前端框架的新選擇-Svelte

撰文｜叡揚資訊 前端設計部 前端設計師 初振言

提到前端工程，在開發者腦中浮現出來的東西，不外乎就是 React、Vue、Angular，這三大框架儼然佔據了現代前端開發的大部分江山，在 2019 年也有不少關於這些框架的話題，比方說 React 圈非常關心的 hook 終於發布正式版，Concurrent Mode 也被推到試用版本上了，Vue 這邊也即將推進一個版號，在寫法上有著不亞於 React Hook 的改變，還因為版本取名的關係，在社群上鬧出一些小風波，不過 2019 年也不只是只有這些老大哥的新聞， 在外國前端圈有一個話題一直維持在非常火熱的狀態，那就是 Svelte。

Svelte 是什麼？

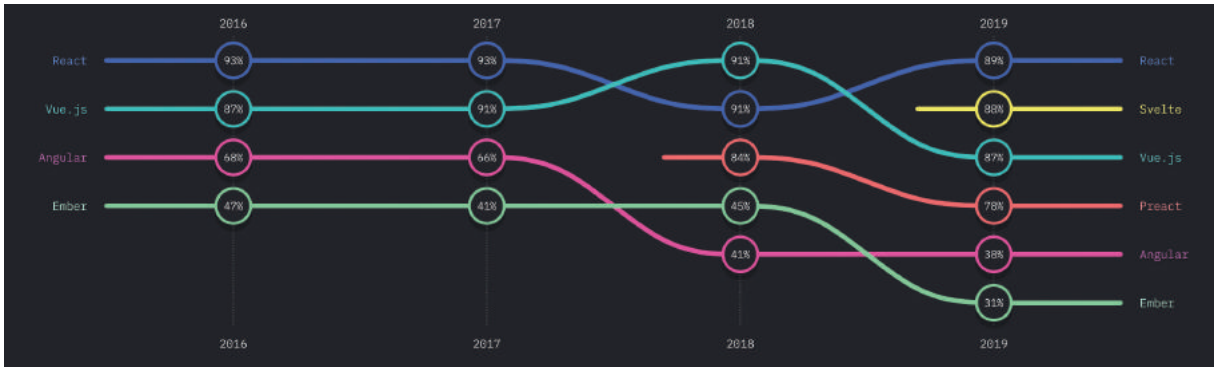
在今年 The State of JavaScript 2019 趨勢報告中，我們看到框架趨勢中有一個很有趣的現象。

在這個框架滿意度的調查中，有一個名叫 Svelte 的框架在 2019 年橫空出世，直接一舉拿下滿意度調查的第二名，看到這份成績單，我們在驚嘆過後，心中都充滿疑惑：『為什麼框架生態已經很成熟了，還要再出更多的框架呢，它們有什麼不同？』，在接下來，我將從現代框架的瓶頸來帶大家認識 Svelte 厲害的地方。

Virtual DOM 沒有你想像中的快

我們對於 Virtual DOM 高效的印象，來自於它高效的 diff 策略，但是使用過 React 開發的新手，肯定都有遇到過畫面嚴重卡頓的情況，因為在 diff 之前，必須先執行 render function（以下簡稱 render），才能產出對應的 Virtual DOM，之後才有辦法去做其高效的 diff，但是效能往往就在 diff 之前的這些 render 被不斷的

2019 Svelte 橫空出世



js-frmework-benchark 速度評比

Name Duration for...	vanillajs-keyed	svelte-v3.18.1-keyed	vue-v2.6.2-keyed	react-v16.8.6-keyed	angular-v8.2.14-keyed
create rows creating 1,000 rows	125.3 ± 4.0 (1.00)	138.1 ± 2.7 (1.10)	171.0 ± 4.3 (1.37)	173.7 ± 3.5 (1.39)	156.3 ± 5.9 (1.25)
replace all rows updating all 1,000 rows (5 warmup runs).	109.2 ± 1.7 (1.00)	135.6 ± 1.3 (1.24)	132.4 ± 1.0 (1.21)	135.9 ± 2.1 (1.24)	145.6 ± 10.4 (1.33)
select row highlighting a selected row. (5 warmup runs). 16x CPU slowdown.	20.3 ± 2.3 (1.00)	24.9 ± 2.3 (1.23)	105.6 ± 2.7 (5.20)	33.7 ± 3.3 (1.66)	28.1 ± 2.4 (1.38)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4x CPU slowdown.	55.2 ± 3.8 (1.00)	57.6 ± 3.2 (1.04)	71.3 ± 2.4 (1.29)	449.8 ± 5.7 (8.14)	452.3 ± 3.4 (8.19)
remove row removing one row. (5 warmup runs).	44.3 ± 1.9 (1.03)	45.5 ± 1.1 (1.06)	48.4 ± 0.8 (1.13)	42.8 ± 0.7 (1.00)	45.4 ± 1.4 (1.06)
clear rows clearing a table with 1,000 rows. 8x CPU slowdown	100.1 ± 3.2 (1.00)	148.5 ± 3.9 (1.48)	156.8 ± 3.6 (1.57)	149.1 ± 8.0 (1.49)	246.1 ± 3.5 (2.46)
slowdown geometric mean	1.01	1.18	1.64	1.81	1.91

觸發，按一下按鈕就跑了幾十次的 render 造成嚴重卡頓的情況屢見不鮮，白白浪費了許多的效能。

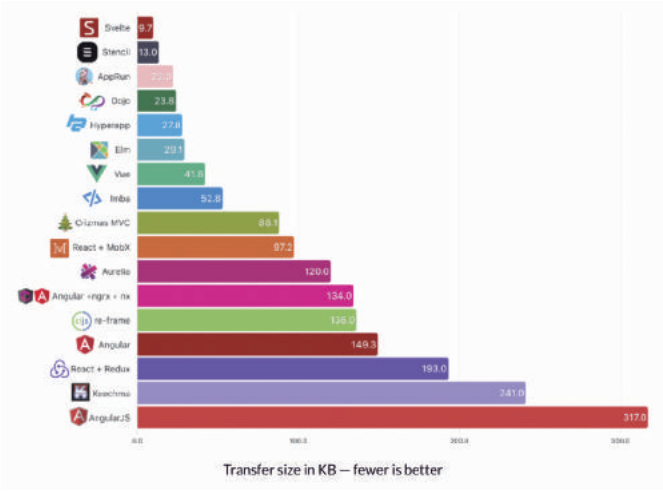
React為了解決這件事，將 memo、shouldComponentUpdate.....等抽象暴露出來，但是這和當初 React Core Team 中的一員講的：『 you could re-render your entire app on every single state change without worrying about performance. 』 明顯背道而馳，不然開發者應該不需要去考量 shouldComponentUpdate 這件事情才對，為了解決過度 re-render 的問題，這無疑的增加了程式架構的複雜度。

而 Svelte 對此的解決辦法相當的簡單粗暴，如果 Virtual DOM 會造成開發複雜度的提高，那不要使用 Virtual DOM 不就好了？Svelte 選擇在資料變化的時候，直接去改變原生 DOM，令人意外的是，在這個策略底下常見的 DOM 操作，效率比三大框架還快，甚至達到了接近原生的效能。

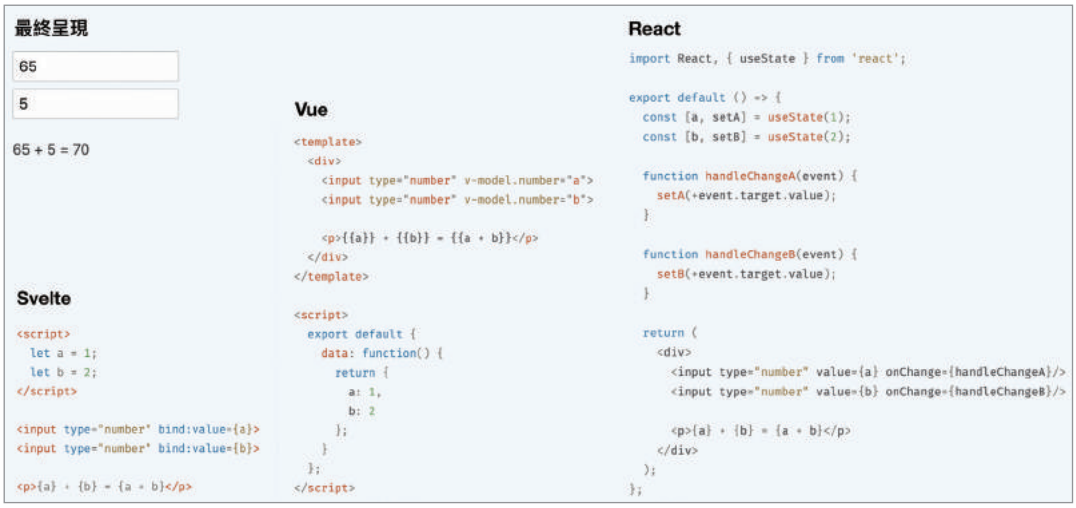
說是說框架
但其實它是編譯器

不管是 React、Vue 還是 Angular，在專案打包完畢後，裡面一定會包含框架本體，才有辦法在 Runtime 的時候，運行自家的優化策略，而 Svelte 的核心思想在於『透過靜態編譯減少 Runtime 程式碼』，在編譯階段解析這個專案用到了哪些 API，按專案的需求加載，而且 Svelte 並不需要 Virtual DOM，也大大的降低了 Runtime 所需要的程式碼，所

Real World 2019 容量評比



Svelte 擁有更好的可讀性



以打包的結果幾乎是 pure JavaScript，這使得 Svelte 在打包後的大小相當具有優勢。

更少的程式碼，更佳的可讀性，更好的學習曲線

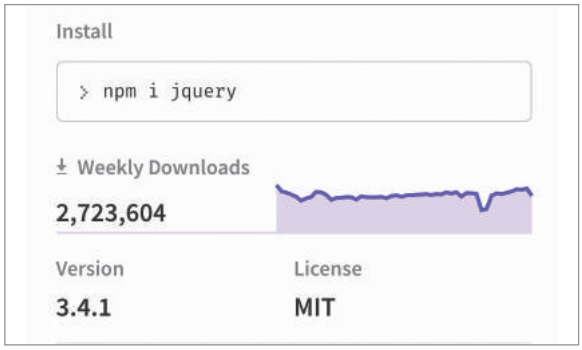
Svelte 希望給開發者一個更好的學習曲線，在堅信『沒有 API 就是最好的 API』精神下，經過數次改版，其 API 不斷被簡化，如上圖範例，官方非常自豪的說，如果要開發一個 component，使用 Svelte 與其它框架相比，有著更好的可讀性，並且差不多可以省下 40% 的程式碼，而更少的程式碼，帶來的是更少的 bug，以及更高效的開發。

Svelte 會擊敗現有的三大框架嘛？

這個大概是大家最關心的問題了，要學東西總是會想選一個最好的，但誰是最好的框架？這是一個沒有答案的問題，各家都有各家的擁護者，不然也不會有現在三大框架鼎立的局面，而是『 一框獨大 』了，沒有最好的框架，只有最適當的應用場景，這也是為什麼我們還是在很多的專案中，還是能見到 jQuery 的存在，每週的下載量甚至還達到了 2,700,000。

不過就算我們沒有辦法斷言誰是最好的框架，我們依然可以討論 Svelte 相關的生態圈會不會蓬勃發展。

2019 JQuery 每週下載量



商業考量

首先我們要了解一家新創公司在技術選型的時候，如何去選擇一個框架，是看執行效率？是看它的開發速度？其實都不盡然，就像前面說

的，一個專案在做技術選型的時候，首先考量的是它的業務場景：要不要開源？專案長期還是短期？複雜度高不高？團隊的技術能量如何等。

而三大框架都是被市場所驗證過的可行方案，不管是 Library 的數量、社群的活絡度甚至是公司徵才的難易度，其相關的生態圈都已經發展得非常成熟，所以現在開始一個新專案，公司沒有必要冒這個風險去使用一項新興技術，一個框架要能夠發展起來，單單引起工程師的興趣是不夠的，要有大公司當推手，其他小公司才會跟進，就業市場有需求，社群的生態才會蓬勃，React 有 Facebook、Angular 有 Google，而 Vue 有中國 13 億人口基數在那邊頂著，在還沒有一個更有力的推手出現之前，我認為 Svelte 還沒有辦法擊敗這些老前輩，如果你是要找到一份工作，Svelte 可能沒有辦法幫助你太多，至少 2020 還不行。

技術考量

如果你是一位熱愛新技術的開發者，Svelte 絕對是值得玩一下的技術，初出茅廬就拿滿意度第二名的開發體驗不是開玩笑的，而且除了上述內容提及的優點之外，有一個業務場景是它比眾前輩還要更優秀的，那就是開發 Web Component 。

不管在什麼領域，模組化都是很重要的一個概念，但是前端工程在程式上有分成 HTML、CSS、JS 三個區塊，彼此的相依性非常高，不同的模組又很容易互相影響，而且前端工程就是一個百家爭鳴的圈子，大家在各個面向的 solution 都不一樣，這造成模組跨專案的重用變得複雜，而 Web Component 就是為了解決這件事情而被提出的規範，主要的精神是把 HTML、CSS、JS 封裝成一個單獨的模組，而且幾乎隔絕模組內外的世界，讓不同模組之間不會互相影響，從而達到隨插隨用的地步。

但一般框架在打包完後，程式碼裡面一定會包含框架本體 ，如果你的專案裡面用了 10 個 Web Component ，其中 6 個是 Vue 寫的、4 個是 React 寫的，那麼你的專案中就會有 6 份 Vue 和 4 份 React 的本體，而且為了達到隨插隨用，這個基本上是無法避免的，但是就像前面說的，Svelte 它本身只是個編譯器，編譯出來的程式碼 ，幾乎是純粹的 JaveScript，所以說 Svelte 不只是跟前輩們搶大餅吃而已，他的出現也解決了一些現有的技術難點。

結語

Svelte 官方不只是提供一般的技術文件 ，它還擁有非常出色的互動式新手教學，上手難易度基本可以說完全無痛，而且官方有線上的 IDE 給大家試玩，有興趣的開發者可以去它官網試玩看看。

Svelte 2019 年才剛剛展露頭角，在開發大型專案的能力，還需要時間來驗證，但是它走上的是開源這條路，相信會有更多熱血的工程師們將它變得更成熟，而且它相關的生態系還非常小，也就是說可以開發的空間非常的大，現在加入才有機會成為大家口中的先驅者，才有機會成為該領域中的大神！