# Lodash

## Inledning

Lodash är ett bibliotek i JavaScript. Lodash är riktat mer mot problemlösning samt matematiska beräkningar medan bibliotek som react och jQuery fokuserar mer på det grafiska. Lodash hjälper utvecklaren att programmera funktioner vilket gör det mer användbart (D'amour, 2016).

* Vad är för och nackdelarna av att använda sig av Lodash.
* Stöds Lodash av de flesta webläsare.

## Bakgrund

Valde att fördjupa mig i bilbloteket Lodash, då matematik och problemlösning är ett av mina huvudintressen. För att få mer kunskap om Lodash så har jag tagit del av vad olika utvecklare anser om Lodash. Har hört på föreläsningarna att Lodash är ett enkelt ramverk och som student så har min nyfikenhet riktat sig till att för djupa mig i Lodash.

Frågan jag även har ställt mig är vilka är det som använder sig av Lodash? Det visade sig att flera kända applikationer så som Slack, Snapchat och NoRedInk gör detta (2).

# Resultat

Stepanov anser i sin artikel ”why you shouldn’t use Lodash anymore and use pure JavaScript instead” att prestandan kan disktueras. Han belyser i sin artikel att det 2 olika listorna skiljer sig åt tidsmässigt den som använder sig av lodash hastighet på 140ms medan det som använt sig av ren JavaScript hade 0ms. Vilket är en stor skillnad och gör Lodash till ett långsammare val av bibliotek (Stepanov, 2018).

Stepanov anser också att det viktigt att använda sig av en strukturerad och organiserad kod. Lodash är svårt att skriva ihop med olika funktioner, vilket han menar är ett problem för Lodash men han säger också att det fungerar väl om det gäller 1 eller 2 funktioner som ska vara sammankopplade (Stepanov, 2018).

I artikeln ”Higher-order functions in Lodash” beskrivs Lodash på ett annat sätt av författaren Piotrkowski som är av än annan åsikt Stepanov. Piotrkowski menar att Lodash har kraftfulla hjälpfunktioner vilket gör det till hans favoritbibliotek. Piotrkowski anser att Lodash är mer flexibel och mer nybörjar vänligt bibliotek. Han belyser att JavaScript saknar en riktigt Slice funktion, vilket är en fördel i Lodash. Detta är för att Lodash accepterar argument som en funktions parameter när en array parameter är tillgänglig. Han beskriver i sin sammanfattning att det är viktigt att förstå en del starka funktioner i Lodash speciellt som \_.partial och .\_curry om vill utnyttja Lodash bibliotek tillfullo (Piotrkowski, 2017).

Victor Olsson anser att när man använder sig av Lodash och dess integrerade funktioner kan det kan resultera till en betydligt mindre kod mängd, vilket i sin tur effektivisera kodskrivandet. Lodash är uppbyggt av komplexa algoritmer för att effektivisera prestandan på koden. Efter sig som Lodash har använt sig av både nya och gamla Javascriptstandarder har det gjorts så att Lodash stöds av de flesta webbläsare och även Internet Explorer 6 (Olsson, 2018).

## Diskussion

Utifrån de 3 artiklar som jag har tagit del av känns Lodash som ett nybörjar vänligt bibliotek (Olsson, Stepanov och Piotrkowski). Vilket gör biblioteket mer tillgängligt för fler användare. Som student tycker jag Lodash är ett användarvänligt bibliotek blir dock mer kritisk till min åsikt när jag tar del av Stepanovs (2018) artikel där han påpekar att uppladdningshastigheten försämrades när han använde Lodash jämfört med andra bibliotek så känns det som Lodash svaghet. Det finns en möjlighet att jag som student kommer att få samma åsikt som Stepanov när jag har fått fördjupade kunskaper om andra bibliotek.

## Framtid

Är övertygad om att Lodash kommer existera i framtiden då dess fördelar väger tyngre än dess nackdelarna av biblioteket. Biblioteket är så pass användarvänligt som gör det möjligt att det rikta sig till en stor målgrupp. Biblioteket förenklar processen att ta ut information från Arrayer vilket är en stor stryka för biblioteket.

## För och Nackdelar

Fördelar

* Användarvänligt.
* Ta ut information från Arrayer.
* Användarvänligt med matematisk problemlösning.

Nackdelar

* Långsamt bibliotek.
* Svårt att skriva ihop med olika funktioner.

## Kodexempel

I kodexemplet nedan visas 2 olika exempel på hur Lodash funktioner kan se ut. Jag har valt att visa \_.shuffle och även \_.random(1, 100). Då \_.shuffle ändrar om en ordning i en lista och \_.random generar ett tal då i vårt fall 1 till och med 100.

Källor:

D'amour, V. (2016). *10 Lodash functions everyone should know*. [online] Medium. Available at: https://medium.com/voobans-tech-stories/10-lodash-functions-everyone-should-know-334b372aec5d [Accessed 21 Feb. 2020].

D'amour, V. (2016). *10 Lodash functions everyone should know*. [online] Medium. Available at: https://medium.com/voobans-tech-stories/10-lodash-functions-everyone-should-know-334b372aec5d [Accessed 21 Feb. 2020].

2. <https://stackshare.io/lodash>

Stepanov, V. (2018). *Why You shouldn’t use lodash anymore and use pure JavaScript instead*. [online] Medium. Available at: https://codeburst.io/why-you-shouldnt-use-lodash-anymore-and-use-pure-javascript-instead-c397df51a66 [Accessed 21 Feb. 2020].

Piotrkowski, M. (2017). *Higher-order functions in Lodash*. [online] Medium. Available at: https://blog.pragmatists.com/higher-order-functions-in-lodash-3283b7625175 [Accessed 21 Feb. 2020].

Olsson, V. (2018). *Lodash Basics - AIL inom Informatik*. [online] AIL inom Informatik. Available at: http://media.hv.se/kurser/informatik-ail/old-vri401-vru401/lodash-basics/ [Accessed 23 Feb. 2020].

Vad behövs läggas till och vad är din åsikt på rapporten än så länge?