,
 - a. Let *Pk* be ! ToString(F(*k*)).
 - b. Let *kValue* be ? Get(*O*, *Pk*).
 - c. Let *testResult* be ! ToBoolean(? Call(*predicate*, *thisArg*, « *kValue*, F(*k*), *O* »)).
 - d. If *testResult* is **true**, return F(*k*).
 - e. Set *k* to *k* - 1.
7. Return -1_F.

Array.prototype [@@unscopables]

The initial value of the @@unscopables data property is an object created by the following steps:

1. Let *unscopableList* be OrdinaryObjectCreate(**null**).
2. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "copyWithin", true).
3. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "entries", true).
4. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "fill", true).
5. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "find", true).
6. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "findIndex", true).
7. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "findLast", true).
8. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "findLastIndex", true).
9. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "flat", true).
10. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "flatMap", true).
11. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "includes", true).
12. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "keys", true).
13. Perform ! CreateDataPropertyOrThrow(*unscopableList*, "values", true).

Stage 4 Qualifications

- Test 262
 - ☒ Merged Manually
- Specification Text PR
 - ☒ Ready
- Multiple Engine Implementations
 - ☒ V8
 - ☒ JSC (behind a flag)
 - ☒ ChakraCore (behind a flag)
 - ☒ SpiderMonkey (close to done!)



Stage 4?