Semantic Patches for Adaptation of JavaScript Programs to Evolving Libraries

Supplementary Material

I. Additional Examples

Example S1 Consider the breaking change affecting the `applyEach` function of the *async* library when it is updated to version 3.0.0. Prior to version 3, `applyEach` took as arguments a collection of functions followed by a varying number of general arguments and at last a callback function. It would then apply each of the functions in the collection with all the general arguments. Upon success or failure the callback would be called. In version 3, `applyEach` is modified to become a curried function that no longer takes the callback argument, but instead returns a function that is called with the callback. Since `applyEach` is variadic, the transformation must use negative indexing to reference all but the last argument (`[0, -2]`) and to reference just the last argument (`$-1`) as demonstrated by the transformation:

```
call <async>.applyEach [2,] ⇝ $callee($args[0, -2])($-1)
```

The detection pattern part of the semantic patch matches calls to the `applyEach` method on the *async* module where `applyEach` is called with at least two arguments.

Example S2 Semantic patches can also be useful for post-processing transformed code, to improve its readability and performance. In version 6 of *rxjs*, all operator functions called on *rxjs* observable objects must be modified to use the `pipe` function. For example, assuming `x` is an observable the following transformation is needed for the calls to `map` and `filter`.

```
31 - x.map(...).filter(...)
32 + import {map, filter} from 'rxjs/operators';
33 + x.pipe(map(...)).pipe(filter(...))
```

That transformation is performed by applying the following semantic patch twice:

```
call <rxjs>**.{map,filter} ⇝ $base.pipe(<rxjs/operators>.$prop($args))
```

The '`**`' part of the detection pattern matches any (potentially empty) sequence of operations between the load of the *rxjs* module and the read of `map` or `filter`. For example, the detection pattern would match both `require('rxjs').map` and `require('rxjs').foo.bar.map`.

For this particular breaking change, a more desirable result is obtained by combining the operators into a single `pipe` call, as shown by the following transformation.

```
34 - x.pipe(map(...)).pipe(filter(...))
35 + x.pipe(map(...), filter(...))
```

The resulting code is more idiomatic and also more efficient. This post-processing transformation can be described using an extra semantic patch that accompanies the one shown above:

```
call <rxjs>**.pipe().pipe ⇝ $base:callee($base:args, $args)
```

It takes two sequential calls to `pipe` and merges the arguments.

In principle, such semantic patches could be applied independently of the semantic patches used for adapting client code to breaking changes in libraries. For example, the transformation in line 35 is also sensible for `pipe` calls that were not inserted by JSFIX. Nevertheless, JSFIX only applies such post-processing patches to clean up code it has inserted, to avoid unnecessary transformations that are unrelated to breaking changes.

Example S3 The promise library *bluebird* contains a function `promisify` that takes as argument a normal callback-based event style function, and then returns a promise wrapper around this function, where the promise is resolved whenever the callback of the wrapped function is called. Prior to the update of *bluebird* to version 3, the promise

would resolve to a single value when the callback is called with a single success value and resolve to an array of values when the callback is called with multiple success values.[1] In version 3, the promise will always resolve to the value corresponding to the first argument of the callback, unless an object with a `multiArgs` field set to `true` is passed to `promisify` in which case the promise always resolves to an array.

To preserve the semantics of code affected by this breaking change, the `multiArgs` field must be set to `true` for exactly those calls to `promisify` where, in version 2 of *bluebird*, the calls would result in a promise that resolves to an array. Because it is impossible to express this constraint in the detection pattern language, JSFIX will ask the user "*Is the first argument a function that calls its callback with more than two arguments?*".[2] Answering "yes" to this question will result in the following semantic patch being applied:

```
call <bluebird>.{promisify,asCallback}
    ⇝ $callee($1, {multiArgs: true})
```

and answering "no" will result in no transformation.

Example S4 The largest accepted pull request was to the *core-js* client *prebid.js* with 97 transformations performed, all for fixing locations affected by the same breaking change. Of the accepted pull requests, the two most complicated ones were for two *rxjs* clients, where one required 12 transformations for 3 different breaking changes, and the other required 16 transformations for 4 different breaking changes. The first of these clients is the *contentful-cli* command line interface npm package for the contentful[3] content manager. The *contentful-cli* package has more than 13 000 weekly downloads. Below are three excerpts of the transformation of *contentful-cli* made by JSFIX.

```
36 - const { Observable, Subject } = require('rxjs/Rx')
37 + const { map, filter } = require('rxjs/operators')
38 + const { merge, Subject } = require('rxjs')

39 - const loggingData$ = scopedEvents$
40 -   .map(({ payload }) => { ... })
41 -   .filter(message => ...)
42 + const loggingData$ = scopedEvents$.pipe(
43 +   map(({ payload }) => { ... }),
44 +   filter(message => ...)
45 + )

46 - const ls$ = Observable.merge(...lss)
47 + const ls$ = merge(...lss)
```

The imports are transformed, to adapt to a breaking change where imports from `'rxjs/Rx'` should be changed to imports from `'rxjs'` instead. Notice also how the import of the `Observable` property has been replaced with an import of the `merge` property, due to a breaking change that results in `merge` no longer being a property on `rxjs.Observable` but instead a function that must be imported directly from `'rxjs'`. As a result of this change, `merge` (line 47) now is used instead of `Observable.merge` (line 46). The last breaking change affecting *contentful-cli* concerns the move of operators, such as `map` and `filter`, from methods on observable objects into independent functions that must be called through `pipe`, as previously demonstrated in Example S2. The two operators, `map` and `filter`, are therefore imported on line 37, and lines 39–41 have been transformed into lines 42–45, resulting in the operators being used inside `pipe`.

II. Semantic Patches

This appendix contains the full list of semantic patches used during the evaluation of JSFIX. For each breaking change in the changelog of the 12 benchmarks, we list all the semantic patches written to handle that breaking change. The manually written questions and the category that each question belongs to (OBJ, EXTRA, or MINOR; questions for CALL are auto-generated) are shown below each semantic patch. The actual notation used by our

[1] The first argument of a standard callback is used to represent errors and is always set to `null` when no error occurs. The remaining arguments are called success values and contain what is typically viewed as the return value of the function to which the callback is supplied.

[2] Deciding such properties fully automatically is beyond the capabilities of any existing static analysis for JavaScript.

[3] https://www.contentful.com/

prototype implementation is JSON (for example, the symbol ⇝ is only used here for readability), but with precisely the information shown below.

lodash 4.0.0

1) Made _#times, _#forEach, _#forIn, _#forOwn, & their right-forms implicitly end chain sequences
   - `call <lodash>()**.{times,forEach,forEachRight,forIn ,forInRight,forOwn,forOwnRight}().value [0, 0] ⇝ $base`

     OBJ: *Is the receiver a lodash implicit chain?*
2) Removed category names from module paths
   - `import lodash/{collection,number,chain,function ,math,array,date,lang,object,string,utility}/* ⇝ <lodash/#2>`
3) Removed _.pluck in favor of _.map with iteratee shorthand
   - `read {<lodash>,<lodash/collection> ,<lodash>()**,<lodash>.chain()**}.pluck ⇝ $base.map`

     OBJ: *Is the function being called from lodash?*
   - `import lodash/collection/pluck ⇝ <lodash/map>`
4) Removed thisArg params from most methods because they were largely unused, complicated implementations, & can be tackled with _.bind, Function#bind, or arrow functions
   - `call {<lodash>,<lodash/utility>}.{callback,iteratee} [2, 2] ⇝ $callee($1.bind($2))`

     OBJ: *Is the function being called from lodash?*
   - `call {<lodash>,<lodash/{collection,number,chain,function,math ,array,date,lang,object,string,utility}>} .{dropRightWhile,dropWhile,findIndex,findLastIndex,remove ,takeRightWhile,takeWhile,unzipWith,zipWith,tap,thru ,countBy,every,all,filter,select,find,findLast,forEach ,each,forEachRight,forEachRight,groupBy,indexBy,map ,collect,partition,reject,some,any,sortBy,cloneDeep ,max,min,sum,findKey,findLastKey,forIn,forInRight ,forOwn,forOwnRight,mapKeys,mapValues,omit,pick,times} [3, 3] 1:function ⇝ $callee($1, $2.bind($3))`

     OBJ: *Is the receiver lodash?*
   - `call {<lodash>(),<lodash>.chain()}** .{dropRightWhile,dropWhile,findIndex,findLastIndex ,remove,takeRightWhile,takeWhile,unzipWith,zipWith ,tap,thru,countBy ,every,all,filter,select,find ,findLast,forEach,each,forEachRight,forEachRight ,groupBy,indexBy,map,collect,partition,reject ,some,any,sortBy,cloneDeep,max,min,sum,findKey ,findLastKey,forIn,forInRight,forOwn,forOwnRight ,mapKeys,mapValues,omit,pick,times} [2, 2] 0: ⇝ $callee($1.bind($2))`

     OBJ: *Is the receiver a lodash chain?*
   - `call <lodash/{collection,number,chain,function,math,array,date ,lang,object,string,utility}/{dropRightWhile ,dropWhile,findIndex,findLastIndex,remove ,takeRightWhile,takeWhile,unzipWith,zipWith,tap ,thru,countBy,every,all,filter,select,find,findLast ,forEach,each,forEachRight,groupBy,indexBy,map ,collect,partition,reject,some,any,sortBy ,cloneDeep,max,min,sum,findKey,findLastKey ,forIn,forInRight,forOwn,forOwnRight,mapKeys ,mapValues,omit,pick,times}> [3, 3] 1:function} ⇝ $callee($1, $2.bind($3))`

     OBJ: *Is the function being called from lodash?*
   - `call <lodash/utility/{callback,iteratee}> [2, 2] ⇝ $callee($1.bind($2))`

     OBJ: *Is the function being called from lodash?*
   - `call {<lodash>,<lodash/{collection,number,chain,function ,math,array,date,lang,object,string,utility}>} .{sortedIndex,sortedLastIndex,uniq,unique,clone,isEqual,eq`

```
    ,isMatch,transform} [4, 4] ⇝ $callee($1, $2, $3.bind($4))
```
OBJ: *Is the function being called from lodash?*

- `call {<lodash>,<lodash/{collection,number,chain,function ,math,array,date,lang,object,string,utility}>} .{reduce,foldl,reduceRight,foldr} [4, 4] ⇝ $callee($1, $2.bind($4), $3)`

  OBJ: *Is the function being called from lodash?*
- `call {<lodash>,<lodash/{collection,number,chain,function,math ,array,date,lang,object,string,utility}>} .{assign,extend,merge} [4,] ⇝ $callee($args[0, -2], $-2.bind($-1))`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Does this call use a customizer function and a thisArg for that customizer?*
- `call <lodash/{collection,number,chain,function,math,array ,date,lang,object,string,utility} /{sortedIndex,sortedLastIndex,uniq,unique,clone,isEqual ,eq,isMatch,transform}> [4, 4] ⇝ $callee($1, $2, $3.bind($4))`

  OBJ: *Is the function being called from lodash?*
- `call <lodash/{collection,number,chain,function,math,array ,date,lang,object,string,utility} /{reduce,foldl,reduceRight,foldr}> [4, 4] ⇝ $callee($1, $2.bind($4), $3)`

  OBJ: *Is the function being called from lodash?*
- `call <lodash/{collection,number,chain,function,math,array ,date,lang,object,string,utility} /{assign,extend,merge}> [4,]} ⇝ $callee($args[0, -2], $-2.bind($-1))`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Does this call use a customizer function?*

5) Split _.max & _.min into _.maxBy & _.minBy
   - `call {<lodash>,<lodash/math>}.{max,min} [2, 3] ⇝ $base.$prop[max => maxBy, min => minBy]($args)`

     OBJ: *Is the function being called from lodash?*
   - `call <lodash/math/{min,max}> [2, 3] ⇝ <lodash/math/#1[min => minBy, max => maxBy]>($args)`

     OBJ: *Is the function being called from lodash?*
6) Removed _.support
   - `read <lodash>.support ⇝ ($base.support || ($base.support = {}))`

     OBJ: *Is the receiver lodash?*
7) Removed _.findWhere in favor of _.find with iteratee shorthand
   - `read {<lodash>,<lodash/collections>}.findWhere ⇝ $base.find`

     OBJ: *Is the receiver lodash?*
   - `import lodash/collections/findWhere ⇝ <lodash/find>`
8) Removed _.where in favor of _.filter with iteratee shorthand
   - `read {<lodash>,<lodash/collections>}.where ⇝ $base.filter`

     OBJ: *Is the receiver lodash?*
   - `import lodash/collections/where ⇝ <lodash/filter>`
9) Renamed _.indexBy to _.keyBy
   - `read {<lodash>,<lodash/collections>}.indexBy ⇝ $base.keyBy`

     OBJ: *Is the receiver lodash?*
   - `import lodash/collections/indexBy ⇝ <lodash/keyBy>`
10) Renamed _.invoke to _.invokeMap
    - `read {<lodash>,<lodash/collections>}.invoke ⇝ $base.invokeMap`

      OBJ: *Is the receiver lodash?*
    - `import lodash/collections/invoke ⇝ <lodash/invokeMap>`

11) Renamed _.modArgs to _.overArgs
    - `read {<lodash>,<lodash/function>}.modArgs ⇝ $base.overArgs`
      OBJ: *Is the receiver lodash?*
    - `import lodash/function/modArgs ⇝ <lodash/overArgs>`
12) Renamed _.padLeft & _.padRight to _.padStart & _.padEnd
    - `read {<lodash>,<lodash/string>}.{padLeft,padRight} ⇝ $base.$prop[padLeft => padStart, padRight => padEnd]`
      OBJ: *Is the receiver lodash?*
    - `import lodash/string/{padLeft,padRight} ⇝ <lodash/#1[padLeft => padStart, padRight => padEnd]>`
13) Renamed _.pairs to _.toPairs
    - `read {<lodash>,<lodash/object>}.pairs ⇝ $base.toPairs`
      OBJ: *Is the receiver lodash?*
    - `import lodash/object/pairs ⇝ <lodash/toPairs>`
14) Renamed _.rest to _.tail
    - `read {<lodash>,<lodash/array>}.rest ⇝ $base.tail`
      OBJ: *Is the receiver lodash?*
    - `import lodash/array/rest ⇝ <lodash/tail>`
15) Renamed _.restParam to _.rest
    - `read {<lodash>,<lodash/function>}.restParam ⇝ $base.rest`
      OBJ: *Is the receiver lodash?*
    - `import lodash/function/restParam ⇝ <lodash/rest>`
16) Renamed _.sortByOrder to _.orderBy
    - `read {<lodash>,<lodash/collections>}.sortByOrder ⇝ $base.orderBy`
      OBJ: *Is the receiver lodash?*
    - `import lodash/collections/sortByOrder ⇝ <lodash/orderBy>`
17) Renamed _.trimLeft & _.trimRight to _.trimStart & _.trimEnd
    - `read {<lodash>,<lodash/string>}.{trimLeft,trimRight} ⇝ $base.$prop[trimLeft => trimStart, trimRight => trimEnd]`
      OBJ: *Is the receiver lodash?*
    - `import lodash/string/{trimLeft,trimRight} ⇝ <lodash/#1[trimLeft => trimStart, trimRight => trimEnd]>`
18) Renamed _.trunc to _.truncate
    - `read {<lodash>,<lodash/string>}.trunc ⇝ $base.truncate`
      OBJ: *Is the receiver lodash?*
    - `import lodash/string/trunc ⇝ <lodash/truncate>`
19) Split _.assign & _.assignIn into _.assignWith & _.assignInWith
    - `call {<lodash>,<lodash/object>}.assign [3,] ⇝ $base.assignWith($args)`
      OBJ: *Is the function being called from lodash?*
      EXTRA: *Does this call to assign use a customizer function?*
    - `call <lodash/object/assign> [3,] ⇝ <lodash/assignWith>($args)`
      OBJ: *Is the function being called from lodash?*
      EXTRA: *Does this call to assign use a customizer function?*
20) Split _.clone & _.cloneDeep into _.cloneWith & _.cloneDeepWith
    - `call {<lodash>,<lodash/lang>}.cloneDeep [2, 3] ⇝ $base.cloneDeepWith($args)`
      OBJ: *Is the function being called from lodash?*
    - `call <lodash/lang/cloneDeep> [2, 3] ⇝ <lodash/cloneDeepWith>($args)`
      OBJ: *Is the function being called from lodash?*
21) Split _.indexOf & _.lastIndexOf into _.sortedIndexOf & _.sortedLastIndexOf

- `call {<lodash>,<lodash/array>}.{indexOf,lastIndexOf} [3, 3] 2:boolean ⇝`
  `($3 ? $base.$prop[indexOf => sortedIndexOf,lastIndexOf => sortedLastIndexOf]($1, $2) : $callee($args))`

  OBJ: *Is the function being called from lodash?*
- `call <lodash/array/{indexOf,lastIndexOf}> [3, 3] 2:boolean ⇝`
  `($3 ? <lodash/array/#1[indexOf => sortedIndexOf,lastIndexOf => sortedLastIndexOf]>($1, $2) : $callee($args))`

  OBJ: *Is the function being called from lodash?*

22) Split _.invert into _.invertBy (see v4.1.0)
    - `call {<lodash>,<lodash/object>}.invert [2, 2] ⇝`
      `($2 ? $base.invertBy($1) : $callee($1))`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/object/invert> [2, 2] ⇝`
      `($2 ? <lodash/invertBy>($1) : $callee($1))`

      OBJ: *Is the function being called from lodash?*
23) Split _.isEqual into _.isEqualWith
    - `call {<lodash>,<lodash/lang>}.isEqual [3, 4] ⇝`
      `$base.isEqualWith($args)`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/lang/isEqual> [3, 4] ⇝`
      `<lodash/lang/isEqualWith>($args)`

      OBJ: *Is the function being called from lodash?*
24) Split _.isMatch into _.isMatchWith
    - `call {<lodash>,<lodash/lang>}.isMatch [3, 4] ⇝`
      `$base.isMatchWith($args)`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/lang/isMatch> [3, 4] ⇝`
      `<lodash/lang/isMatchWith>($args)`

      OBJ: *Is the function being called from lodash?*
25) Split _.merge into _.mergeWith
    - `call {<lodash>,<lodash/object>}.merge [3,] ⇝`
      `$base.mergeWith($args)`

      OBJ: *Is the function being called from lodash?*

      EXTRA: *Does this use of <lodash>.merge use a customizer function?*
    - `call <lodash/object/merge> [3,] ⇝`
      `<lodash/object/mergeWith>($args)`

      OBJ: *Is the function being called from lodash?*

      EXTRA: *Does this use of <lodash>.merge use a customizer function?*
26) Split _.omit & _.pick into _.omitBy & _.pickBy
    - `call {<lodash>,<lodash/object>}.{omit,pick} [2, 2] 1:function ⇝`
      `$base.$prop[omit => omitBy, pick => pickBy]($args)`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/object/{omit,pick}> [2, 2] 1:function ⇝`
      `<lodash/#1[omit => omitBy, pick => pickBy]>($args)`

      OBJ: *Is the function being called from lodash?*
27) Split _.sample into _.sampleSize
    - `call {<lodash>,<lodash/collections>}.sample [2, 2] ⇝`
      `$base.sampleSize($args)`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/collections/sample> [2, 2] ⇝`
      `<lodash/sampleSize>($args)`

      OBJ: *Is the function being called from lodash?*
28) Split _.sortedIndex into _.sortedIndexBy

- `call {<lodash>,<lodash/array>}.sortedIndex [3, 4] ⇝ $base.sortedIndexBy($args)`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/array/sortedIndex> [3, 4] ⇝ <lodash/sortedIndexBy>($args)`
  OBJ: *Is the function being called from lodash?*

29) Split _.sortedLastIndex into _.sortedLastIndexBy
- `call {<lodash>,<lodash/array>}.sortedLastIndex [3, 4] ⇝ $base.sortedLastIndexBy($args)`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/array/sortedLastIndex> [3, 4] ⇝ <lodash/sortedLastIndexBy>($args)`
  OBJ: *Is the function being called from lodash?*

30) Split _.sum into _.sumBy
- `call {<lodash>,<lodash/math>}.sum [2, 3] ⇝ $base.sumBy($args)`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/math/sum> [2, 3] ⇝ <lodash/sumBy>($args)`
  OBJ: *Is the function being called from lodash?*

31) Split _.uniq into _.sortedUniq, _.sortedUniqBy, & _.uniqBy
- `call {<lodash>,<lodash/array>}.uniq [2, 2] 1:boolean ⇝ ($2 ? $base.sortedUniq : $callee)($1)`
  OBJ: *Is the function being called from lodash?*
- `call {<lodash>,<lodash/array>}.uniq [2, 3] 1:function ⇝ $base.uniqBy($args)`
  OBJ: *Is the function being called from lodash?*
- `call {<lodash>,<lodash/array>}.uniq [3, 4] ⇝ ($2 ? $base.sortedUniqBy : $base.uniqBy)($1, $args[2,])`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/array/uniq> [2, 2] 1:boolean ⇝ ($2 ? <lodash/sortedUniq> : $callee)($1)`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/array/uniq> [2, 3] 1:function ⇝ <lodash/uniqBy>($args)`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/array/uniq> [3, 4] ⇝ ($2 ? <lodash/sortedUniqBy> : <lodash/uniqBy>)($1, $args[2,])`
  OBJ: *Is the function being called from lodash?*

32) Split _.zipObject into _.fromPairs
- `call {<lodash>,<lodash/array>}.zipObject [1, 1] ⇝ $base.fromPairs($args)`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/array/zipObject> [1, 1] ⇝ <lodash/fromPairs>($args)`
  OBJ: *Is the function being called from lodash?*

33) Absorbed _.sortByAll into _.sortBy
- `call {<lodash>,<lodash/collections>}.sortByAll ⇝ $base.sortBy($args)`
  OBJ: *Is the function being called from lodash?*
- `call <lodash/collections/sortByAll> ⇝ <lodash/sortBy>($args)`
  OBJ: *Is the function being called from lodash?*

34) Changed the category of _.at to “Object”
- `read <lodash/collections>.at ⇝ <lodash/object>.at`
  OBJ: *Is the receiver lodash?*
- `import lodash/collections/at ⇝ <lodash/at>`

35) Changed the category of _.bindAll to ?Util?

- `read <lodash/function>.bindAll ⇝ <lodash/util>.bindAll`

  OBJ: *Is the receiver lodash?*
- `import lodash/function/bindAll ⇝ <lodash/bindAll>`

36) Changed _.matchesProperty shorthand to an array of [path, srcValue]

- ```
  call {<lodash>,<lodash/{collection,number,chain,function,math
                          ,array,date,lang,object,string,utility}>}
         .{dropRightWhile,dropWhile,findIndex,findLastIndex,remove
          ,takeRightWhile,takeWhile,countBy,every,all
          ,filter,select,find,detect,groupBy,indexBy
          ,map,collect,partition,reject,some,any,sortBy
          ,max,min,findKey,findLastKey,mapValues,uniq,unique}
        [3, 3] 1:string ⇝ $callee($1, [$2, $3])
  ```

  OBJ: *Is the function being called from lodash?*
- ```
  call <lodash/{collection,number,chain,function,math,array
                ,date,lang,object,string,utility}
       /{dropRightWhile,dropWhile,findIndex,findLastIndex
        ,remove,takeRightWhile,takeWhile,countBy,every
        ,all,filter,select,find,detect,groupBy,indexBy
        ,map,collect,partition,reject,some,any,sortBy
        ,max,min,findKey,findLastKey,mapValues,uniq,unique}>
     [3, 3] 1:string ⇝ $callee($1, [$2, $3])
  ```

  OBJ: *Is the function being called from lodash?*
- ```
  call {<lodash>,<lodash/array>}.{sortedIndex,uniq,unique}
       [4, 4] 2:string ⇝ $callee($1, $2, [$3, $4])
  ```

  OBJ: *Is the function being called from lodash?*
- ```
  call <lodash/array/{sortedIndex,uniq,unique}>
       [4, 4] 2:string ⇝ $callee($1, $2, [$3, $4])
  ```

  OBJ: *Is the function being called from lodash?*

37) Enabled _.merge to assign undefined if the destination property doesn?t exist

- ```
  call {<lodash>,<lodash/object>}.merge ⇝
  ((() => {
    const origKeys = Object.keys($1);
    const res = $callee($args);
    Object.keys(res)
      .filter(k => res[k] === undefined && !origKeys.includes(k))
      .forEach(k => delete res[k])
    return res;
  })())
  ```

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Does any of the source properties have the value undefined and the property does not exist at the destination*

  MINOR: *Ask before patching.*
- ```
  call <lodash/object/merge> ⇝
  ((() => {
    const origKeys = Object.keys($1);
    const res = $callee($args);
    Object.keys(res)
      .filter(k => res[k] === undefined && !origKeys.includes(k))
      .forEach(k => delete res[k])
    return res;
  })())
  ```

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Does any of the source properties have the value undefined and the property does not exist at the destination*

  MINOR: *Ask before patching.*

38) Made “By” methods like _.groupBy & _.sortBy provide a single param to iteratees

- ```
  call {<lodash>,<lodash/collections>}
       .{sortBy,countBy,groupBy,indexBy}
       [2, 3] 1:{function2,function3} ⇝
  ((() => { function* getIndex() {
      const keys = Object.keys($1);
  ```
  ```

```
      for (var k of keys) {
        yield k;
      }}
    return $callee($1, val => ($2).call(this, val, getIndex()));})())
```

OBJ: *Is the function being called from lodash?*

- `call <lodash/collections/{sortBy,countBy,groupBy,indexBy}> [2, 3] 1:{function2,function3} ⇝`

```
(() => { function* getIndex() {
     const keys = Object.keys($1);
     for (var k of keys) {
       yield k;
     }}
   return $callee($1, val => ($2).call(this, val, getIndex()));})())
```

OBJ: *Is the function being called from lodash?*

39) Made _.add, _.max, _.min, & _.sum no longer coerce values to numbers
- `call {{<lodash>,<lodash/math>}.{max,min,sum},<lodash/math/{max,min,sum}>} [2, 3] ⇝ $callee(<lodash>.map($1, v => +v), $args[1,])`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Is the the first argument NOT a collection of numbers?*
- `call {{<lodash>,<lodash/math>}.{max,min,sum},<lodash/math/{max,min,sum}>} [1, 1] ⇝ $callee(<lodash>.map($1, v => +v))`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Is the the first argument NOT a collection of numbers?*
- `call {{<lodash>,<lodash/math>}.add,<lodash/math/{add}>} ⇝ $callee(+$1, +$2)`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Is the the first argument NOT a collection of numbers?*

40) Made _.capitalize uppercase the first character & lowercase the rest (see _.upperFirst)
- `call {<lodash>,<lodash/string>}.capitalize [0, 1] ⇝ $base.upperFirst($args)`

  OBJ: *Is the function being called from lodash?*
- `call <lodash/string/capitalize> [0, 1] ⇝ <lodash/upperFirst>($args)`

  OBJ: *Is the function being called from lodash?*

41) Made _.eq its own method instead of an alias for _.isEqual
- `call {<lodash>,<lodash/lang>}.eq ⇝ $base.isEqual`

  OBJ: *Is the function being called from lodash?*
- `call <lodash/lang/eq> ⇝ <lodash/isEqual>($args)`

  OBJ: *Is the function being called from lodash?*

42) Made _.functions return only own method names
- `call {<lodash>,<lodash/object>}.{methods,functions} ⇝ $base.functionsIn($args)`

  OBJ: *Is the function being called from lodash?*
- `call <lodash/object/functions> ⇝ <lodash/functionsIn>($args)`

  OBJ: *Is the function being called from lodash?*

43) Made _.max & _.min return undefined when passed an empty array
- `call {<lodash>,<lodash/math>}.max ⇝ ($callee($args) || -Infinity)`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Can the function be called with an empty array?*
- `call {<lodash>,<lodash/math>}.min ⇝ ($callee($args) || Infinity)`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Can the function be called with an empty array?*
- `call <lodash/math/max> ⇝ ($callee($args) || -Infinity)`

  OBJ: *Is the function being called from lodash?*

EXTRA: *Can the function be called with an empty array?*

- `call <lodash/math/min> ⇝ ($callee($args) || Infinity)`

  OBJ: *Is the function being called from lodash?*

  EXTRA: *Can the function be called with an empty array?*

44) Made _.words chainable by default
    - `call <lodash>()**.words ⇝ $callee($args).value()`

      OBJ: *Is the function being called from lodash?*

45) Removed isDeep params from _.clone & _.flatten
    - `call {<lodash>,<lodash/lang>}.clone [3, 4] 1:boolean ⇝ ($2 ? $base.cloneDeepWith($1, $args[2,]) : $base.cloneWith($1, $args[2,]))`

      OBJ: *Is the function being called from lodash?*
    - `call {<lodash>,<lodash/lang>}.clone [2, 3] 1:function ⇝ $base.cloneWith($args)`

      OBJ: *Is the function being called from lodash?*
    - `call {<lodash>,<lodash/lang>}.clone [2, 2] 1:boolean ⇝ ($2 ? $base.cloneDeepWith($1) : $base.cloneWith($1))`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/lang/clone> [3, 4] 1:boolean ⇝ ($2 ? <lodash/cloneDeepWith>($1, $args[2,]) : <lodash/cloneWith>($1, $args[2,]))`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/lang/clone> [2, 3] 1:function ⇝ <lodash/lang/cloneWith>($args)`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/lang/clone> [2, 2] 1:boolean ⇝ ($2 ? <lodash/cloneDeep>($1) : $callee($1))`

      OBJ: *Is the function being called from lodash?*

46) Removed support for binding all methods by default from _.bindAll
    - `call {<lodash>,<lodash/function>}.bindAll [1, 1] ⇝ $callee($1, <lodash>.keysIn($1) .filter(k => typeof $1[k] === 'function'))`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/function/bindAll> [1, 1] ⇝ $callee($1, <lodash>.keysIn($1) .filter(k => typeof $1[k] === 'function'))`

      OBJ: *Is the function being called from lodash?*

47) Dropped boolean options param support in _.debounce, _.mixin, & _.throttle
    - `call {<lodash>,<lodash/{function,utility}>} .{debounce,throttle} [3, 3] 2:boolean ⇝ $callee($1, $2, {leading: $3})`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/{function,utility}/{debounce,throttle}> [3, 3] 2:boolean ⇝ $callee($1, $2, {leading: $3})`

      OBJ: *Is the function being called from lodash?*
    - `call {<lodash>,<lodash/{function,utility}>}.mixin [3, 3] 2:boolean ⇝ $callee($1, $2, {chain: $3})`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/{function,utility}/mixin> [3, 3] 2:boolean ⇝ $callee($1, $2, {chain: $3})`

      OBJ: *Is the function being called from lodash?*

48) Dropped support for boolean orders param in _.orderBy
    - `call {<lodash>,<lodash/collections>}.sortOrderBy [3, 3] ⇝ $base.orderBy($1, $2, $3.map(e => e ? 'asc' : 'desc'))`

      OBJ: *Is the function being called from lodash?*
    - `call <lodash/collections/sortOrderBy> [3, 3] ⇝`

```
<lodash/orderBy>($1, $2, $3.map(e => e ? 'asc' : 'desc'))
```
OBJ: *Is the function being called from lodash?*

49) Made _.escapeRegExp align to the defunct ES7 proposal
- ```
  call {<lodash>,<lodash/string>}.escapeRegExp [1, 1] ⇝
  $callee($args).replace('/', '\/')
  ```
  OBJ: *Is the function being called from lodash?*

  EXTRA: *Do the argument contain '/'?*

  MINOR: *Ask before patching.*
- ```
  call <lodash/string/escapeRegExp> [1, 1] ⇝
  $callee($args).replace('/', '\/')
  ```
  OBJ: *Is the function being called from lodash?*

  EXTRA: *Do the argument contain '/'?*

  MINOR: *Ask before patching.*

50) Made _.max, _.min, & _.sum support arrays only
- ```
  call <lodash>.{max,min,sum} 0:{string,object} ⇝
  $callee($base.values($1), $args[1,])
  ```
  OBJ: *Is the function being called from lodash?*
- ```
  call <lodash/math>.{max,min,sum} 0:{string,object} ⇝
  $callee(<lodash/values>($1), $args[1,])
  ```
  OBJ: *Is the function being called from lodash?*
- ```
  call <lodash/math/{max,min,sum}> 0:{string,object} ⇝
  $callee(<lodash/values>($1), $args[1,])
  ```
  OBJ: *Is the function being called from lodash?*

51) 17 aliases removed
- ```
  read {<lodash>,<lodash/{collection,number,chain,function,math
                         ,array,date,lang,object,string,utility}>
                  ,<lodash>()**,<lodash>.chain()**}
       .{all,any,backflow,callback,collect,compose,contains,detect
        ,foldl,foldr,include,inject,methods,object,#run,select,unique}
  ⇝ $base.$prop[all => every, any => some, backflow => flowRight
      , callback => iteratee, collect => map, compose => flowRight
      , contains => includes, detect => find, foldl => reduce
      , foldr => reduceRight, include => includes, inject => reduce
      , methods => functions, object => fromPairs, run => value
      , select => filter, unique => uniq]
  ```
  OBJ: *Is the receiver lodash?*
- ```
  import lodash/{collection,number,chain,function,math,array
                ,date,lang,object,string,utility}
         /{all,any,backflow,callback,collect,compose,contains
          ,detect,foldl,foldr,include,inject,methods,object
          ,#run,select,unique} ⇝
  <lodash/#2[all => every, any => some, backflow => flowRight
      , callback => iteratee, collect => map, compose => flowRight
      , contains => includes, detect => find, foldl => reduce
      , foldr => reduceRight, include => includes, inject => reduce
      , methods => functions, object => fromPairs, select => filter
      , unique => uniq]>
  ```

async 3.0.0

1) In queue, priorityQueue, cargo and cargoQueue, the "event"-style methods, like q.drain and q.saturated are now methods that register a callback, rather than properties you assign a callback to. They are now of the form q.drain(callback). If you do not pass a callback a Promise will be returned for the next occurrence of the event, making them await-able, e.g. await q.drain(). (#1586, #1641)
- ```
  write {<async>.{queue,priorityQueue,cargo,cargoQueue}
        ,<async/{queue,priorityQueue,cargo,cargoQueue}>}()?
        .{drain,saturated,unsaturated,empty,error} ⇝
  $base.$prop($value)
  ```
  OBJ: *Is the receiver to the write a async queue or cargo?*

2) during and doDuring have been removed, and instead whilst, doWhilst, until and doUntil now have asynchronous test functions. (#850, #1557)

- `read <async>.{during,doDuring} ⇝ $base.$prop[during => whilst, doDuring => doWhilst]`

  OBJ: *Is the receiver async?*
- `import async/{during,doDuring} ⇝ <async/#1[during => whilst, doDuring => doWhilst]>`
- `call {<async>.{doWhilst,doUntil},<async/{doWhilst,doUntil}>} ⇝`
```
$callee(
  $1,
  function() {
    const cb = arguments[arguments.length - 1];
    const args = Array.prototype.slice
        .call(arguments, 0, arguments.length - 1);
    const f = $2;
    try {
      cb(null, f.apply(this, args));
    } catch (e) {
      cb(e, null);
    }
  },
  $3)
```
  OBJ: *Is the function being called from async?*
- `call {<async>.{whilst,until},<async/{whilst,until}>} ⇝`
```
$callee(
  function() {
    const cb = arguments[arguments.length - 1];
    const args = Array.prototype.slice
        .call(arguments, 0, arguments.length - 1);
    const f = $1;
    try {
      cb(null, f.apply(this, args));
    } catch (e) {
      cb(e, null);
    }
  },
  $2,
  $3)
```
  OBJ: *Is the function being called from async?*

3) memoize no longer memoizes errors (#1465, #1466)
   - A semantic patch could not be expressed for the pattern:
     `call {<async>.memoize,<async/memoize>}`
4) applyEach/applyEachSeries have a simpler interface, to make them more easily type-able. It always returns a function that takes in a single callback argument. If that callback is omitted, a promise is returned, making it awaitable. (#1228, #1640)
   - `call {<async>.{applyEach,applyEachSeries},<async/{applyEach,applyEachSeries}>} [2,] ⇝ $callee($args[0, -1])($-1)`

     OBJ: *Is the function being called from async?*

express 4.0.0

1) remove connect and connect's patches except for charset handling
   - `read <express>?**.headerSent ⇝ $base.headersSent`

     OBJ: *Is the receiver an express response object?*
   - `call <express>?**.on [2, 2] 0:"header" 1:function ⇝ ((() => {<on-headers>($base, $2); return $base;})())`

     OBJ: *Is the receiver an express response object?*
   - `call <express>?**.setHeader 0:"Set-Cookie" 1:string ⇝ $base.cookie(...($1.split('=')))`

     OBJ: *Is the receiver an express response object?*
   - `call <express>?**.setHeader 0:"Set-Cookie" 1:array ⇝ $1.forEach(str => $base.cookie(...(str.split('='))))`

     OBJ: *Is the receiver an express response object?*
2) remove bodyParser middelware

- `call <express>.bodyParser ⇝ <body-parser>($args)`
  OBJ: *Is the receiver express?*
- `call <express>.json ⇝ <body-parser>.json($args)`
  OBJ: *Is the receiver express?*
- `call <express>.urlencoded ⇝ <body-parser>.urlencoded($args)`
  OBJ: *Is the receiver express?*

3) remove all bundled middleware except static
   - ```
     call <express>.{compress,timeout,cookieParser,cookieSession
                    ,csrf,errorHandler,session,methodOverride
                    ,logger,responseTime,favicon,directory,vhost} ⇝
     $prop[compress => <compression>, timeout => <connect-timeout>
         , cookieParser => <cookie-parser>
         , cookieSession => <cookie-session>, csrf => <csurf>
         , errorHandler => <errorhandler>, session => <express-session>
         , methodOverride => <method-override>, logger => <morgan>
         , responseTime => <response-time>, favicon => serve-favicon
         , directory => <serve-index>, vhost => <vhost>]($args)
     ```
     OBJ: *Is the receiver express?*
4) remove express.createServer() - it has been deprecated for a long time. Use express()
   - `read <express>.createServer ⇝ $base`
     OBJ: *Is the receiver express?*
5) remove app.configure - use logic in your own app code
   - `call <express>?**.configure [1, 1] ⇝ $1()`
     OBJ: *Is the receiver an express app?*
   - ```
     call <express>?**.configure [2, 2] ⇝
     if ($1 === $base.get('env')) {
       $2()
     }
     ```
     OBJ: *Is the receiver an express app?*
6) remove app.router - is removed
   - A semantic patch could not be expressed for the pattern:
     `read <express>?**.router`
7) remove req.auth - use basic-auth instead
   - `read <express>.basicAuth ⇝ <basic-auth-connect>`
     OBJ: *Is the receiver express?*
8) remove req.accepted* - use req.accepts*() instead
   - ```
     read <express>?**.{acceptedCharsets,acceptedLanguages} ⇝
     <accepts>(req).$prop[acceptedCharsets => charsets
         , acceptedLanguages => languages]
     ```
     OBJ: *Is the receiver an express request?*
9) res.location - relative URL resolution is removed
   - ```
     call <express>?.location [1, 1] 0:string ⇝
     $callee(((res, url) => {
       var app = res.app
         , req = res.req
         , path;

       // "back" is an alias for the referrer
       if ('back' == url) url = req.get('Referrer') || '/';

       // relative
       if (!~url.indexOf('://') && 0 != url.indexOf('//')) {
         // relative to path
         if ('.' == url[0]) {
           path = <parseurl>.original(req).pathname;
           path = path + ('/' == path[path.length - 1] ? '' : '/');
           url = <url>.resolve(path, url);
           // relative to mount-point
         } else if ('/' != url[0]) {
           path = app.path();
           url = path + '/' + url;
     ```

```
    }
  }
  return url;
})($base, $1))
```

OBJ: *Is the receiver an express response?*

EXTRA: *Is the argument a relative URL?*

10) remove res.charset - include the charset in the content type when using res.set()
    - `write <express>?**.charset ⇝ $base.set('Content-Type', ($base.get('Content-Type') ? $base.get('Content-Type') + '; ' : '') + 'charset=' + $value)`

    OBJ: *Is the receiver an express response?*
11) change app.route -> app.mountpath when mounting an express app in another express app
    - `read <express>()?**.route ⇝ $base.mountPath`

    OBJ: *Is the receiver an express app?*
12) change req.accepts* -> req.accepts*s - i.e. req.acceptsEncoding -> req.acceptsEncodings
    - `read <express>?**.{acceptsEncoding,acceptsCharset,acceptsLanguage} ⇝ $base.$prop[acceptsEncoding => acceptsEncodings , acceptsCharset => acceptsCharsets , acceptsLanguage => acceptsLanguages]`

    OBJ: *Is the receiver an express request?*
13) change req.params is now an object instead of an array
    - `read <express>?**.params ⇝ Object.assign([], $base.params)`

    OBJ: *Is the receiver an express request?*

    MINOR: *Ask before patching.*
14) change res.locals is no longer a function. It is a plain js object. Treat it as such.
    - `call <express>?**.locals ⇝ Object.assign($callee, $1)`

    OBJ: *Is the receiver an express response?*
15) refactor req.accepts* with accepts
    - `call <express>?**.{acceptsCharset,acceptsLanguage} ⇝ (!!$base.$prop[acceptsCharset => acceptsCharsets , acceptsLanguage => acceptsLanguages]($args))`

    OBJ: *Is the receiver an express request?*
16) refactor req.is with type-is
    - `call <express>?**.is ⇝ (!!$base.is($args))`

    OBJ: *Is the receiver an express request?*

    MINOR: *Ask before patching.*
17) Removed app.routes
    - `read <express>?**.routes ⇝ ['post', 'get', 'delete', 'put'] .map(rt => [rt, $base._router.stack.map(r => r.route) .filter(r => r && r.methods[rt])]) .reduce((acc, elem) => {acc[elem[0]] = elem[1]; return acc}, {})`

    OBJ: *Is the receiver an express app?*
18) express.mime is removed. Use express.static.mime
    - `read <express>.mime ⇝ $base.static.mime`

    OBJ: *Is the receiver express?*

chalk 2.0.0

1) Removed chalk.hasColor(). Use the has-ansi package directly instead. 04cae22
    - `read <chalk>.hasColor ⇝ <has-ansi@2.0.0>`

    OBJ: *Is the receiver chalk?*
2) Removed chalk.stripColor(). Use the strip-ansi package directly instead. 04cae22
    - `read <chalk>.stripColor ⇝ <strip-ansi@3.0.0>`

    OBJ: *Is the receiver chalk?*

3) Removed chalk.styles. Use the ansi-styles package directly instead. 8702496
   - `read <chalk>.styles ⇝ <ansi-styles@2.2.1>`
     OBJ: *Is the receiver chalk?*

bluebird 3.0.0

1) If a second argument is passed to Promise.promisify, Promise.promisifyAll or Promise.asCallback, it should be wrapped inside an object, i.e., if the second argument is arg2, then the patch should make the second argument into the object {context: arg2}.
   - `call <bluebird>.{promisify,promisifyAll,asCallback} [2, 2] ⇝ $callee($1, {context: $2})`
     OBJ: *Is the function being called from bluebird?*
2) Previously if the callback to be promisified was with multiple values the resolved value was an array off all the arguments and just the value in the case of the callback only being called with one argument. Now by default the resolved value is always a single even though the callback is called multiple times. There is however an option that can be enabled, such that the resolved value always is an array (multiArgs). The patch could then use this option and in the case of a single element array, resolve with the single value.
   - `call <bluebird>.{promisify,asCallback} [1, 1] ⇝ $callee($1, {multiArgs: true})`
     OBJ: *Is the function being called from bluebird?*
     EXTRA: *Is the first argument a function that calls its callback with more than two arguments?*
   - `call <bluebird>.{promisify,asCallback} [2, 2] ⇝ $callee($1, Object.assign({}, $2, {multiArgs: true}))`
     OBJ: *Is the function being called from bluebird?*
     EXTRA: *Is the first argument a function that calls its callback with more than two arguments?*
   - `call <bluebird>.promisifyAll [1, 1] ⇝ $callee($1, {multiArgs: true})`
     OBJ: *Is the function being called from bluebird?*
     EXTRA: *Is the first argument an object with functions that calls their callback with more than two arguments?*
   - `call <bluebird>.promisifyAll [2, 2] ⇝ $callee($1, Object.assign({}, $2, {multiArgs: true}))`
     OBJ: *Is the function being called from bluebird?*
     EXTRA: *Is the first argument an object with functions that calls their callback with more than two arguments?*
3) Cancellation redesign.
   - No general patch exists for the pattern:
     `read <bluebird>?**.{cancellable,uncancellable}`
4) Promise progression has been completely removed.
   - A semantic patch could not be expressed for the pattern:
     `read <bluebird>?**.{progressed,progress}`
   - A semantic patch could not be expressed for the pattern:
     `call <bluebird>?**.{then,done,fork} [3, 3]`
5) .spread's second argument has been removed.
   - No general patch exists for the pattern:
     `call <bluebird>?**.spread [2, 2]`
6) .done causes an irrecoverable fatal error in Node.js environments now. See #471 for rationale.
   - No general patch exists for the pattern:
     `call <bluebird>?**.done`
7) Errors created with Promise.reject or reject callback of new Promise are no longer marked as OperationalErrors.
   - `call <bluebird>?**.error [1, 1] 0:function ⇝ $base.catch($args)`
     OBJ: *Is the receiver a Bluebird promise?*
     EXTRA: *Is this errorhandler used to catch errors created with Promise.reject or reject callback of new Promise*
8) Submodules moved from folder 'main' to folder 'release'

- `import bluebird/js/main/* ⇝ <bluebird/js/release/#1>`

uuid 3.0.0

1) remove .parse and .unparse
   - `call <uuid>.{parse,unparse} ⇝ <uuid-parse>.$prop($args)`
     OBJ: *Is the receiver uuid?*

rxjs 6.0.0

1) webSocket: webSocket creator function now exported from rxjs/websocket as websocket.
   - `import rxjs/**/*observable/dom/{WebSocket,WebSocketSubject}{,*,/**/*} ⇝ <rxjs/websocket>.#3`
2) utils: Many internal use utilities like isArray are now hidden under rxjs/internal, they are implementation details and should not be used.
   - No general patch exists for the pattern:
     ```
     import rxjs/**/*util/{isArray,applyMixins,ArgumentOutOfRangeError
                          ,EmptyError,errorObject,identity,Immediate
                          ,isDate,isNumeric,isObject,isFunction,isPromise
                          ,isScheduler,noop,not,ObjectUnsubscribedError
                          ,pipe,root,SubscribeToResult,TimeoutError
                          ,toSubscriber,tryCatch,UnsubscriptionError}
           {,*,/**/*}
     ```
3) testing observables: HotObservable and ColdObservable, and other testing support types are no longer exported directly.
   - No general patch exists for the pattern:
     `import rxjs/**/*testing/{HotObservable,ColdObservable}{,*,/**/*}`
4) creation functions: All create functions such as of, from, combineLatest and fromEvent should now be imported from rxjs/create.
   - ```
     read <rxjs{,/**/*}>.Observable
          .{from,fromPromise,of,combineLatest,fromEvent
           ,timer,interval,merge,bindCallback,bindNodeCallback
           ,empty,if,throw,defer,concat,combineAll,concatAll
           ,endWith,forkJoin,mergeAll,pairwise,race,startWith
           ,withLatestFrom,zip,ajax,generate,range} ⇝
     <rxjs>.$prop[fromPromise => from, throw => throwError]
     ```
     OBJ: *Is the receiver rxjs.Observable?*
   - ```
     import rxjs/observable/{from,fromPromise,of,combineLatest
                            ,fromEvent,timer,interval,merge
                            ,bindCallback,bindNodeCallback,empty
                            ,if,throw,defer,concat,combineAll
                            ,concatAll,endWith,forkJoin,mergeAll
                            ,pairwise,race,startWith,withLatestFrom
                            ,zip,ajax,generate,range} ⇝
     <rxjs/create/#1>
     ```
5) symbols: Symbols are no longer exported directly from modules such as rxjs/symbol/observable please use Symbol.observable and Symbol.iterator (polyfills may be required)
   - `import rxjs/symbol/* ⇝ ∅`
   - `read <rxjs/symbol/*>.$$observable ⇝ Symbol.$prop[$$observable => observable]`
     OBJ: *Is the receiver from rxjs/symbol?*
6) deep imports: Can no longer deep import top-level types such as rxjs/Observable, rxjs/Subject, rxjs/ReplaySubject, et al. All imports should be done directly from rxjs, for example: import
   Observable, Subject
   from 'rxjs';
   - ```
     import rxjs/{Observable,Subject,ReplaySubject
                 ,BehaviorSubject,Subscription,Subscriber} ⇝ <rxjs>
     ```
7) schedulers: Scheduler instances have changed names to be suffixed with Scheduler, (e.g. asap -> asapScheduler)
   - `read <rxjs>.Scheduler.{async,virtualTime,animationFrame,asap,queue} ⇝ <rxjs>.$propScheduler`
     ```

OBJ: *Is the receiver rxjs.Scheduler?*

8) operators: Pipeable operators must now be imported from rxjs like so: import { map, filter, switchMap } from 'rxjs/operators';. No deep imports.
   - `import rxjs/operator/{audit,combineAll,debounceTime,do,findIndex,map,mergeScan,publishBehavior,retry,single,switchMap,timeInterval,windowTime,auditTime,combineLatest,defaultIfEmpty,elementAt,find,mapTo,min,publish,retryWhen,skip,switchMapTo,timeout,windowToggle,bufferCount,concatAll,delay,every,first,materialize,multicast,publishLast,sample,skipLast,take,timeoutWith,windowWhen,buffer,concat,delayWhen,exhaust,groupBy,max,observeOn,publishReplay,sampleTime,skipUntil,takeLast,timestamp,withLatestFrom,bufferTime,concatMap,dematerialize,exhaustMap,ignoreElements,mergeAll,onErrorResumeNext,race,scan,skipWhile,takeUntil,toArray,zipAll,bufferToggle,concatMapTo,distinct,expand,isEmpty,merge,pairwise,reduce,sequenceEqual,startWith,takeWhile,toPromise,zip,bufferWhen,count,distinctUntilChanged,filter,last,mergeMap,partition,repeat,share,subscribeOn,throttle,windowCount,catch,debounce,distinctUntilKeyChanged,finally,let,mergeMapTo,pluck,repeatWhen,shareReplay,switch,throttleTime,window} ⇝ <rxjs/operators>.#1`

9) ajax: Ajax observable should be imported from rxjs/ajax.
   - `import rxjs/**/*observable/dom/ajax ⇝ <rxjs/ajax>`

10) Observable: You should no longer deep import custom Observable implementations such as ArrayObservable or ForkJoinObservable.
   - `import rxjs/**/observable/EmptyObservable ⇝ ∅`
   - `call <rxjs{,/**/*}>.EmptyObservable.create ⇝ <rxjs>.EMPTY`

     OBJ: *Is the receiver EmptyObservable from rxjs?*
   - `import rxjs/**/observable/{ArrayLikeObservable,BoundNodeCallbackObservable,DeferObservable,ForkJoinObservable,RangeObservable,TimerObservable,ArrayObservable,FromObservable,IfObservable,PairsObservable,ScalarObservable,FromEventObservable,NeverObservable,PromiseObservable,SubscribeOnObservable,UsingObservable,ConnectableObservable,ErrorObservable,IntervalObservable,BoundCallbackObservable,FromEventPatternObservable,GenerateObservable,IteratorObservable} ⇝ ∅`
   - `read <rxjs/**/observable/{ArrayLikeObservable,FromObservable,IteratorObservable,PromiseObservable}>.create ⇝ <rxjs>.from`

     OBJ: *Is the receiver one of ArrayLikeObservable, FromObservable, IteratorObservable or PromiseObservable from rxjs?*
   - `read <rxjs/**/observable/{ArrayObservable,ScalarObservable}>.create ⇝ <rxjs>.of`

     OBJ: *Is the receiver ArrayObservable or ScalarObservable from rxjs?*
   - `read <rxjs/**/observable/BoundCallbackObservable>.create ⇝ <rxjs>.bindCallback`

     OBJ: *Is the receiver BoundCallbackObservable from rxjs?*
   - `read <rxjs/**/observable/BoundNodeCallbackObservable>.create ⇝ <rxjs>.bindNodeCallback`

     OBJ: *Is the receiver BoundNodeCallbackObservable from rxjs?*
   - `read <rxjs/**/observable/DeferObservable>.create ⇝ <rxjs>.defer`

     OBJ: *Is the receiver DeferObservable from rxjs?*
   - `read <rxjs/**/observable/EmptyObservable>.create ⇝ <rxjs>.empty`

     OBJ: *Is the receiver EmptyObservable from rxjs?*
   - `read <rxjs/**/observable/ErrorObservable>.create ⇝ <rxjs>.throwError`

     OBJ: *Is the receiver ErrorObservable from rxjs?*
   - `read <rxjs/**/observable/ForkJoinObservable>.create ⇝`

`<rxjs>.forkJoin`

OBJ: *Is the receiver ForkJoinObservable from rxjs?*

- `read <rxjs/**/observable/FromEventObservable>.create ⇝ <rxjs>.fromEvent`

  OBJ: *Is the receiver FromEventObservable from rxjs?*
- `read <rxjs/**/observable/FromEventPatternObservable>.create ⇝ <rxjs>.fromEventPattern`

  OBJ: *Is the receiver FromEventPatternObservable from rxjs?*
- `read <rxjs/**/observable/GenerateObservable>.create ⇝ <rxjs>.generate`

  OBJ: *Is the receiver GenerateObservable from rxjs?*
- `read <rxjs/**/observable/IfObservable>.create ⇝ <rxjs>.iif`

  OBJ: *Is the receiver IfObservable from rxjs?*
- `read <rxjs/**/observable/IntervalObservable>.create ⇝ <rxjs>.interval`

  OBJ: *Is the receiver IntervalObservable from rxjs?*
- `read <rxjs/**/observable/NeverObservable>.create ⇝ <rxjs>.never`

  OBJ: *Is the receiver NeverObservable from rxjs?*
- `read <rxjs/**/observable/PairsObservable>.create ⇝ <rxjs>.pairs`

  OBJ: *Is the receiver PairsObservable from rxjs?*
- `read <rxjs/**/observable/RangeObservable>.create ⇝ <rxjs>.range`

  OBJ: *Is the receiver RangeObservable from rxjs?*
- `read <rxjs/**/observable/TimerObservable>.create ⇝ <rxjs>.timer`

  OBJ: *Is the receiver TimerObservable from rxjs?*
- `read <rxjs/**/observable/UsingObservable>.create ⇝ <rxjs>.using`

  OBJ: *Is the receiver UsingObservable from rxjs?*

11) _throw: _throw is now exported as throwError
    - `read <rxjs/**/observable/throw>._throw ⇝ <rxjs>.throwError`

      OBJ: *Is the receiver from rxjs?*
12) if: if is now exported as iif
    - `import rxjs/**/if ⇝ <rxjs>.iif`
    - `read <rxjs>.Observable.if ⇝ <rxjs>.iif`

      OBJ: *Is the receiver rxjs.Observable.if?*
13) operators removed: Operator versions of static observable creators such as merge, concat, zip, onErrorResumeNext, and race have been removed. Please use the static versions of those operations. e.g. a.pipe(concat(b, c)) becomes concat(a, b, c).
    - `import rxjs/operator/{merge,concat,zip,onErrorResumeNext,race}{,/**/*} ⇝ ∅`
    - `call (<rxjs{,/**/*}>?** \ <rxjs{,/**/*}>.Observable).{merge,concat,zip,onErrorResumeNext,race} ⇝ <rxjs>.$prop($base, $args)`

      OBJ: *Is the receiver a rxjs observable?*
14) Symbol.observable: RxJS will no longer be polyfilling Symbol.observable. That should be done by an actual polyfill library. This is to prevent duplication of code, and also to prevent having modules with side-effects in rxjs.
    - `read <Symbol>.observable ⇝ ((Symbol && Symbol.observable) || '@@observable')`

      OBJ: *Is the receiver the builtin Symbol constructor?*

      EXTRA: *Did the user depend on RxJS polyfilling Symbol.observable?*
15) Rx.ts: importing from rxjs/Rx is no longer available. Upcoming backwards compat solution will allow that
    - `import rxjs/Rx ⇝ <rxjs>`
16) never: no longer exported. Use the NEVER constant instead.
    - `import rxjs/**/never ⇝ <rxjs>.never`

17) ajax: will no longer execute a CORS request by default, you must opt-in with the crossDomain flag in the config.
- `call {<rxjs{,/**/*}>,<rxjs{,/**/*}>.Observable}.ajax 0:string ⇝ $callee({ url: $1, crossDomain: true})`
  OBJ: *Is the function being called the ajax function from rxjs?*
  EXTRA: *Is the request a CORS request and the crossDomain flag not set?*
- `call {<rxjs{,/**/*}>,<rxjs{,/**/*}>.Observable}.ajax 0:object ⇝ $callee(Object.assign({}, $1, {crossDomain: true})`
  OBJ: *Is the function being called the ajax function from rxjs?*
  EXTRA: *Is the request a CORS request and the crossDomain flag not set?*
- `call <rxjs/**/operator/ajax{,*,/**/*}> 0:string ⇝ $callee({ url: $1, crossDomain: true})`
  OBJ: *Is the function being called the ajax function from rxjs?*
  EXTRA: *Is the request a CORS request and the crossDomain flag not set?*
- `call <rxjs/**/operator/ajax{,*,/**/*}> 0:object ⇝ $callee(Object.assign({}, $1, {crossDomain: true})`
  OBJ: *Is the function being called the ajax function from rxjs?*
  EXTRA: *Is the request a CORS request and the crossDomain flag not set?*

18) websocket: WebSocketSubject will now JSON serialize all messages sent over it by default, to return to the old behavior, pass a config setting of serializer: x => x like so: websocket({ url, serializer: x => x })
- `call <rxjs/**/observable/dom/{webSocket,WebSocketSubject}{,*,/**/*}> 0:string ⇝ $callee({url: $1, serializer: x => x})`
  OBJ: *Is the function being called the webSocket or WebSocketSubject function from rxjs?*
- `call <rxjs/**/observable/dom/{webSocket,WebSocketSubject}{,*,/**/*}> 0:object ⇝ $callee(Object.assign({}, $1, {serializer: x => x}))`
  OBJ: *Is the function being called the webSocket or WebSocketSubject function from rxjs?*

19) Removed fromPromise, should use from instead.
- `import rxjs/observable/fromPromise ⇝ <rxjs>.from`
- `read <rxjs{,/**/*}>.Observable.fromPromise ⇝ <rxjs>.from`
  OBJ: *Is the receiver rxjs.Observable.fromPromise?*

20) Removed default import such that import Rx from 'rxjs'; fails. Instead import * as Rx from 'rxjs'; works
- `importD rxjs{,/Rx} ⇝ <import * as Rx from 'rxjs'>`

21) Dropping support for chaining operators (use pipe instead)
- ```
  call (<rxjs{,/**/*}>?** \ <rxjs{,/**/*}/operators>)
       .{flatMap,map,mapTo,filter,take,takeUntil,takeWhile,delay
        ,last,null,do,catch,finally,switch,withLatestFrom,startWith
        ,timeout,defaultIfEmpty,scan,debounceTime,throttleTime
        ,share,mergeMap,retryWhen,timestamp,mergeMapTo
        ,distinctUntilChanged,switchMap,bufferToggle,concatMap
        ,windowWhen,concatAll,first,toArray,isEmpty,mergeAll
        ,groupBy,reduce,buffer} ⇝
  $base.pipe(<rxjs/operators>.$prop($args))
  ```
  OBJ: *Is the receiver a rxjs observable?*
- `import rxjs/add{,/**/*} ⇝ ∅`

22) Operator renames: do -> tap, catch -> catchError, switch -> switchAll, finally -> finalize, throw -> throwError
- ```
  read <rxjs{,/**/*}>?**.{do,catch,switch,finally} ⇝
  $base.$prop[do => tap, catch => catchError, switch => switchAll
              , finally => finalize]
  ```
  OBJ: *Is the receiver a rxjs observable?*
- `import rxjs/**/throw ⇝ <rxjs/throwError>`
- `read <rxjs{,/**/*}>.Observable.throw ⇝ <rxjs>.throwError`
  OBJ: *Is the receiver rxjs.Observable.throw?*

23) rxjs@6.0.0 is incompatible with the npm package symbol-observable
- ```
  importD symbol-observable ⇝
  const $locName = (Symbol && Symbol.observable) || '@@observable';
  ```
  ```

core-js 3.0.0

1) update: asap (old stage 0 proposal) replaced by queueMicrotask (a part of HTML spec)
   - `import core-js/**/*asap ⇝ <core-js/#1/#2queueMicrotask>`
   - `read {<core-js>,<global>}.asap ⇝ $base.queueMicrotask`

     OBJ: *Is the receiver core-js or the global object?*
2) update: Update Observable (#257, #276, etc.)
   - A semantic patch could not be expressed for the pattern:
     `call <core-js/**/*observable>`
3) update: Update Array#flatten -> Array#flat and Array#flatMap
   - `import core-js/**/flatten ⇝ <import/#1flat>`
   - `read {<core-js>?,<Array>?}**.flatten ⇝ $base.flat`

     OBJ: *Is the receiver an array or the core-js object?*
4) update: Update String#matchAll (proposal-string-matchall#17, proposal-string-matchall#38, proposal-string-matchall#41, etc.) and move to the stage 3
   - No general patch exists for the pattern:
     `call {<core-js>?,<String>?}**.matchAll`
5) update: Update .name properties of String#{trimStart, trimEnd , trimLeft, trimRight}, move to the stage 3
   - `import core-js/**/*{trim-left,trim-right} ⇝`
     `<core-js/#1/#2#3[trim-left => trim-start, trim-right => trim-end]>`
6) remove obsolete: Error.isError (withdrawn)
   - `import core-js/**/is-error ⇝ ∅`
   - `readO <Error>.isError ⇝`
     `((arg) => Object.prototype.toString`
     `.call(arg).slice(8, -1) === 'Error')`

     OBJ: *Is the receiver the builtin Error constructor?*
   - `call <Error>.isError ⇝`
     `(Object.prototype.toString($1).slice(8, -1) === 'Error')`

     OBJ: *Is the receiver the builtin Error constructor?*
7) remove obsolete: System.global and global (replaced by globalThis)
   - `read <System>.global ⇝ globalThis`

     OBJ: *Is the receiver System?*
8) remove obsolete: Map#toJSON and Set#toJSON (rejected)
   - `read <Map>?**.toJSON ⇝`
     `((() => {<map-tojson>.shim(); return $base.$prop})())`

     OBJ: *Is the receiver a Map?*
   - `read <Set>?**.toJSON ⇝`
     `((() => {<set-tojson>.shim(); return $base.$prop})())`

     OBJ: *Is the receiver a Set?*
9) remove obsolete: RegExp.escape (rejected)
   - `import core-js/**/escape ⇝ ∅`
   - `read <RegExp>.escape ⇝ <regexp.escape>`

     OBJ: *Is the receiver the builtin RegExp constructor?*
10) remove obsolete: Reflect.enumerate (removed from the spec)
    - No general patch exists for the pattern:
      `read <Reflect>.enumerate`
11) remove: Dict
    - No general patch exists for the pattern:
      `call <Dict>`
12) remove: Object.{classof, isObject, define, make}
    - `import core-js/**/object/{classof,is-object,define,make} ⇝ ∅`
    - No general patch exists for the pattern:
      `read <Object>.{classof,isObject,define,make}`

13) remove: Function#part
    - `import core-js/**/part ⇝ ∅`
    - No general patch exists for the pattern:
      `read {<core-js>?,<Function>?}**.part`
14) remove: String#{escapeHTML, unescapeHTML}
    - `import core-js/**/*{escape-html,unescape-html} ⇝ ∅`
    - `call <String>?**.escapeHTML ⇝ <escape-html>($base)`
      OBJ: *Is the receiver a string?*
    - `call <String>?**.unescapeHTML ⇝ <unescape-html>($base)`
      OBJ: *Is the receiver a string?*
15) remove: delay
    - `import core-js/**/delay ⇝ ∅`
    - `readO <core-js/**/>.delay ⇝`
      ```
      ((time) => new Promise(function (resolve) {
            setTimeout(resolve(true), time);
          }));
      ```
      OBJ: *Is the delay function from core-js?*
    - `call {<core-js/**/>.delay,<core-js/**/delay>} ⇝`
      ```
      (new Promise(function (resolve) {
            setTimeout(resolve(true), $1);
          }));
      ```
      OBJ: *Is the delay function from core-js?*
16) Package-related: Leave only one pair of bundles (global, with all polyfills) and move it to core-js-bundle package.
    - `import core-js/{client,client/**/*} ⇝ <core-js-bundle>`
17) Commonjs API/namespace: Move core-js/library to separate core-js-pure package.
    - `import core-js/library/{es5,es6}/* ⇝ <core-js-pure/es/#2>`
    - `import core-js/library/es7/* ⇝ <core-js-pure/features/#1>`
18) Commonjs API/namespace: Because of removing all non-standard features, we no longer need core-js/shim entry point, replace it just with core-js.
    - `import core-js/{shim,shim/**/*} ⇝ <core-js>`
19) Commonjs API/namespace: Move all features from ES5, ES2015, ES2016, ES2017, ES2018 and ES2019 to one namespace for stable ES - it's available as core-js/es, all those features in modules folder has es. prefix.
    - ```
      import core-js/{es5,es6,es5/**/*,es6/**/*} ⇝
      <core-js/#1[es5 => es, es6 => es, es6/array => es/array
          , es6/function => es/function, es6/map => es/map
          , es6/number => es/number
          , es6/parse-float => es/number/parse-float
          , es6/promise => es/promise, es6/regexp => es/regexp
          , es6/string => es/string, es6/typed => es/typed-array
          , es6/weak => es/weak-set, es6/date => es/date
          , es6/math => es/math, es6/object => es/object
          , es6/parse-int => es/number/parse-int
          , es6/reflect => es/reflect, es6/set => es/set
          , es6/symbol => es/symbol, es6/weak => es/weak-map]>
      ```
    - `import core-js/es7{,/**/*} ⇝ <core-js/es#1>`
    - ```
      import core-js/modules/
             {es6.array.copy-within,es6.array.every,es6.array.fill
             ,es6.array.filter,es6.array.find-index,es6.array.find
             ,es6.array.for-each,es6.array.from,es6.array.index-of
             ,es6.array.is-array,es6.array.iterator,es6.array.join
             ,es6.array.last-index-of,es6.array.map,es6.array.of
             ,es6.array.reduce-right,es6.array.reduce,es6.array.slice
             ,es6.array.some,es6.array.sort,es6.array.species,es6.date.now
             ,es6.date.to-iso-string,es6.date.to-json,es6.date.to-primitive
             ,es6.date.to-string,es6.function.bind,es6.function.has-instance
             ,es6.function.name,es6.map,es6.math.acosh,es6.math.asinh
             ,es6.math.atanh,es6.math.cbrt,es6.math.clz32,es6.math.cosh
             ,es6.math.expm1,es6.math.fround,es6.math.hypot,es6.math.imul
             ,es6.math.log1p,es6.math.log2,es6.math.log10,es6.math.sign
      ```

```
        ,es6.math.sinh,es6.math.tanh,es6.math.trunc,es6.number.constructor
        ,es6.number.epsilon,es6.number.is-finite,es6.number.is-integer
        ,es6.number.is-nan,es6.number.is-safe-integer
        ,es6.number.max-safe-integer,es6.number.min-safe-integer
        ,es6.number.parse-float,es6.number.parse-int
        ,es6.number.to-fixed,es6.number.to-precision,es6.object.assign
        ,es6.object.create,es6.object.define-properties
        ,es6.object.define-property,es6.object.freeze
        ,es6.object.get-own-property-descriptor
        ,es6.object.get-own-property-names,es6.object.get-prototype-of
        ,es6.object.is-extensible,es6.object.is-frozen,es6.object.is-sealed
        ,es6.object.is,es6.object.keys,es6.object.prevent-extensions
        ,es6.object.seal,es6.object.set-prototype-of,es6.object.to-string
        ,es6.parse-float,es6.parse-int,es6.promise,es6.reflect.apply
        ,es6.reflect.construct,es6.reflect.define-property
        ,es6.reflect.delete-property,es6.reflect.get-own-property-descriptor
        ,es6.reflect.get-prototype-of,es6.reflect.get,es6.reflect.has
        ,es6.reflect.is-extensible,es6.reflect.own-keys
        ,es6.reflect.prevent-extensions,es6.reflect.set-prototype-of
        ,es6.reflect.set,es6.regexp.constructor
        ,es6.regexp.exec,es6.regexp.flags,es6.regexp.to-string,es6.set
        ,es6.string.anchor,es6.string.big,es6.string.blink
        ,es6.string.bold,es6.string.code-point-at,es6.string.ends-with
        ,es6.string.fixed,es6.string.fontcolor,es6.string.fontsize
        ,es6.string.from-code-point,es6.string.includes
        ,es6.string.italics,es6.string.iterator,es6.string.link
        ,es6.string.raw,es6.string.repeat,es6.string.small
        ,es6.string.starts-with,es6.string.strike,es6.string.sub
        ,es6.string.sup,es6.string.trim,es6.symbol
        ,es6.typed.array-buffer,es6.typed.data-view
        ,es6.typed.float32-array,es6.typed.float64-array
        ,es6.typed.int8-array,es6.typed.int16-array
        ,es6.typed.int32-array,es6.typed.uint8-array
        ,es6.typed.uint8-clamped-array,es6.typed.uint16-array
        ,es6.typed.uint32-array,es6.weak-map,es6.weak-set
        ,es7.array.flat-map,es7.array.flatten,es7.array.includes
        ,es7.object.define-getter,es7.object.define-setter
        ,es7.object.entries,es7.object.get-own-property-descriptors
        ,es7.object.lookup-getter,es7.object.lookup-setter
        ,es7.object.values,es7.promise.finally,es7.string.pad-end
        ,es7.string.pad-start,es7.string.trim-left
        ,es7.string.trim-right,es7.symbol.async-iterator} ⇝
<core-js/modules/#1[es6.object.is => es.object.is
    , es7.string.trim-left => es.string.trim-left
    , es6.object.is-sealed => es.object.is-sealed
    , es7.string.trim-right => es.string.trim-right
    , es6.object.keys => es.object.keys
    , es7.symbol.async-iterator => es.symbol.async-iterator
    , es6.object.prevent-extensions => es.object.prevent-extensions
    , es7.symbol.observable => es.symbol.observable
    , es6.object.seal => es.object.seal
    , es7.system.global => es.system.global
    , es6.object.set-prototype-of => es.object.set-prototype-of
    , es7.weak-map.from => es.weak-map.from
    , es6.object.to-string => es.object.to-string
    , es7.weak-map.of => es.weak-map.of
    , es6.parse-float => es.parse-float
    , es7.weak-set.from => es.weak-set.from
    , es6.parse-int => es.parse-int, es7.weak-set.of => es.weak-set.of
    , es6.promise => es.promise, es6.reflect.apply => es.reflect.apply
    , es6.reflect.construct => es.reflect.construct
    , es6.reflect.define-property => es.reflect.define-property
    , es6.reflect.delete-property => es.reflect.delete-property
    , es6.reflect.enumerate => es.reflect.enumerate
    , es6.reflect.get => es.reflect.get
    , es6.reflect.get-own-property-descriptor =>
        es.reflect.get-own-property-descriptor
    , es6.reflect.get-prototype-of => es.reflect.get-prototype-of
    , es6.reflect.has => es.reflect.has
    , es6.reflect.is-extensible => es.reflect.is-extensible
    , es6.reflect.own-keys => es.reflect.own-keys
    , es6.reflect.prevent-extensions => es.reflect.prevent-extensions
    , es6.reflect.set => es.reflect.set
    , es6.reflect.set-prototype-of => es.reflect.set-prototype-of
```

```
, es6.regexp.constructor => es.regexp.constructor
, es6.regexp.exec => es.regexp.exec
, es6.regexp.flags => es.regexp.flags
, es6.regexp.match => es.string.match
, es6.regexp.replace => es.string.replace
, es6.regexp.search => es.string.search
, es6.regexp.split => es.string.split
, es6.regexp.to-string => es.regexp.to-string, es6.set => es.set
, es6.string.anchor => es.string.anchor
, es6.string.big => es.string.big
, es6.string.blink => es.string.blink
, es6.string.bold => es.string.bold
, es6.string.code-point-at => es.string.code-point-at
, es6.string.ends-with => es.string.ends-with
, es6.string.fixed => es.string.fixed
, es6.string.fontcolor => es.string.fontcolor
, es6.string.fontsize => es.string.fontsize
, es6.string.from-code-point => es.string.from-code-point
, es6.string.includes => es.string.includes
, es6.string.italics => es.string.italics
, es6.string.iterator => es.string.iterator
, es6.string.link => es.string.link
, es6.array.copy-within => es.array.copy-within
, es6.string.raw => es.string.raw, es6.array.every => es.array.every
, es6.string.repeat => es.string.repeat
, es6.array.fill => es.array.fill
, es6.string.small => es.string.small
, es6.array.filter => es.array.filter
, es6.string.starts-with => es.string.starts-with
, es6.array.find-index => es.array.find-index
, es6.string.strike => es.string.strike
, es6.array.find => es.array.find
, es6.string.sub => es.string.sub
, es6.array.for-each => es.array.for-each
, es6.string.sup => es.string.sup
, es6.array.from => es.array.from
, es6.string.trim => es.string.trim
, es6.array.index-of => es.array.index-of
, es6.symbol => es.symbol, es6.array.is-array => es.array.is-array
, es6.typed.array-buffer => es.typed-array.array-buffer
, es6.array.iterator => es.array.iterator
, es6.typed.data-view => es.typed-array.data-view
, es6.array.join => es.array.join
, es6.typed.float32-array => es.typed-array.float32-array
, es6.array.last-index-of => es.array.last-index-of
, es6.typed.float64-array => es.typed-array.float64-array
, es6.array.map => es.array.map
, es6.typed.int16-array => es.typed-array.int16-array
, es6.array.of => es.array.of
, es6.typed.int32-array => es.typed-array.int32-array
, es6.array.reduce => es.array.reduce
, es6.typed.int8-array => es.typed-array.int8-array
, es6.array.reduce-right => es.array.reduce-right
, es6.typed.uint16-array => es.typed-array.uint16-array
, es6.array.slice => es.array.slice
, es6.typed.uint32-array => es.typed-array.uint32-array
, es6.array.some => es.array.some
, es6.typed.uint8-array => es.typed-array.uint8-array
, es6.array.sort => es.array.sort
, es6.typed.uint8-clamped-array => es.typed-array.uint8-clamped-array
, es6.array.species => es.array.species, es6.weak-map => es.weak-map
, es6.date.now => es.date.now, es6.weak-set => es.weak-set
, es6.date.to-iso-string => es.date.to-iso-string
, es7.array.flat-map => es.array.flat-map
, es6.date.to-json => es.date.to-json
, es7.array.flatten => es.array.flatten
, es6.date.to-primitive => es.date.to-primitive
, es7.array.includes => es.array.includes
, es6.date.to-string => es.date.to-string
, es7.asap => web.queue-microtask
, es6.function.bind => es.function.bind
, es7.error.is-error => es.error.is-error
, es6.function.has-instance => es.function.has-instance
, es7.global => es.global, es6.function.name => es.function.name
, es7.map.from => es.map.from, es6.map => es.map
```

```
, es7.map.of => es.map.of, es6.math.acosh => es.math.acosh
, es7.map.to-json => es.map.to-json, es6.math.asinh => es.math.asinh
, es7.math.clamp => es.math.clamp, es6.math.atanh => es.math.atanh
, es7.math.deg-per-rad => es.math.deg-per-rad
, es6.math.cbrt => es.math.cbrt, es7.math.degrees => es.math.degrees
, es6.math.clz32 => es.math.clz32, es7.math.fscale => es.math.fscale
, es6.math.cosh => es.math.cosh, es7.math.iaddh => es.math.iaddh
, es6.math.expm1 => es.math.expm1, es7.math.imulh => es.math.imulh
, es6.math.fround => es.math.fround, es7.math.isubh => es.math.isubh
, es6.math.hypot => es.math.hypot
, es7.math.radians => es.math.radians
, es6.math.imul => es.math.imul
, es7.math.rad-per-deg => es.math.rad-per-deg
, es6.math.log10 => es.math.log10, es7.math.scale => es.math.scale
, es6.math.log1p => es.math.log1p
, es7.math.signbit => es.math.signbit
, es6.math.log2 => es.math.log2, es7.math.umulh => es.math.umulh
, es6.math.sign => es.math.sign
, es7.object.define-getter => es.object.define-getter
, es6.math.sinh => es.math.sinh
, es7.object.define-setter => es.object.define-setter
, es6.math.tanh => es.math.tanh
, es7.object.entries => es.object.entries
, es6.math.trunc => es.math.trunc
, es7.object.get-own-property-descriptors =>
    es.object.get-own-property-descriptors
, es6.number.constructor => es.number.constructor
, es7.object.lookup-getter => es.object.lookup-getter
, es6.number.epsilon => es.number.epsilon
, es7.object.lookup-setter => es.object.lookup-setter
, es6.number.is-finite => es.number.is-finite
, es7.object.values => es.object.values
, es6.number.is-integer => es.number.is-integer
, es7.observable => es.observable
, es6.number.is-nan => es.number.is-nan
, es7.promise.finally => es.promise.finally
, es6.number.is-safe-integer => es.number.is-safe-integer
, es7.promise.try => es.promise.try
, es6.number.max-safe-integer => es.number.max-safe-integer
, es7.reflect.define-metadata => es.reflect.define-metadata
, es6.number.min-safe-integer => es.number.min-safe-integer
, es7.reflect.delete-metadata => es.reflect.delete-metadata
, es6.number.parse-float => es.number.parse-float
, es7.reflect.get-metadata => es.reflect.get-metadata
, es6.number.parse-int => es.number.parse-int
, es7.reflect.get-metadata-keys => es.reflect.get-metadata-keys
, es6.number.to-fixed => es.number.to-fixed
, es7.reflect.get-own-metadata => es.reflect.get-own-metadata
, es6.number.to-precision => es.number.to-precision
, es7.reflect.get-own-metadata-keys =>
    es.reflect.get-own-metadata-keys
, es6.object.assign => es.object.assign
, es7.reflect.has-metadata => es.reflect.has-metadata
, es6.object.create => es.object.create
, es7.reflect.has-own-metadata => es.reflect.has-own-metadata
, es6.object.define-properties => es.object.define-properties
, es7.reflect.metadata => es.reflect.metadata
, es6.object.define-property => es.object.define-property
, es7.set.from => es.set.from, es6.object.freeze => es.object.freeze
, es7.set.of => es.set.of
, es6.object.get-own-property-descriptor =>
    es.object.get-own-property-descriptor
, es7.set.to-json => es.set.to-json
, es6.object.get-own-property-names =>
    es.object.get-own-property-names, es7.string.at => es.string.at
, es6.object.get-prototype-of => es.object.get-prototype-of
, es7.string.match-all => es.string.match-all
, es6.object.is-extensible => es.object.is-extensible
, es7.string.pad-end => es.string.pad-end
, es6.object.is-frozen => es.object.is-frozen
, es7.string.pad-start => es.string.pad-start]>
```

20) Commonjs API/namespace: Change prefix for ES proposals from es7. to esnext., they no longer available in

core-js/es7, use core-js/stage/* instead of that.

- `import core-js/modules/{es7.map.from,es7.map.of,es7.math.clamp,es7.math.deg-per-rad,es7.math.degrees,es7.math.fscale,es7.math.iaddh,es7.math.imulh,es7.math.isubh,es7.math.rad-per-deg,es7.math.radians,es7.math.scale,es7.math.signbit,es7.math.umulh,es7.observable,es7.promise.try,es7.reflect.define-metadata,es7.reflect.delete-metadata,es7.reflect.get-metadata,es7.reflect.get-metadata-keys,es7.reflect.get-own-metadata,es7.reflect.get-own-metadata-keys,es7.reflect.has-metadata,es7.reflect.has-own-metadata,es7.reflect.metadata,es7.set.of,es7.string.at,es7.string.match-all,es7.symbol.observable,es7.weak-map.from,es7.weak-map.of,es7.weak-set.from,es7.weak-set.of}` ⇝

```
<core-js/modules/#1[es7.map.from => esnext.map.from
    , es7.map.of => esnext.map.of, es7.math.clamp => esnext.math.clamp
    , es7.math.deg-per-rad => esnext.math.deg-per-rad
    , es7.math.degrees => esnext.math.degrees
    , es7.math.fscale => esnext.math.fscale
    , es7.math.iaddh => esnext.math.iaddh
    , es7.math.imulh => esnext.math.imulh
    , es7.math.isubh => esnext.math.isubh
    , es7.math.rad-per-deg => esnext.math.rad-per-deg
    , es7.math.radians => esnext.math.radians
    , es7.math.scale => esnext.math.scale
    , es7.math.signbit => esnext.math.signbit
    , es7.math.umulh => esnext.math.umulh
    , es7.observable => esnext.observable
    , es7.promise.try => esnext.promise.try
    , es7.reflect.define-metadata => esnext.reflect.define-metadata
    , es7.reflect.delete-metadata => esnext.reflect.delete-metadata
    , es7.reflect.get-metadata => esnext.reflect.get-metadata
    , es7.reflect.get-metadata-keys => esnext.reflect.get-metadata-keys
    , es7.reflect.get-own-metadata => esnext.reflect.get-own-metadata
    , es7.reflect.get-own-metadata-keys =>
        esnext.reflect.get-own-metadata-keys
    , es7.reflect.has-metadata => esnext.reflect.has-metadata
    , es7.reflect.has-own-metadata => esnext.reflect.has-own-metadata
    , es7.reflect.metadata => esnext.reflect.metadata
    , es7.set.of => esnext.set.of
    , es7.string.at => esnext.string.at
    , es7.string.match-all => esnext.string.match-all
    , es7.symbol.observable => esnext.symbol.observable
    , es7.weak-map.from => esnext.weak-map.from
    , es7.weak-map.of => esnext.weak-map.of
    , es7.weak-set.from => esnext.weak-set.from
    , es7.weak-set.of => esnext.weak-set.of]>
```

21) Commonjs API/namespace: Rename core-js(/library)/fn to core-js(-pure)/features for improve readability.
    - `import core-js/fn{,/**/*}` ⇝ `<core-js/features#1>`
    - `import core-js/library/fn{,/**/*}` ⇝ `<core-js-pure/features#1>`
22) Commonjs API/namespace: Rename web.dom namespace to web.dom-collections.
    - `import core-js/modules/web.dom.*` ⇝ `<core-js/internals/dom-iterables>`
23) Commonjs API/namespace: Relax /modules/ directory by moving internal modules to /internals/ directory.
    - `import core-js/modules/_*` ⇝ `<core-js/internals/#1>`
24) Commonjs API/namespace: Remove deprecated array entry points: core-js(/library)/fn/array/{pop, push, reverse, shift, unshift}.
    - `import core-js{/library/,/}fn/array/{pop,push,reverse,shift,unshift}` ⇝ ∅
25) Commonjs API/namespace: core object no longer available in the global version, entry points which previously returned it now returns globalThis object. Also, don't set global core property.
    - `read <global>.core` ⇝ `$base`

      OBJ: *Is the receiver the global object?*

node-fetch 2.0.0

1) Major: require('node-fetch/lib/response') etc. is now unsupported; use require('node-fetch').Response or ES6 module imports
   - `import node-fetch/lib/**/* ⇝ <node-fetch>.#2[fetch-error => FetchError, response => Response , request => Request, body => Body, headers => Headers]`
2) Major: response.text() no longer attempts to detect encoding, instead always opting for UTF-8 (per spec); use response.textConverted() for the v1 behavior
   - `call <node-fetch>?**.text ⇝ $base.textConverted($args)`
     OBJ: *Is the receiver a node-fetch response?*
     EXTRA: *Is the response text not encoded in UTF-8?*
     MINOR: *Ask before patching.*
3) Major: make response.json() throw error instead of returning an empty object on 204 no-content respose (per spec; reverts behavior changed in v1.6.2)
   - `call <node-fetch>?**.json ⇝ ($base.status === 204 ? {} : $base.json())`
     OBJ: *Is the receiver a node-fetch response?*
     EXTRA: *Can the response code be 204?*
     MINOR: *Ask before patching.*
4) Major: internal methods are no longer exposed
   - No general patch exists for the pattern:
     `call <node-fetch>.{_clone,_decode,_convert}`
5) Major: throw error when a GET or HEAD Request is constructed with a non-null body (per spec)
   - `call <node-fetch> [2, 2] ⇝ $callee($1, Object.assign($2, {body: null}))`
     OBJ: *Is the function being called node-fetch?*
     EXTRA: *Is the second argument a GET or HEAD Request with non-null body?*
6) Major: remove headers.getAll(); make get() return all headers delimited by commas (per spec)
   - No general patch exists for the pattern:
     `call <node-fetch>?**.getAll`
7) Enhance: make sure header names and values are valid in HTTP
   - `call <node-fetch>?**.{set,append,has,delete} [1, 2] 0:string ⇝`
     ```
     try {
       $callee($args)
     } catch (_e) {}
     ```
     OBJ: *Is the receiver a node-fetch headers object?*
     EXTRA: *Is the header name valid?*
     MINOR: *Ask before patching.*
8) Enhance: add response.arrayBuffer() (also applies to Requests)
   - `read <node-fetch>?**.arrayBuffer ⇝ undefined`
     OBJ: *Is the receiver a node-fetch request or response object?*
9) request.url can no longer be written in a property write
   - `write <node-fetch>?**.url ⇝ Object.defineProperty($base, 'url', {value: $value})`
     OBJ: *Is the receiver a node-fetch request?*

winston 3.0.0

1) winston.Logger has been replaced with winston.createLogger.
   - `call <winston>.Logger ⇝ $base.createLogger($args)`
     OBJ: *Is the function being called winston.Logger?*
   - `read0 <winston>.Logger ⇝ $base.createLogger().constructor.__proto__`
     OBJ: *Is the property being read winston.Logger?*

2) winston.setLevels has been removed. Levels are frozen at the time of Logger creation.
   - No general patch exists for the pattern:
     `call <winston>.setLevels`

3) winston.transports.Memory was removed. Use any Node.js stream.Writeable with a large highWaterMark instance instead.
   - `readO <winston>.transports.Memory ⇝ $base.Stream`

     OBJ: *Is the receiver winston.transports?*
   - `call <winston>.transports.Memory [0, 0] ⇝ $base.Stream({stream: new <stream>.Writable()})`

     OBJ: *Is the receiver winston.transports?*
   - `call <winston>.transports.Memory [1, 1] ⇝`

```
$base.Stream(Object.assign({stream: new <stream>.Writable()}),
    (function(opts) {
      const newOpts = {};
      const formatArray = [];
      const formatOptions = {
        stringify: () => <winston>.format((info) => {
            info.message = JSON.stringify(info.message);
        })(),
        formatter: () => <winston>.format((info) => {
            info.message = opts.formatter(Object.assign(info, opts));
        })(),
        json: () => <winston>.format.json(),
        raw: () => <winston>.format.json(),
        label: () => <winston>.format.label(opts.label),
        logstash: () => <winston>.format.logstash(),
        prettyPrint: () => <winston>.format
            .prettyPrint({depth: opts.depth || 2}),
        colorize: () => <winston>.format.colorize({
            level: opts.colorize === true || opts.colorize === 'level',
            all: opts.colorize === 'all',
            message: opts.colorize === 'message'
        }),
        timestamp: () => <winston>.format.timestamp(),
        align: () => <winston>.format.align(),
        showLevel: () => <winston>.format((info) => {
            info.message = info.level + ': ' + info.message;
        })()
      }
      Object.keys(opts).filter(k => !formatOptions.hasOwnProperty(k))
          .forEach((k) => { newOpts[k] = opts[k]; });
      Object.keys(opts).filter(k => formatOptions.hasOwnProperty(k) &&
          formatOptions[k])
              .forEach(k => formatArray.push(formatOptions[k]()));
      newOpts.format = <winston>.format.combine(...formatArray);
      return newOpts;
    })($1))
```

     OBJ: *Is the receiver winston.transports?*

4) When writing transports use winston-transport instead of winston.Transport.
   - `read <winston>.Transport ⇝ <winston-transport>`

     OBJ: *Is the receiver winston?*

5) In winston.transports.Console, output for all log levels are now sent to stdout by default. - stderrLevels option now defaults to []. - debugStdout option has been removed.
   - `call <winston>.transports.Console [0, 0] ⇝ $callee({stderrLevels: ['error', 'debug']})`

     OBJ: *Is the receiver winston.transports?*

     MINOR: *Ask before patching.*
   - `call <winston>.transports.Console [1, 1] ⇝`

```
$callee(((opts) => Object.assign({}, opts, opts.stderrLevels ? {}
    : {stderrLevels: opts.debugStdout ? ['error'] : ['error', 'debug']}))
    ($1))
```

     OBJ: *Is the receiver winston.transports?*

     MINOR: *Ask before patching.*
   - `call <winston>.transports.Console [2, 2] ⇝`

```
$callee(((opts) => Object.assign({}, opts, opts.stderrLevels ? {}
    : {stderrLevels: opts.debugStdout ? ['error'] : ['error', 'debug']}))
    ($1), $2)
```

OBJ: *Is the receiver winston.transports?*

MINOR: *Ask before patching.*

6) winston.Container instances no longer have default Console transports

- ```
  call <winston>.Container [0, 0] ⇝
  $callee({
      transports: [
        new winston.transports.Console({
          level: 'silly',
          colorize: false
        })
      ]
    })
  ```

  OBJ: *Is the receiver winston?*
- ```
  call <winston>.Container [1, 1] ⇝
  $callee(Object.assign($1, {
      transports: [
        new winston.transports.Console({
          level: 'silly',
          colorize: false
        })
      ]
    }))
  ```

  OBJ: *Is the receiver winston?*

  EXTRA: *Does the options object NOT include transports?*

7) winston.Container.prototype.add no longer does crazy options parsing. Implementation inspired by segmentio/winston-logger

- ```
  read <winston>.Container()?**.add ⇝
  $callee($1, (((opts) => {
    const existing = opts.transports || $base.options.transports;
    opts.transports = existing ? existing.slice() : [];
    Object.keys(opts).forEach(function (key) {
      if (key === 'transports' || key === 'filters'
          || key === 'rewriters') {
        return;
      }

      var name = key[0].toUpperCase() + key.slice(1);

      if (!<winston>.transports[name]) {
        throw new Error('Cannot add unknown transport: ' + name);
      }

      var namedOptions = opts[key];
      namedOptions.id = $1;
      opts.transports.push(new (<winston>.transports[name])(namedOptions));
    });
  })()))
  ```

  OBJ: *Is the receiver a winston Container?*

8) winston.Logger.log and level-specific methods (.info, .error, etc) no longer accepts a callback. The vast majority of use cases for this feature was folks awaiting all logging to complete, not just a single logging message. To accomplish this:

- No general patch exists for the pattern:
  `call <winston>?**.{info,warn,error} [2, 2] 1:function`
- No general patch exists for the pattern:
  `call <winston>?**.log [3, 3] 2:function`

9) winston.Logger.add no longer accepts prototypes / classes. Pass an instance of our transport instead.

- ```
  call <winston>?**.add [1, 1] 0:function ⇝
  $callee(new $1())
  ```

  OBJ: *Is the receiver a winston logger?*
- `call <winston>?**.add [2, 2] 0:function ⇝`

```
$callee(new $1((function(opts) {
      const newOpts = {};
      const formatArray = [];
      const formatOptions = {
        stringify: () => <winston>.format((info) => {
            info.message = JSON.stringify(info.message);
        })(),
        formatter: () => <winston>.format((info) => {
            info.message = opts.formatter(Object.assign(info, opts));
        })(),
        json: () => <winston>.format.json(),
        raw: () => <winston>.format.json(),
        label: () => <winston>.format.label(opts.label),
        logstash: () => <winston>.format.logstash(),
        prettyPrint: () => <winston>.format
            .prettyPrint({depth: opts.depth || 2}),
        colorize: () => <winston>.format.colorize({
            level: opts.colorize === true || opts.colorize === 'level',
            all: opts.colorize === 'all',
            message: opts.colorize === 'message'
        }),
        timestamp: () => <winston>.format.timestamp(),
        align: () => <winston>.format.align(),
        showLevel: () => <winston>.format((info) => {
            info.message = info.level + ': ' + info.message;
        })()
      }
      Object.keys(opts).filter(k => !formatOptions.hasOwnProperty(k))
         .forEach((k) => { newOpts[k] = opts[k]; });
      Object.keys(opts).filter(k => formatOptions.hasOwnProperty(k) &&
          formatOptions[k])
             .forEach(k => formatArray.push(formatOptions[k]()));
      newOpts.format = <winston>.format.combine(...formatArray);
      return newOpts;
    })($2)))
```

OBJ: *Is the receiver a winston logger?*

EXTRA: *Is the first argument one of winston.transports.File,Console,Http and the second argument an object containing formatting options?*

10) Logger.prototype.stream - options.transport is removed. Use the transport instance on the logger directly.

- ```
  call <winston>?**.stream [1, 1] ⇝
  (((opts) => {
      const transport = $base.transports[opts.transport]
      delete opts.transport;
      if (transport && transport.stream)
          return transport.stream(opts);
      return $callee(opts);
  })())
  ```

  OBJ: *Is the receiver a winston logger?*

  EXTRA: *Is the transport option set?*

11) Logger.prototype.query - options.transport is removed. Use the transport instance on the logger directly.

- ```
  call <winston>?**.query [2, 2] 1:function ⇝
  (((opts) => {
      const transport = $base.transports[opts.transport.toLowerCase()]
      if (opts.query) {
          opts.query = transport
              .formatQuery(Object.assign({}, opts.query));
      }

      transport.query(opts, function (err, results) {
          if (err) {
              return $2(err);
          }
          $2(null, transport.formatResults(results, opts.format));
      });
  })($1))
  ```

  OBJ: *Is the receiver a winston logger?*

  EXTRA: *Is the transport option set?*

12) winston.exception has been removed. Use: const exception = winston.ExceptionHandler();

- `read <winston>.exception ⇝ $base.ExceptionHandler()`
  OBJ: *Is the receiver winston?*

13) winston.hash was removed.
- `read0 <winston>.hash ⇝ ((str) => <crypto>.createHash('sha1').update(str).digest('hex'))`
  OBJ: *Is the receiver winston?*
- `call <winston>.hash ⇝ <crypto>.createHash('sha1').update($1).digest('hex')`
  OBJ: *Is the receiver winston?*

14) winston.common.pad was removed.
- `read0 <winston>.common.pad ⇝ ((n) => n < 10 ? '0' + n.toString(10) : n.toString(10))`
  OBJ: *Is the receiver winston.common?*
- `call <winston>.common.pad ⇝ ($1 < 10 ? '0' + $1.toString(10) : $1.toString(10))`
  OBJ: *Is the receiver winston.common?*

15) winston.common.serialized was removed (use winston-compat).
- `read <winston>.common.serialize ⇝ <winston-compat>.serialize`
  OBJ: *Is the receiver winston.common?*

16) winston.common.log was removed (use winston-compat).
- `read <winston>.common.log ⇝ <winston-compat>.log`
  OBJ: *Is the receiver winston.common?*

17) Removed winston.transports.{File,Console,Http} formatting options
- `call <winston>.transports.{File,Console,Http} [1, 1] ⇝`
```
new $callee((function(opts) {
      const newOpts = {};
      const formatArray = [];
      const formatOptions = {
        stringify: () => <winston>.format((info) => {
            info.message = JSON.stringify(info.message);
        })(),
        formatter: () => <winston>.format((info) => {
            info.message = opts.formatter(Object.assign(info, opts));
        })(),
        json: () => <winston>.format.json(),
        raw: () => <winston>.format.json(),
        label: () => <winston>.format.label(opts.label),
        logstash: () => <winston>.format.logstash(),
        prettyPrint: () => <winston>.format
            .prettyPrint({depth: opts.depth || 2}),
        colorize: () => <winston>.format.colorize({
            level: opts.colorize === true || opts.colorize === 'level',
            all: opts.colorize === 'all',
            message: opts.colorize === 'message'
        }),
        timestamp: () => <winston>.format.timestamp(),
        align: () => <winston>.format.align(),
        showLevel: () => <winston>.format((info) => {
            info.message = info.level + ': ' + info.message;
        })()
      }
      Object.keys(opts).filter(k => !formatOptions.hasOwnProperty(k))
          .forEach((k) => { newOpts[k] = opts[k]; });
      Object.keys(opts).filter(k => formatOptions.hasOwnProperty(k) &&
          formatOptions[k])
              .forEach(k => formatArray.push(formatOptions[k]()));
      newOpts.format = <winston>.format.combine(...formatArray);
      return newOpts;
    })($1))
```
  OBJ: *Is the receiver winston.transports?*

  EXTRA: *Does the options argument include formatting options?*

18) Migrating filters and rewriters to formats in winston@3

- A semantic patch could not be expressed for the pattern:
  `read <winston>?**.{rewriters,filters}`

19) Removed console property from transports
- `read <winston>?**.console ⇝`
  ```
  $base.filter(transport =>
      transport instanceof <winston>.transports.Console)[0]
  ```
  OBJ: *Is the receiver the transports object of a winston logger?*

20) winston.config.colorize is removed
- `read <winston>.config.colorize ⇝ winston.format.colorize().colorize`

  OBJ: *Is the receiver winston.config?*

redux 4.0.0

1) Throw if getState, subscribe, or unsubscribe called while dispatching (including inside a reducer) (#1569 by @mjw56)
- No general patch exists for the pattern:
  `call <redux{/**/*,}>?**.{getState,subscribe}`

2) Bundle cjs and es formats (#2358 by @TrySound) - (direct, private imports (import createStore from 'redux/lib/createStore') will no longer work)
- `import redux/**/* ⇝ <redux>.#2`

mongoose 5.0.0

1) BREAKING CHANGE: always use mongoose aggregation cursor when using .aggregate().cursor() #5941
- A semantic patch could not be expressed for the pattern:
  `write <mongoose>?**.cursor`

2) BREAKING CHANGE: attach query middleware when compiling model #5939
- A semantic patch could not be expressed for the pattern:
  `call <mongoose>?**.{pre,post} [2, 2] 0:string 1:function`

3) BREAKING CHANGE: remove passRawResult option for findOneAndUpdate, use rawResult #5869
- No general patch exists for the pattern:
  `write <mongoose>?**.passRawResult`

4) BREAKING CHANGE: implicit async validators (based on number of function args) are removed, return a promise instead #5824
- `call <mongoose>?**.validate 0:function2 ⇝`
  ```
  $callee(function (arg) {
    return new Promise(resolve => {
      ($1).call(this, arg, resolve);
    });
  }, $args[1,])
  ```
  OBJ: *Is the receiver a mongoose schema?*

5) BREAKING CHANGE: mapReduce resolves to an object with 2 keys rather than 2 separate args #5816
- `call <mongoose>?**.mapReduce ⇝`
  ```
  $callee($args[0, -1], function (errArg, resArg) {
    ($-1).call(this, errArg, resArg.results, resArg.stats);
  });
  ```
  OBJ: *Is the receiver a mongoose model?*

6) BREAKING CHANGE: mongoose.connect() returns a promise, removed MongooseThenable #5796
- `callR <mongoose>.connect [1, 1] ⇝`
  ```
  ((() => {
    $callee($args);
    return $base;
  })())
  ```
  OBJ: *Is the receiver mongoose?*

  EXTRA: *Is it NOT exploited that the result of connect was a MongooseThenable?*
- `callR <mongoose>.connect [2, 3] ⇝`
  ```
  ((() => {
    $callee($args);
  ```

```
  return $base.connection;
})())
```

OBJ: *Is the receiver mongoose?*

EXTRA: *Is it NOT exploited that the result of connect was a MongooseThenable?*

EXTRA: *Is the option useMongoClient enabled?*

- ```
  callR <mongoose>.createConnection [1, 3] ⇝
  ((() => {
    const conn = $callee();
    conn.openUri(connectionUrl).catch(e => e);
    return conn;
  })())
  ```

  OBJ: *Is the receiver mongoose?*

  EXTRA: *Is useMongoClient not enabled and is the return value not used as a Thenable?*

7) BREAKING CHANGE: query stream removed, use cursor() instead #5795
   - ```
     call <mongoose>?**.stream [0, 1] ⇝
     $base.cursor($args)
     ```

     OBJ: *Is the receiver a mongoose query?*

8) BREAKING CHANGE: connection open() and openSet() removed, use openUri() instead #5795
   - A semantic patch could not be expressed for the pattern:
     ```
     call <mongoose>?**.{open,openSet}
     ```

9) BREAKING CHANGE: remove support for $pushAll, remove usePushEach option #5670
   - ```
     write <mongoose>?**.usePushEach ⇝ ∅
     ```

     OBJ: *Is the receiver a mongoose schema?*
   - ```
     write <mongoose>?**.$pushAll ⇝
     $base.$push = Object.keys($value).reduce((acc, elem) =>
         {acc[elem] = {$each: $value[elem]}; return acc;}, {})
     ```

     OBJ: *Is the receiver a mongoose model?*

10) BREAKING CHANGE: remove saveErrorIfNotFound, always error out if save() did not update a document #4973
    - ```
      write <mongoose>?**.saveErrorIfNotFound ⇝ ∅
      ```

      OBJ: *Is the receiver a mongoose model?*

      EXTRA: *Is the object a mongoose schema?*
    - No general patch exists for the pattern:
      ```
      call <mongoose>?**.save
      ```

11) BREAKING CHANGE: don't execute getters in reverse order #4835
    - ```
      call <mongoose>?**.get().get [1, 1] 0:function ⇝
      $base:callee($args).get($base:args)
      ```

      OBJ: *Is the receiver a mongoose virtual type?*

      MINOR: *Ask before patching.*

12) BREAKING CHANGE: toObject() and toJSON() option parameter merges with defaults rather than overwriting #4131
    - ```
      call <mongoose>?**.{toObject,toJson} [1, 1] ⇝
      ((() => {
        const rcv = $base;
        const oldOptions = rcv.schema.options.$prop;
        const res = rcv.$prop($args);
        rcv.schema.options.$prop = oldOptions;
        return res;
      })())
      ```

      OBJ: *Is the receiver a mongoose document?*

13) BREAKING CHANGE: deleteX() and remove() promise resolves to the write object result #4013
    - No general patch exists for the pattern:
      ```
      callR <mongoose>?**.{remove,deleteOne,deleteMany}
      ```

14) BREAKING CHANGE: aggregate() no longer accepts a spread #2716
    - ```
      call <mongoose>?**.aggregate [2,] 0:object ⇝
      ```

`$callee([$args[0, -1]], $-1)`

OBJ: *Is the receiver a mongoose model?*

EXTRA: *Does this call provide a callback?*

- `call <mongoose>?**.aggregate [1, 1] 0:object ⇝ $callee([$args])`

  OBJ: *Is the receiver a mongoose model?*