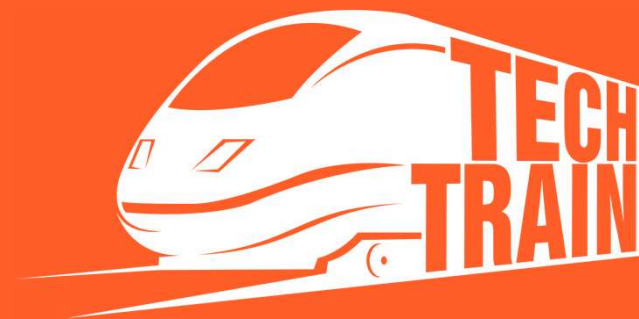
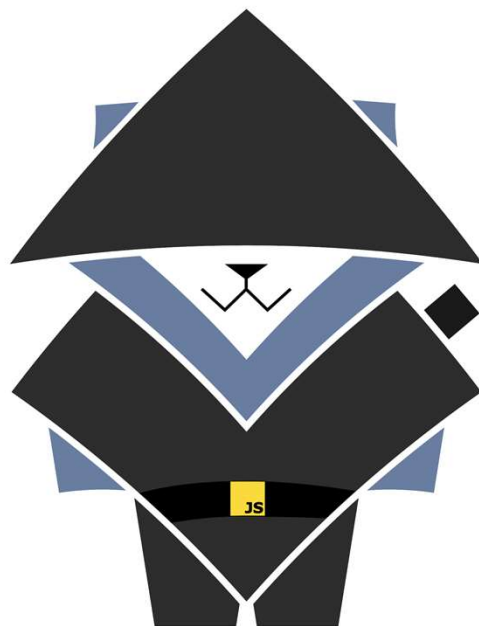




JavaScript для тех,
кого там нет

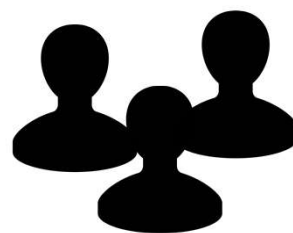
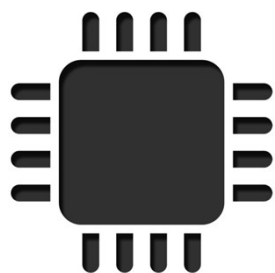




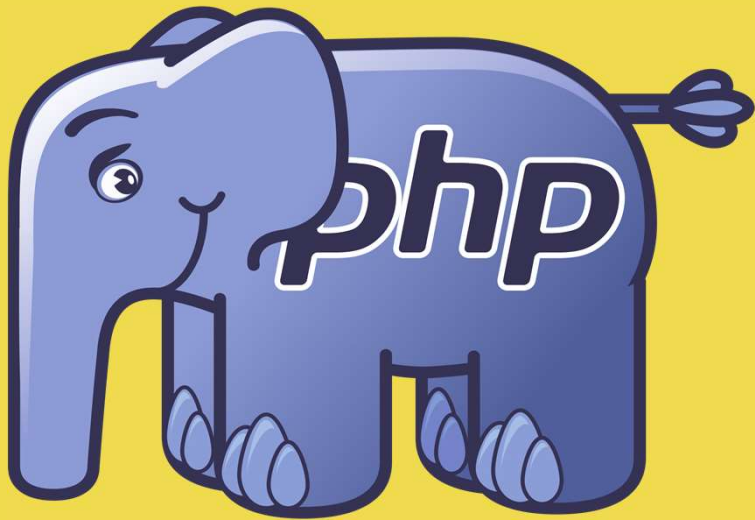
КТО я?



Что я делаю?



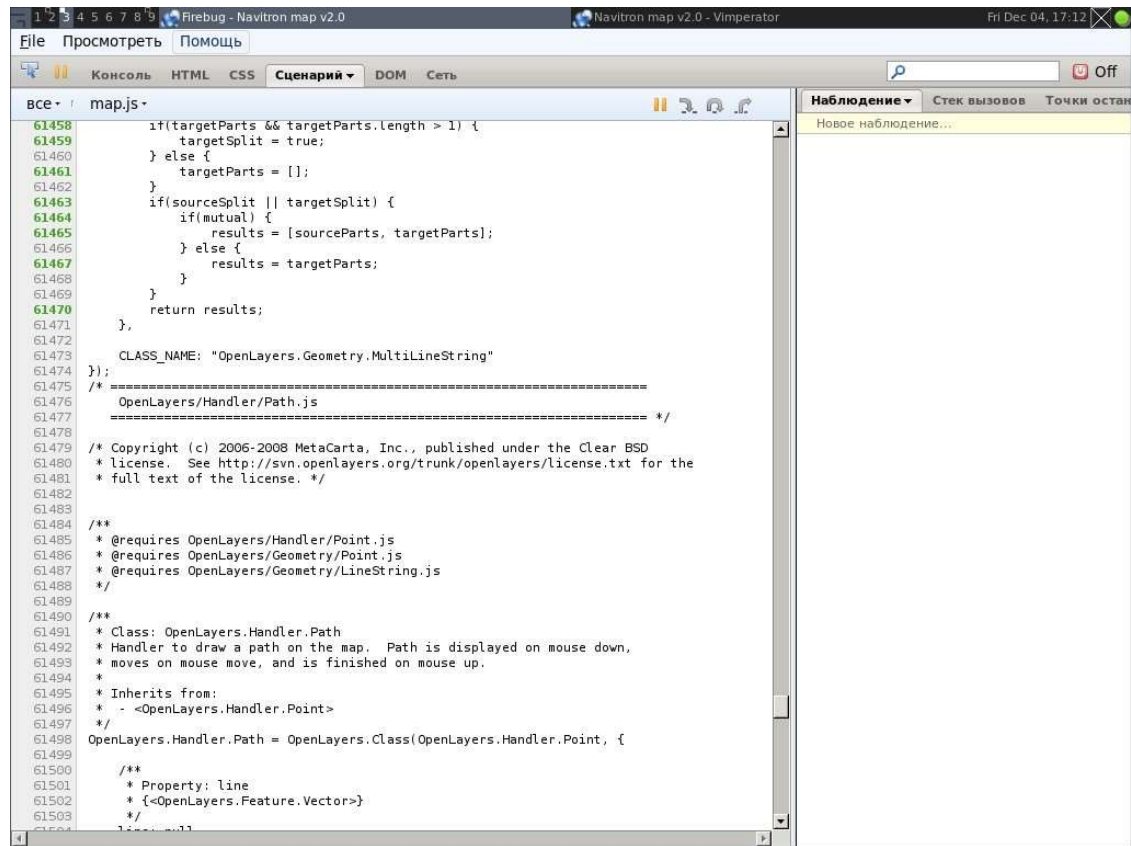
JS



JS



2008



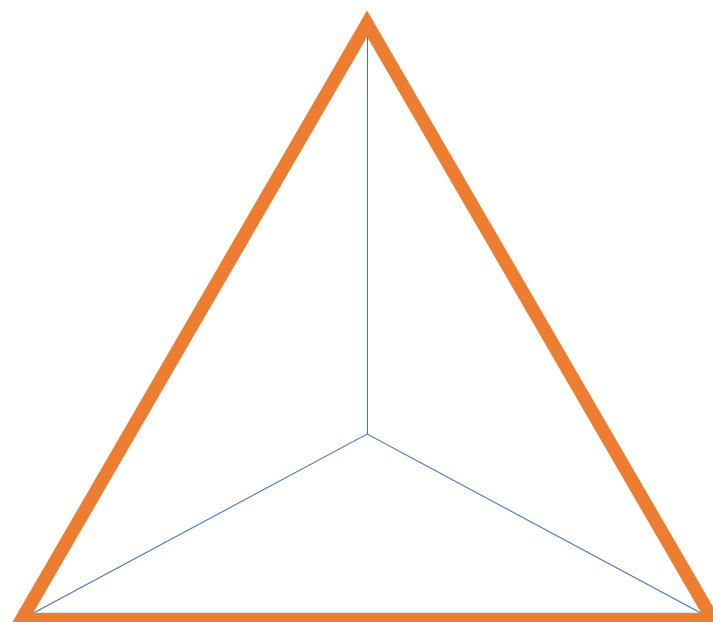
The screenshot shows a web browser window with the address bar displaying "Navitron map v2.0 - Vimperator". The Firebug console is open, showing the "Сценарий" (Script) tab. The code being executed is from "map.js" and includes a function for handling path drawing. The code is as follows:

```
61458     if(targetParts && targetParts.length > 1) {
61459         targetSplit = true;
61460     } else {
61461         targetParts = [];
61462     }
61463     if(sourceSplit || targetSplit) {
61464         if(mutual) {
61465             results = [sourceParts, targetParts];
61466         } else {
61467             results = targetParts;
61468         }
61469     }
61470     return results;
61471 },
61472
61473     CLASS_NAME: "OpenLayers.Geometry.MultiLineString"
61474 });
61475 /* =====
61476    OpenLayers/Handler/Path.js
61477    ===== */
61478
61479 /* Copyright (c) 2006-2008 MetaCarta, Inc., published under the Clear BSD
61480    * license. See http://svn.openlayers.org/trunk/openlayers/license.txt for the
61481    * full text of the license. */
61482
61483 /**
61484    * @requires OpenLayers/Handler/Point.js
61485    * @requires OpenLayers/Geometry/Point.js
61486    * @requires OpenLayers/Geometry/LineString.js
61487    */
61488
61489 /**
61490    * Class: OpenLayers.Handler.Path
61491    * Handler to draw a path on the map. Path is displayed on mouse down,
61492    * moves on mouse move, and is finished on mouse up.
61493    *
61494    * Inherits from:
61495    * - <OpenLayers.Handler.Point>
61496    */
61497 OpenLayers.Handler.Path = OpenLayers.Class(OpenLayers.Handler.Point, {
61498
61499     /**
61500      * Property: line
61501      * {<OpenLayers.Feature.Vector>}
61502      */
61503     line: null,
```


2008



Разработка



Поддержка

Сложность

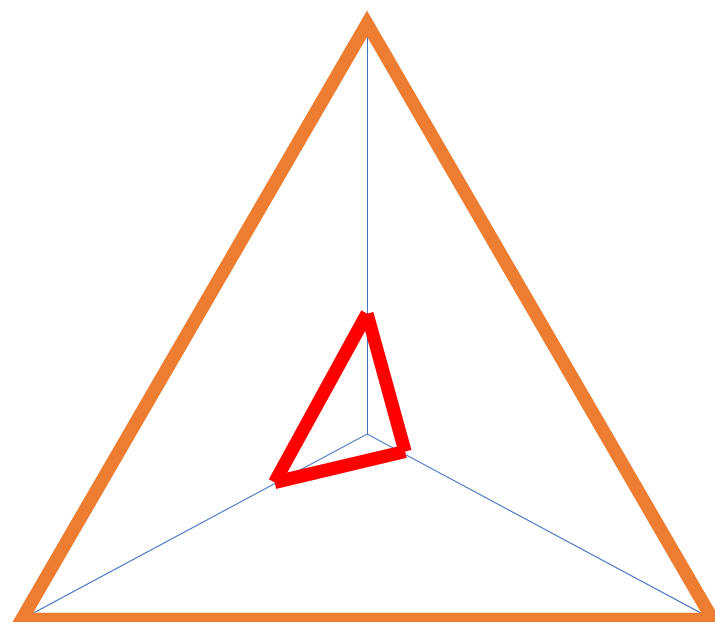
2008



Разработка

Поддержка

Сложность



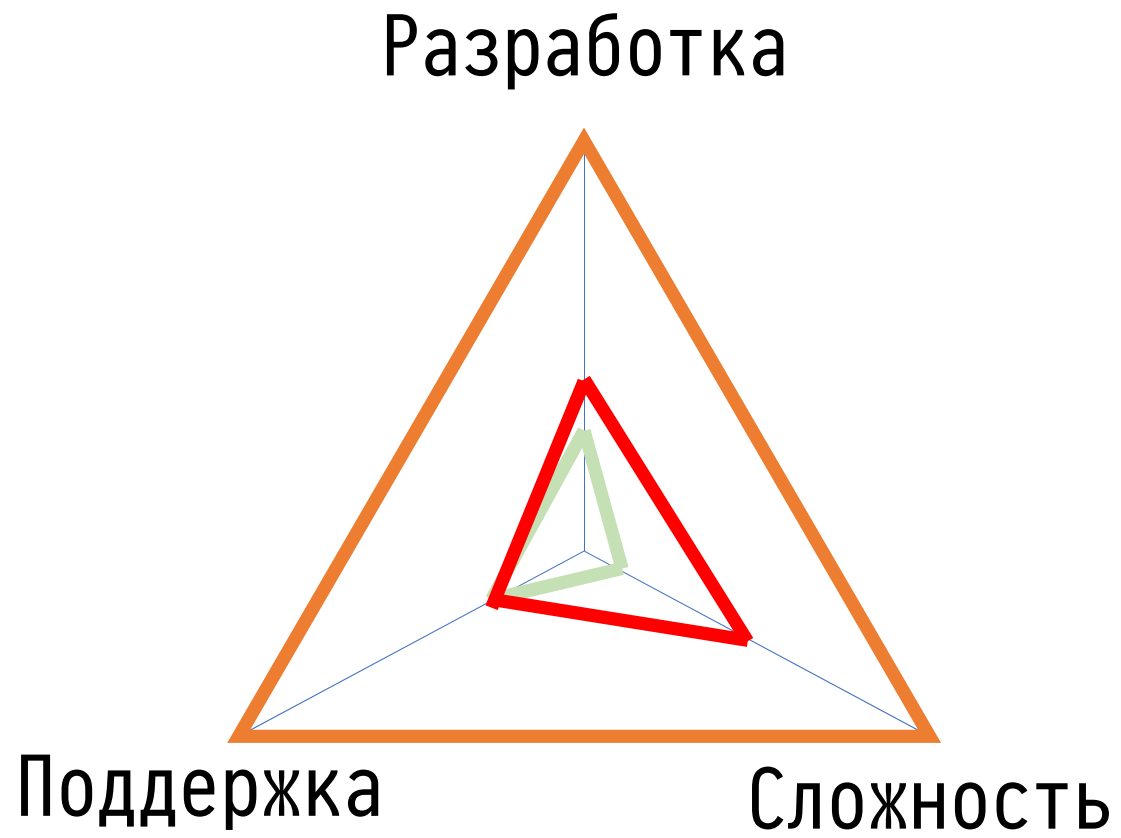
2008



2008



2009



2009

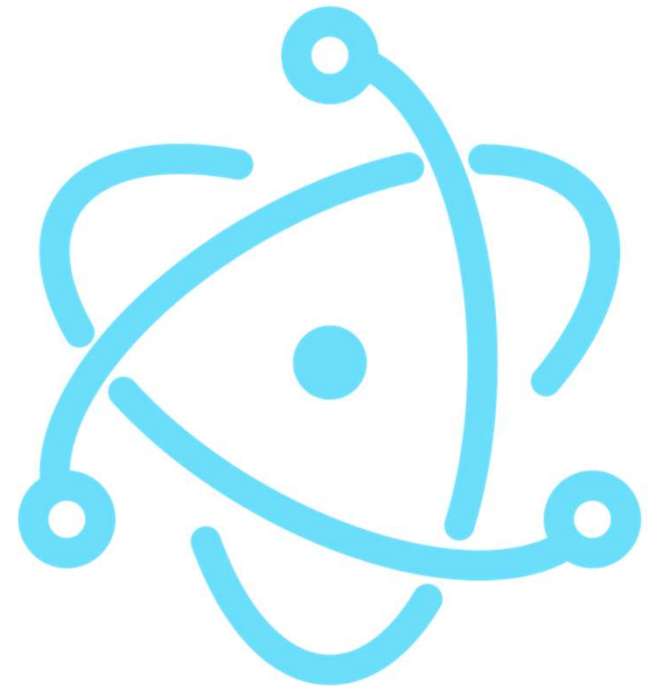


2012

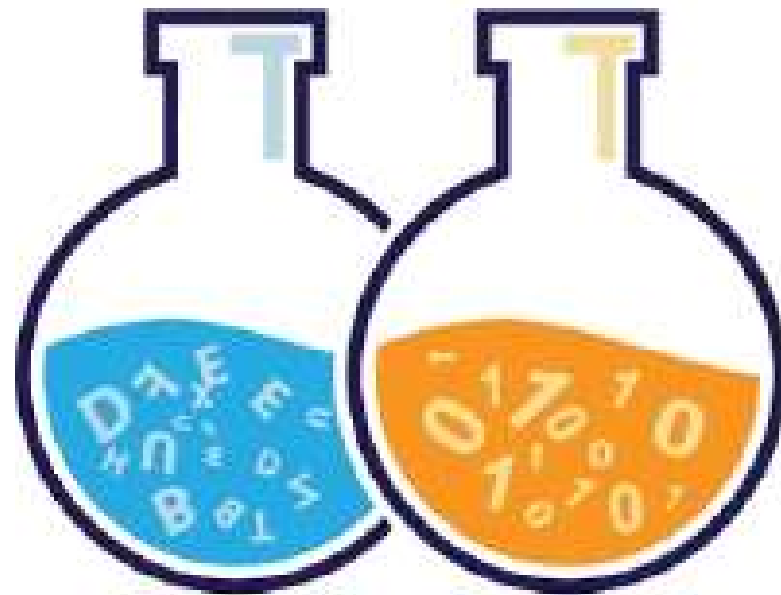


webpack

2013



2013



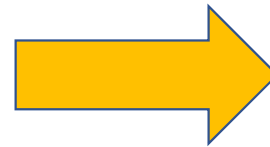
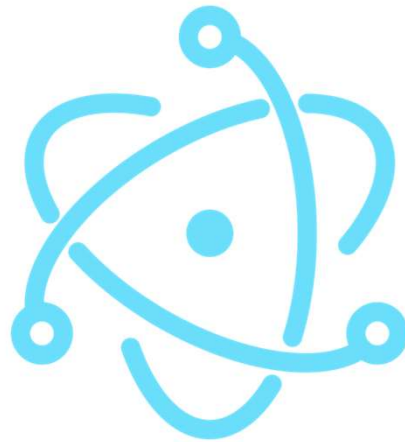
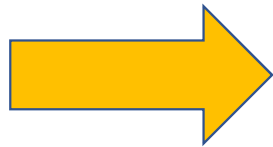
2014

6705

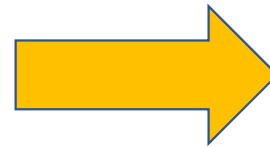
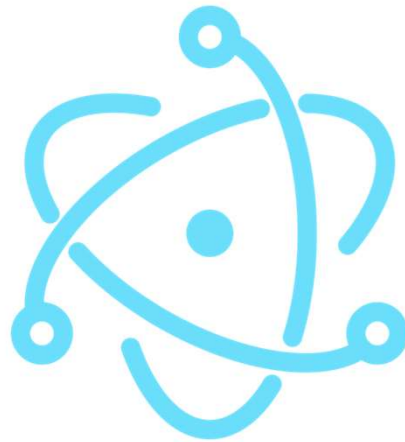
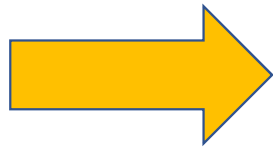
2014

BABEL

2015

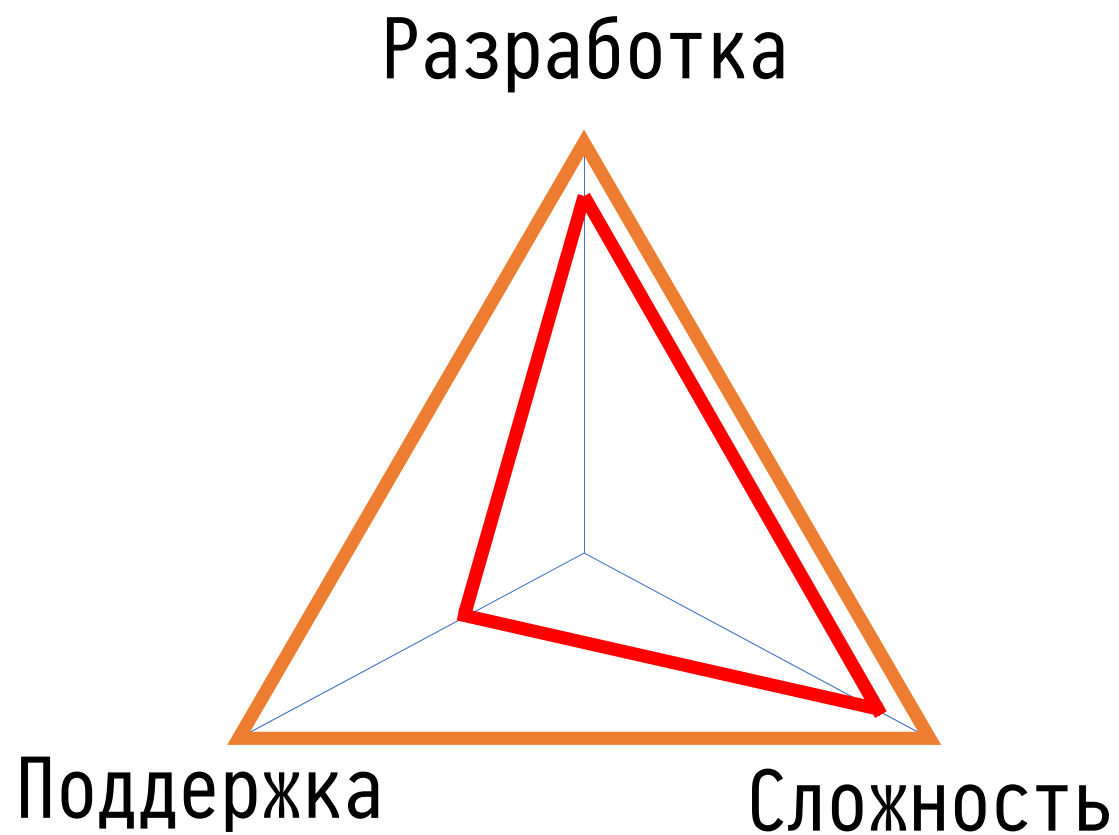


2015

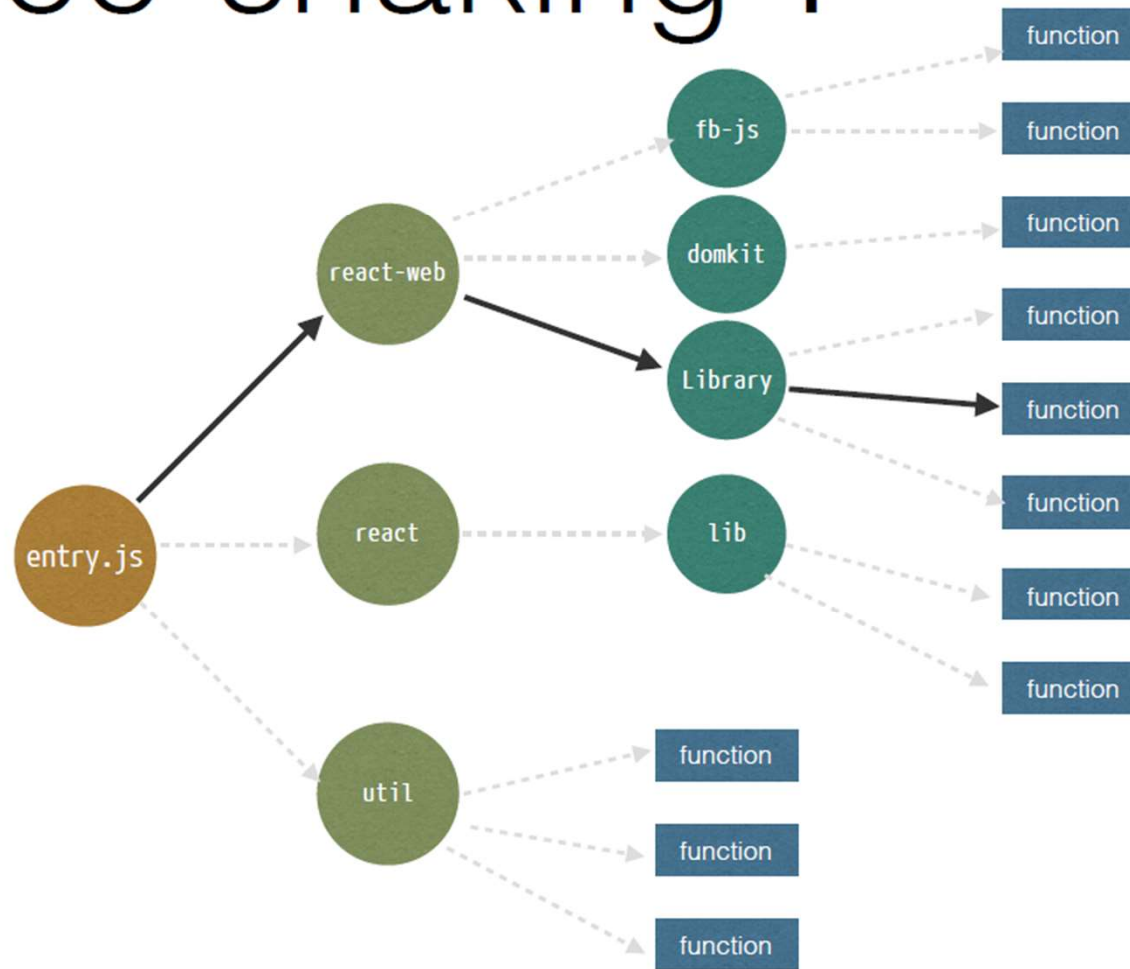


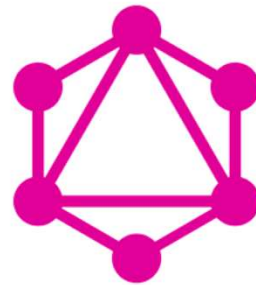
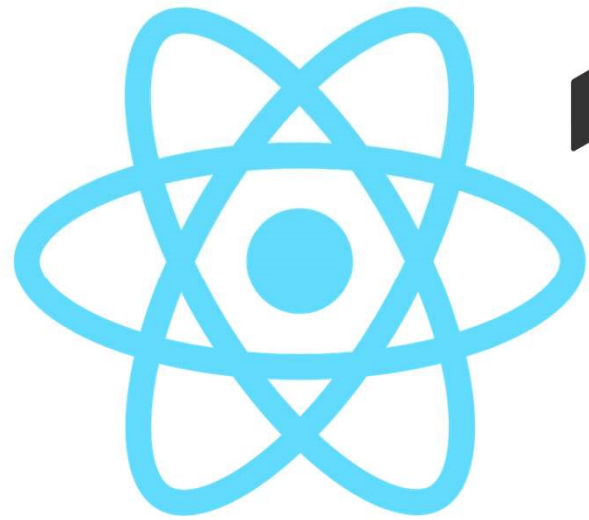
VS Code

2018

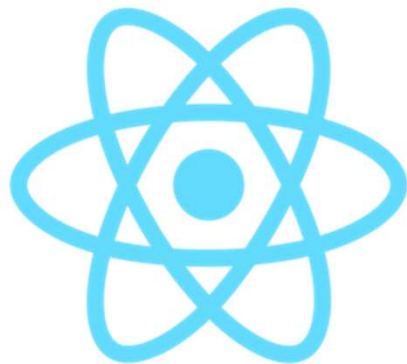


tree-shaking ?





GraphQL



React Native

2018+



WEBASSEMBLY

2018+

2019+

JS

2018+



Stage 3

	Proposal	Author	Champion	Tests
	<code>Function.prototype.toString</code> revision	Michael Ficarra	Michael Ficarra	✓
	<code>global</code>	Jordan Harband	Jordan Harband	✓
	<code>import()</code>	Domenic Denicola	Domenic Denicola	?
	Legacy RegExp features in JavaScript	Claude Pache	Mark Miller Claude Pache	✓
	<code>BigInt</code>	Daniel Ehrenberg	Daniel Ehrenberg	⚠
	<code>import.meta</code>	Domenic Denicola	Domenic Denicola	?
	Private instance methods and accessors	Daniel Ehrenberg	Daniel Ehrenberg Kevin Gibbons	?
	<code>Array.prototype.{flat,flatMap}</code>	Brian Terlson Michael Ficarra	Brian Terlson Michael Ficarra	✓
	Class Public Instance Fields & Private Instance Fields	Daniel Ehrenberg Kevin Gibbons	Daniel Ehrenberg Jeff Morrison Kevin Smith Kevin Gibbons	?
	Static class fields and private static methods	Daniel Ehrenberg Kevin Gibbons Jeff Morrison Kevin Smith	Shu-Yu Guo and Daniel Ehrenberg	
	<code>String.prototype.{trimStart,trimEnd}</code>	Sebastian Markbåge	Sebastian Markbåge	✓
	<code>String.prototype.matchAll</code>	Jordan Harband	Jordan Harband	
	<code>Symbol.prototype.description</code>	Michael Ficarra	Michael Ficarra	
	<code>Object.fromEntries</code>	Darien Maillet Valentine	Jordan Harband Kevin Gibbons	

2018+

2019+

JS

2018+



CoffeeScript



Dart





Вопросы?



javascript_ninja