Pretty-print ☐

```
{"version":3,"file":"fastdom.inline.fbeb22f8.bundle.min.js","mappings":"uHAAA,mBAAaA,aAOA,IAAIA
,EAAqD,WAAY,EAOjEC,EAAMC,EAAIC,uBACTD,EAAIE,6BACJF,EAAIG,0BACJH,EAAII,yBACJ,SAASC,GAAM,OAAOC,W
AAWD,EAAI,GAAK,EAO/C,SAASE,IACP,IAAIC,EAAOC,KACXD,EAAKE,MAAQ,GACbF,EAAKG,OAAS,GACdH,EAAKT,IAAM
A,EAAIa,KAAKZ,GACpBF,EAAM,cAAeU,EACvB,CA2HA,SAASK,EAAcC,GAChBA,EAAQC,YACXD,EAAQC,WAAY,EACpBD,E
AAQf,IAAIiB,EAAMJ,KAAK,KAAME,IAC7BhB,EAAM,mBAEV,CAWA,SAASkB,EAAMF,GACbhB,EAAM,SAEN,IAEImB,EAFA
N,EAASG,EAAQH,OACjBD,EAAQI,EAAQJ,MAGpB,IACEZ,EAAM,iBAAkBY,EAAMQ,QAC9BJ,EAAQK,SAAST,GACjBZ,EAAM
,kBAAmBa,EAAOO,QAChCJ,EAAQK,SAASR,EACnB,CAAE,MAAOS,GAAKH,EAAQG,CAAG,CAOzB,GALAN,EAAQC,WAAY,GAG
hBL,EAAMQ,QAAUP,EAAOO,SAAQL,EAAcC,GAE7CG,EAAO,CAET,GADAnB,EAAM,eAAgBmB,EAAMI,UACxBP,EAAQQ,MACP
,MAAML,EADQH,EAAQQ,MAAML,EAEnC,CACF,CASA,SAASM,EAAOC,EAAOC,GACrB,IAAIC,EAAQF,EAAMG,QAAQF,GAC1B
,SAAUC,KAAAWF,EAAMI,OAAOF,EAAO,EAC3C,CA9KAnB,EAAQsB,UAAY,CAClBC,YAAavB,EAUbY,SAAU,SAASY,GAEjB,I
AAIC,EAAM,IADVlC,EAAM,aACWkC,EAAOD,EAAME,SAASD,GACzC,EAUAE,QAAS,SAASC,EAAIC,GACpBtC,EAAM,WACN,
IAAIkC,EAAQI,EAAWD,EAAGvB,KAAKwB,GAAAbD,EAGlB,OAFA1B,KAAKC,MAAM2B,KAAKL,GAChBnB,EAAcJ,MACPuB,CA
CT,EAWAM,OAAQ,SAASH,EAAIC,GACnBtC,EAAM,UACN,IAAIkC,EAAQI,EAAWD,EAAGvB,KAAKwB,GAAAbD,EAGlB,OAFA1
B,KAAKE,OAAO0B,KAAKL,GACjBnB,EAAcJ,MACPuB,CACT,EASAO,MAAAO,SAASP,GAEd,OADAlC,EAAM,QAASkC,GACRT,
EAAOd,KAAKC,MAAOsB,IAAST,EAAOd,KAAKE,OAAQqB,EACzD,EAoCAQ,OAAQ,SAASC,GAEf,GADA3C,EAAM,SAAU2C,,GA
CI,iBAATA,EAAmB,MAAM,IAAIC,MAAM,mBAE9C,IAAIC,EAAQC,OAAOC,OAAOpC,MAO1B,OA6EJ,SAAeqC,EAAQC,GACrB
,IAAK,IAAIC,KAAAOD,EACVA,EAAOE,eAAeD,KAAMF,EAAOE,GAAAOD,EAAOC,GAEzD,CAvFIE,CAAMP,EAAOF,GACbE,EAA
M7B,QAAUL,KAGZkC,EAAMQ,YAAAYR,EAAMQ,aAErBR,CACT,EAKArB,MAAAO,MA+ET,IAAI8B,EAAUpD,EAAIc,QAAWd,EAA
Ic,SAAW,IAAIP,OAGwB,KAArC,aAAoB,OAAO6C,CAAAU,+BAGvE,CAnPD,CAmPsB,oBAAXC,OAAAyBA,YAAAwB,IAAR5C,KAA
sBA,KAAO6C,W","sources":
["webpack:///../../../node_modules/fastdom/fastdom.js"],"sourcesContent":["!(function(win)
{\n\n/**\n * FastDom\n *\n * Eliminates layout thrashing\n * by batching DOM read/write\n *
interactions.\n *\n * @author Wilson Page <wilsonpage@me.com>\n * @author Kornel Lesinski
<kornel.lesinski@ft.com>\n */\n\n'use strict';\n\n/**\n * Mini logger\n *\n * @return
{Function}\n */\nvar debug = 0 ? console.log.bind(console, '[fastdom]') : function()
{};\n\n/**\n * Normalized rAF\n *\n * @type {Function}\n */\nvar raf =
win.requestAnimationFrame\n  || win.webkitRequestAnimationFrame\n  ||
win.mozRequestAnimationFrame\n  || win.msRequestAnimationFrame\n  || function(cb) { return
setTimeout(cb, 16); };\n\n/**\n * Initialize a `FastDom`.\n *\n * @constructor\n */\nfunction
FastDom() {\n  var self = this;\n  self.reads = [];\n  self.writes = [];\n  self.raf =
raf.bind(win); // test hook\n  debug('initialized', self);\n}\n\nFastDom.prototype = {\n
constructor: FastDom,\n\n  /**\n   * We run this inside a try catch\n   * so that if any jobs
error, we\n   * are able to recover and continue\n   * to flush the batch until it's empty.\n
*\n   * @param {Array} tasks\n   */\n  runTasks: function(tasks) {\n    debug('run tasks');\n
var task; while (task = tasks.shift()) task();\n  },\n\n  /**\n   * Adds a job to the read
batch and\n   * schedules a new frame if need be.\n   *\n   * @param  {Function} fn\n   *
@param  {Object} ctx the context to be bound to `fn` (optional).\n   * @public\n   */\n
measure: function(fn, ctx) {\n    debug('measure');\n    var task = !ctx ? fn :
fn.bind(ctx);\n    this.reads.push(task);\n    scheduleFlush(this);\n    return task;\n
},\n\n  /**\n   * Adds a job to the\n   * write batch and schedules\n   * a new frame if need
be.\n   *\n   * @param  {Function} fn\n   * @param  {Object} ctx the context to be bound to
`fn` (optional).\n   * @public\n   */\n  mutate: function(fn, ctx) {\n    debug('mutate');\n
var task = !ctx ? fn : fn.bind(ctx);\n    this.writes.push(task);\n    scheduleFlush(this);\n
return task;\n  },\n\n  /**\n   * Clears a scheduled 'read' or 'write' task.\n   *\n   *
@param {Object} task\n   * @return {Boolean} success\n   * @public\n   */\n  clear:
function(task) {\n    debug('clear', task);\n    return remove(this.reads, task) ||
remove(this.writes, task);\n  },\n\n  /**\n   * Extend this FastDom with some\n   * custom
functionality.\n   *\n   * Because fastdom must *always* be a\n   * singleton, we're actually
extending\n   * the fastdom instance. This means tasks\n   * scheduled by an extension still
enter\n   * fastdom's global task queue.\n   *\n   * The 'super' instance can be accessed\n
* from `this.fastdom`.\n   *\n   * @example\n   *\n   * var myFastdom = fastdom.extend({\n   *
initialize: function() {\n   *     // runs on creation\n   *   },\n   *\n   *   // override a
method\n   *   measure: function(fn) {\n   *     // do extra stuff ...\n   *\n   *     // then
call the original\n   *     return this.fastdom.measure(fn);\n   *   },\n   *\n   *   ...\n
* });\n   *\n   * @param  {Object} props  properties to mixin\n   * @return {FastDom}\n   */\n
extend: function(props) {\n    debug('extend', props);\n    if (typeof props != 'object')
throw new Error('expected object');\n\n    var child = Object.create(this);\n    mixin(child,
props);\n    child.fastdom = this;\n\n    // run optional creation hook\n    if
(child.initialize) child.initialize();\n\n    return child;\n  },\n\n  // override this with a
function\n  // to prevent Errors in console\n  // when tasks throw\n  catch: null\n};\n\n/**\n
* Schedules a new read/write\n * batch if one isn't pending.\n *\n * @private\n */\nfunction
scheduleFlush(fastdom) {\n  if (!fastdom.scheduled) {\n    fastdom.scheduled = true;\n
fastdom.raf(flush.bind(null, fastdom));\n    debug('flush scheduled');\n  }\n}\n\n/**\n * Runs
```

Pretty-print

.catch` function has been defined\n * it is called instead.\n *\n * @private\n */\nfunction flush(fastdom) {\n debug('flush');\n\n var writes = fastdom.writes;\n var reads = fastdom.reads;\n var error;\n\n try {\n   debug('flushing reads', reads.length);\n   fastdom.runTasks(reads);\n   debug('flushing writes', writes.length);\n   fastdom.runTasks(writes);\n } catch (e) { error = e; }\n\n fastdom.scheduled = false;\n\n // If the batch errored we may still have tasks queued\n if (reads.length || writes.length) scheduleFlush(fastdom);\n\n if (error) {\n   debug('task errored', error.message);\n   if (fastdom.catch) fastdom.catch(error);\n   else throw error;\n }\n}\n\n/**\n * Remove an item from an Array.\n *\n * @param {Array} array\n * @param {*} item\n * @return {Boolean}\n */\nfunction remove(array, item) {\n var index = array.indexOf(item);\n return !!~index && !!array.splice(index, 1);\n}\n\n/**\n * Mixin own properties of source\n * object into the target.\n *\n * @param {Object} target\n * @param {Object} source\n */\nfunction mixin(target, source) {\n for (var key in source) {\n   if (source.hasOwnProperty(key)) target[key] = source[key];\n }\n}\n\n// There should never be more than\n// one instance of `FastDom` in an app\nvar exports = win.fastdom = (win.fastdom || new FastDom()); // jshint ignore:line\n\n// Expose to CJS & AMD\nif ((typeof define) == 'function') define(function() { return exports; });\nelse if ((typeof module) == 'object') module.exports = exports;\n\n})(typeof window !== 'undefined' ? window : typeof this != 'undefined' ? this : globalThis);\n"],"names": ["debug","raf","win","requestAnimationFrame","webkitRequestAnimationFrame","mozRequestAnimationFrame","msRequestAnimationFrame","cb","setTimeout","FastDom","self","this","reads","writes","bind","scheduleFlush","fastdom","scheduled","flush","error","length","runTasks","e","message","catch","remove","array","item","index","indexOf","splice","prototype","constructor","tasks","task","shift","measure","fn","ctx","push","mutate","clear","extend","props","Error","child","Object","create","target","source","key","hasOwnProperty","mixin","initialize","exports","window","globalThis"],"sourceRoot":""}