JavaScript : que plébiscitent les développeurs en 2022 ?

Connaissance, intérêt, usage, satisfaction. Quatre métriques qui permettent d'en dire long sur l'état de JavaScript. En tout cas dans le cadre de l'étude référente State of JS, dont une nouvelle vague vient de paraître. Échantillon : 16 085 répondants, sondés entre le 13 janvier et le 2 février. Dont au moins 668 en France – un bon quart des participants n'ont pas précisé leur pays d'origine.

Combinées, ces métriques permettent de jauger la santé de l'écosystème sous autant d'angles. Ce à trois niveaux : fonctionnalités du langage, API web et bibliothèques.

Sur les deux premiers volets, le taux d'usage (nombre d'utilisateurs d'un élément divisé par le nombre d'utilisateurs en ayant connaissance) fait office d'indicateur de référence.

Sur la partie fonctionnalités, ressort le trio suivant :

– Chaînage optionnel (opérateur ?.) : 91,4 % de taux d'usage
– Coalescence des nuls (opérateur ??) : 77,9 %
– Importation dynamique : 57,7 %

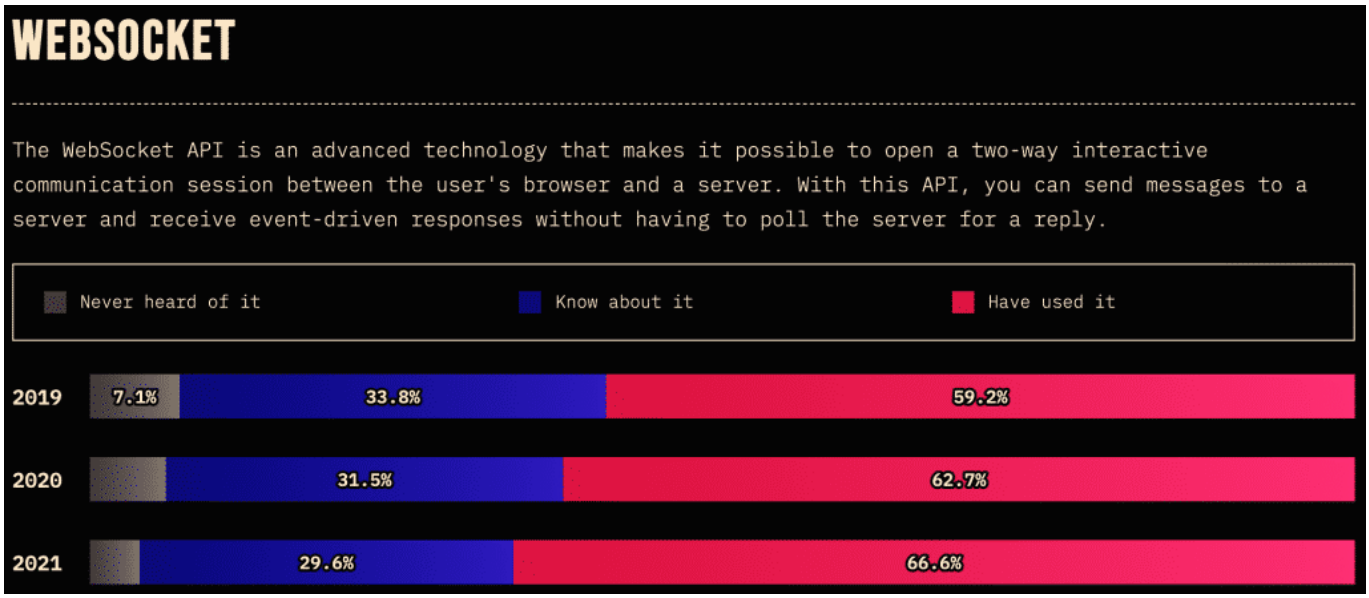
En bas du classement, il y a :

– Champs de classes privés : 29,5 %
– Méthode Promise.allSettled() : 29 %
– BigInt : 20,1 %

En valeur absolue, la moins connue des fonctionnalités du langage est l'affectation logique (opérateurs &&=, ||= et ??=) : 6456 répondants la connaissent.
Les moins utilisées sont les séparateurs numériques (2350), BigInt (2396) et la méthode Array.Prototype.at (2445).

Au niveau des API web, le podium du taux d'usage est le suivant :

– WebSocket : 69,3 %
– Intl : 56,1 %
– Shadow DOM : 55,8 %



En bas du classement :

– WebRTC : 24,5 %
– Web Speech : 16,8 %
– WebXR : 12,1 %

En valeur absolue, la moins connue des API est WebXR (2252 répondants). Les deux plus connues sont étroitement liées : WebSocket (13 329) et Service Worker (12 386). Suit WebGL (12 023).

À noter les taux d'usage des applications web progressives (64,4 %) et de WebAssembly (17,7 %).

Cinq bibliothèques JavaScript plébiscitées

Qu'en est-il de la progression des taux d'usage d'une année sur l'autre ? En se limitant à la part de l'échantillon ayant effectivement répondu à la question, la coalescence des nuls affiche la plus forte croissance (+21,8 points, à 67,7 %). Suivent le chaînage optionnel (+19, à 85,8 %) et l'importation dynamique (+6, à 48,9 %).

Sur la partie API web, les variations sont moins marquées. Elles avoisinent 4 points pour Service Workers comme pour WebGL, WebSocket, Shadow DOM et Custom Elements.

Pour les bibliothèques JavaScript, l'étude met en avant une autre métrique référente : la popularité. Deux dimensions entrent en compte : le nombre d'utilisateurs déclarés et leur satisfaction (entendent-ils continuer ou non à utiliser les bibliothèques ?).

Cinq bibliothèques ont à la fois un taux d'usage supérieur à 50 % et une appréciation globale positive : Webpack, Express, React, tsc CLI et Jest. Elles sont sur une trajectoire ascendante, comme Angular, Cypress, React Native, Puppeteer, Parcel, Svelte, Enzyme, Ionic, Electron, Nest, Expo et Testing Library.

Le paysage est différent si on s'en tient au taux de réponses « Je continuerai à utiliser ». Dans la tranche à plus de 90 %, on retrouve Cypress, Jest, tsc CLI et Testing Library. À leurs côtés, Vite,

esbuild et Next.js. Angular et Ionic sont, au contraire, dans le bas du classement (moins de 60 %). Avec Jasmine, Gatsby, Browserify, Gulp et Cordova.

Svelte, Tauri et Vite percent

Dans le détail sur la partie *front-end* :

– Solid fait une entrée directe au premier rang sur la métrique « satisfaction ».
– Svelte le devance sur la métrique « intérêt ».
– Le trio React-Angular-Vue.js domine sur les métriques « notoriété » et « usage ».

Sur la partie *back-end* :

– Next.js recule des trois rangs sur les critères « satisfaction » et « intérêt », que domine Sveltekit.
– Même top 5 sur les critères « usage » et « connaissance » : Express, Next.js, Gatsby, Nuxt et Nest.

Autant pour le *front-end* que le *back-end*, l'intérêt et la satisfaction apparaissent plus élevés pour de « jeunes » bibliothèques. On retrouve cette tendance sur la partie *testing* :

– Jest, Mocha et Cypress affichent les plus hauts taux de notoriété.
– Sur le critère « intérêt », la palme revient à Vitest ; sur le critère « satisfaction », à Testing Library.

Constat similaire – dans une moindre mesure – sur la partie mobile & *desktop* :

– Notoriété : React Native, devant Electron et Ionic
– Usage : Electron, devant React Native et Cordova
– Intérêt : Tauri, devant Electron et React Native
– Satisfaction : Tauri, devant Capacitor et Electron

Et pour les outils de *build* :

– Notoriété : Webpack, esbuild, SWC
– Usage : Webpack, tsc CLI, Gulp
– Intérêt : Vite, esbuild, SWC
– Satisfaction : Vite, esbuild, SWC

Que relèvent les auteurs de l'étude ? En particulier, que le taux d'usage de TypeScript est passé de 21 à 69 % depuis la première vague (2016). Ils soulignent que React et Vue.js dominent depuis cette même date. Mais que 2021 aura été l'année de Vite : plus de 30 % de taux d'usage et 98 % de satisfaction, après à peine an d'existence.

Illustration principale © Dmitry Baranovskiy – CC BY 2.0