

Research: Sjakknettsider

Introduksjon

Det ser ut til å være en mangel og et behov i Norge for å kunne vise og få oppdatert og ny informasjon om turneringer som foregår "in-person". I dag brukes det hovedsakelig programvare Turneringsservice for å organisere turneringer og digitale sjakkbrett med programvare for å vise tilstanden på disse brettene. Vårt produkt vil prøve å løse dette.

Hva er vårt produkt

Vi vil utvikle en nettside hvor en besøkende kan finne en visning av tilstanden på digitale sjakkbrett, samt oversikt over pågående og kommende runder for pågående "in-person" sjakkturneringer. Dette produktet har som mål å fungere for alle som bruker programvare Turneringsservice og DGT digitale sjakkbrett. Denne kombinasjonen er først og fremst brukt i Norge i dag. Det skal vise raskt oppdatert stilling ved digitale sjakkbrett, vise raskt oppdatert resultatoversikt, og vise raskt oppdatert oversikt over kommende runde. Tjenesten skal være gratis, men det kan tenkes at den kan være reklamefinansiert. Det skal ikke være mulig å spille sjakkpartier i vårt produkt.

Formål

Vårt formål med denne utforskningen er å se hva som finnes i dag av nettsider, hva de tilbyr og om noen kan løse eller løser behovet for turneringsorganisasører i Norge.

Analyse

Vi vil gi en beskrivelse av sidene, sammen med grafisk struktur for sidene og hva som er utfordringene med sidene. En enkel styling-analyse og en liste over offentlige API'er er lagt ved under ressurser.

Hva finnes i dag

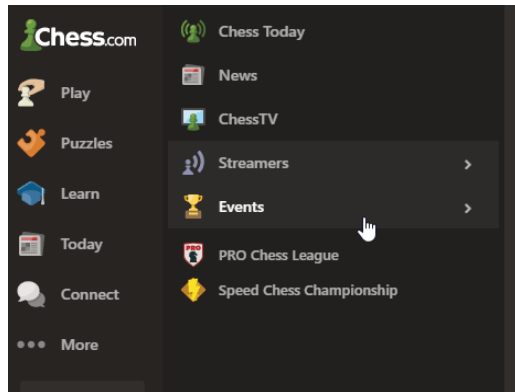
Chess.com:

En side som først og fremst er for å spille sjakk, enten mot andre spillere eller en datamaskin. En bruker kan spille i nett-turneringer, som blir arrangert på Chess.com. Har ressurser for å lære seg å spille sjakk, for å forbedre seg og utfordringer. Siden har også en oversikt over hendelser og turneringer som foregår i dag, samt som kjem i framtiden.

Turneringen og hendelsene ser ut til å være store hendelser, med profesjonelle spillere. Uvisst om man får se oppdaterte trekk. Er og uvisst om disse turneringene er online turneringer som foregår bare på chess.com

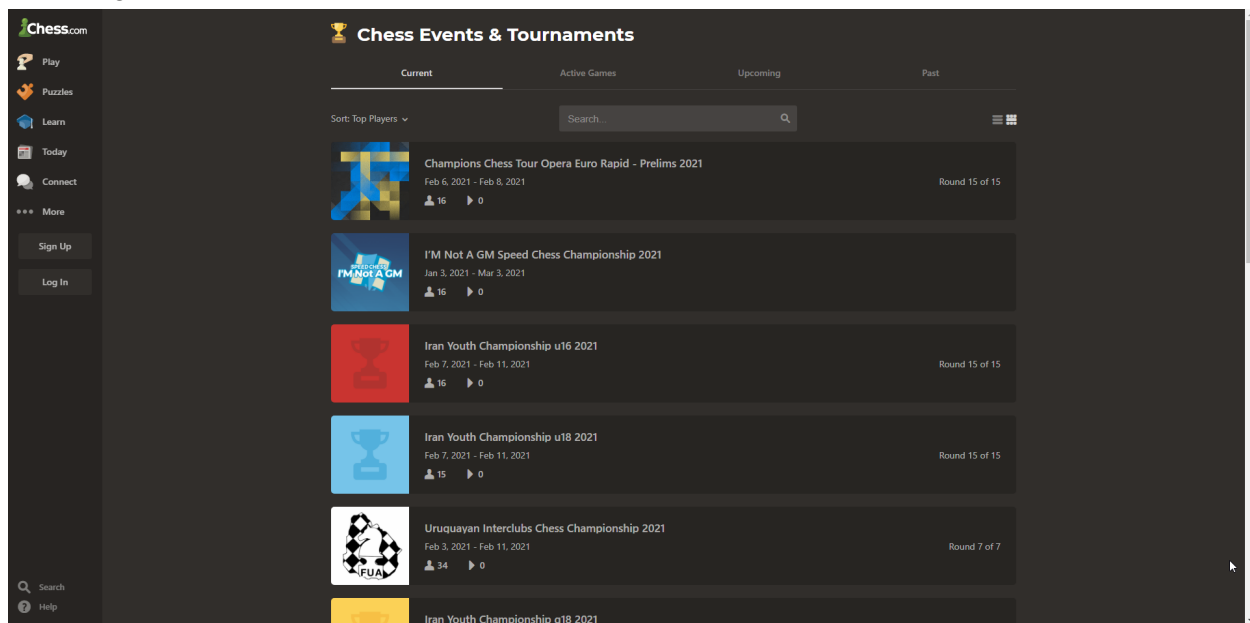
Struktur

Hovedside navigering til tuerneringsområde



Litt uoversiktlig og vanskelig å finne.

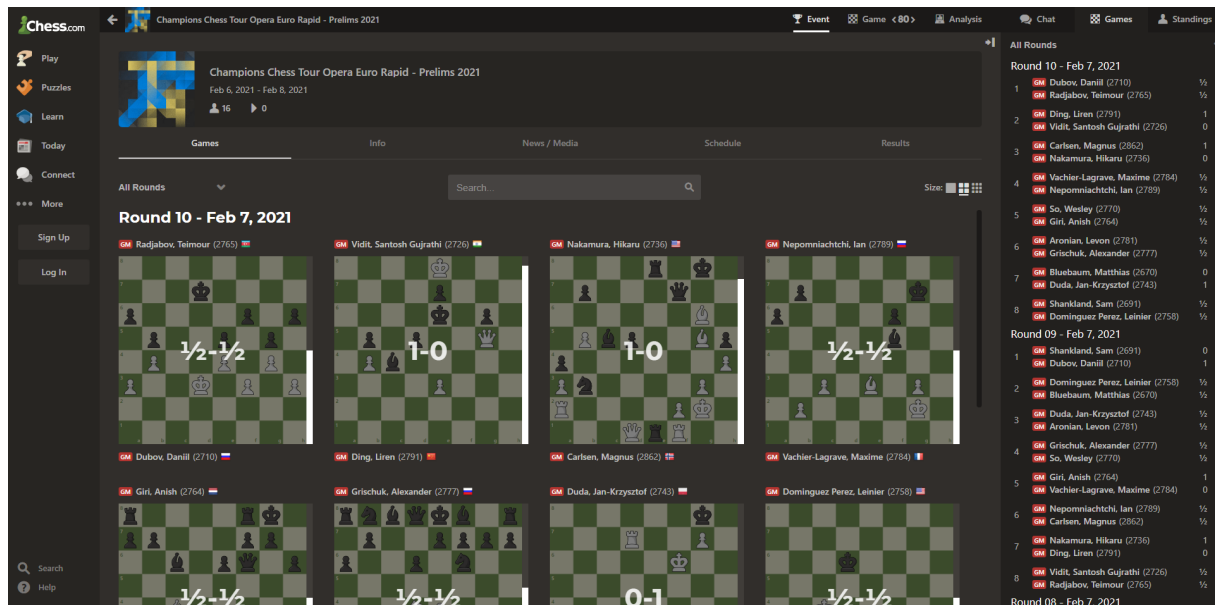
Turneringsoversikt



Viser

- Navnet til turneringen
- Datoer
- Runde nr
- Antall deltakere?

Oversikt innad i turnering



Midten:

- Flere faner hvor man kan se
 - Oversikt over kamper med
 - Resultat
 - Stilling på brettet
 - Sikkert livestilling om kampen er live
 - Hvem som spiller med
 - Tittel
 - Navn
 - Rating
 - Land
 - Info
 - Liste over deltakere
 - Land
 - Navn
 - Hvor de kvalifiserte seg fra
 - Rating
 - Rank i FIDE
 - Generell info om turneringen
 - Nyheter (tomt atm)
 - Tidsskjema med oversikt over når rundene starter
 - Dato
 - Hvilken runde
 - Tid i forskjellige tidssoner
 - Resultatoversikt
 - Rangert etter poeng
 - Viser poeng de fikk i hver runde

Høyre:

- Viser oversikt over resultatene fra hver kamp i hver runde
 - Runde #, dato.
 - Tittel, navn, rating og poeng oppnådd i runden

Innad i en kamp

The screenshot displays the Chess.com interface for a game titled "Champions Chess Tour Opera Euro Rapid - Prelims 2021". The central chessboard shows a game in progress with White to move. The move list on the right includes moves like 13. Bc2 Nc5, 14. O-O a4, 15. Be5 Nb3, etc. The right sidebar shows tournament rounds and player standings, including Round 10 - Feb 7, 2021, with players like Dubov, Daniil (2710) and Radjabov, Teimour (2765).

Venstre:

- Deltakere med
 - Navn
 - Tittel
 - Land
 - Rating
- Gjenstående tid
 - Farget basert på brikker
- Oversikt over hvilke brikker de har tatt fra motstanderen
- Analyse over hvor kampen lenes (hvit vs. svart)

Midten:

- Brettet

Høyre:

- Oversikt over trekk og hvordan det påvirker analysen av hvor kampen lener.
- Liste over resultatet av trekket

Utfordringer

- Om man velger å spille gratis, dette inkluderer både ordinære sjakkparti, situasjons-oppgaver og undervisning, så limiterer man seg til veldig få spill om dagen (ca. 3 oppgaver per dag og 1 videoundervisning per uke). Det reklameres ofte til at man burde skaffe seg "premium". Det er mange forskjellige premiumvarianter og for å ubegrenset tilgang til alle features på nettstedet, må man skaffe den dyreste varianten til 83 kr/mnd (999 kr/år).
- Foreløpig ser Chess.com ut til å være det vanskeligste nettstedet å bruke basert på relativt lite kluss på forsiden og en informativ sidemeny (Ser også renere/bedre ut enn de andre i min mening! :D).
- Kommer tilbake til denne om det dukker noe åpenbart opp.

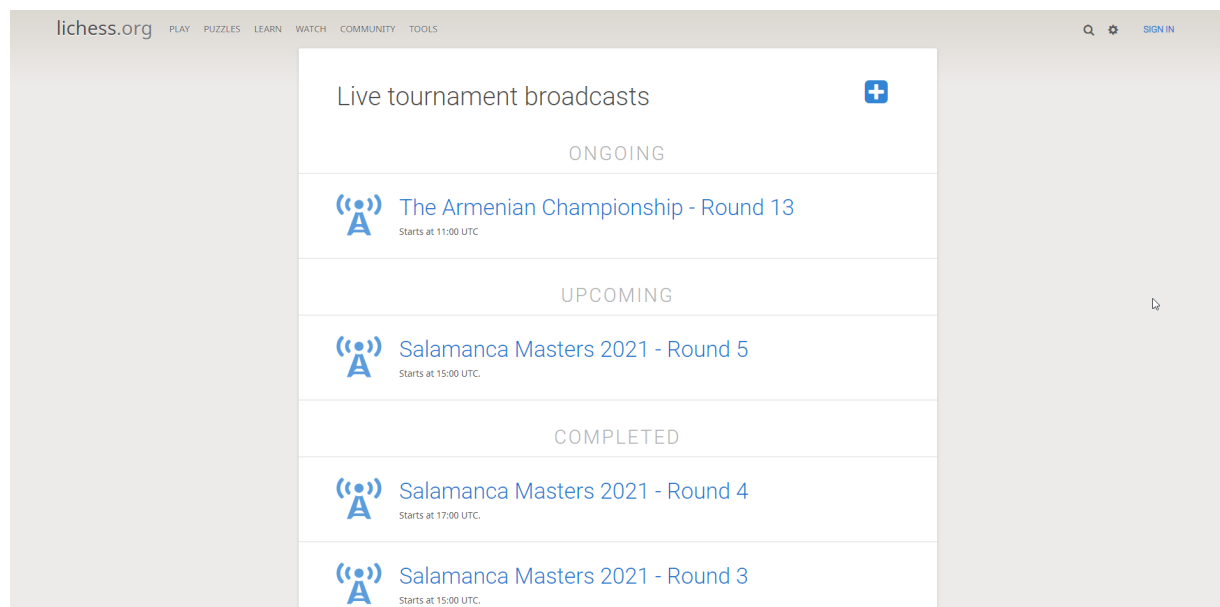
Lichess.org & Lichess Broadcast

Har samme funksjonalitet som Chess.com. Turneringer er berre turneringer som er organisert og spilt på lichess.org. Ingen "in-person" hvor mange får oppdatert informasjon. Sjakkbrettene ser ut til å være like sjakkbrettene produsert av Chessboard.js. Uvisst om Lichess bruker dette biblioteket eller om det er biblioteket som tar inspirasjon fra Lichess. Lichess har i tillegg mulighet til å laste opp PGN filer eller FEN-stringer for å analysere sjakkparti. Lichess ser helt klart ut til å være den mest åpne tjenesten, med mange API-ressurser, og ser ut til å bruke en del open source selv.

Lichess har en underside kalt Lichess Broadcast, hvor man viser stillingen og resultatene fra pågående og tidligere turneringer, samt kommende turneringer.

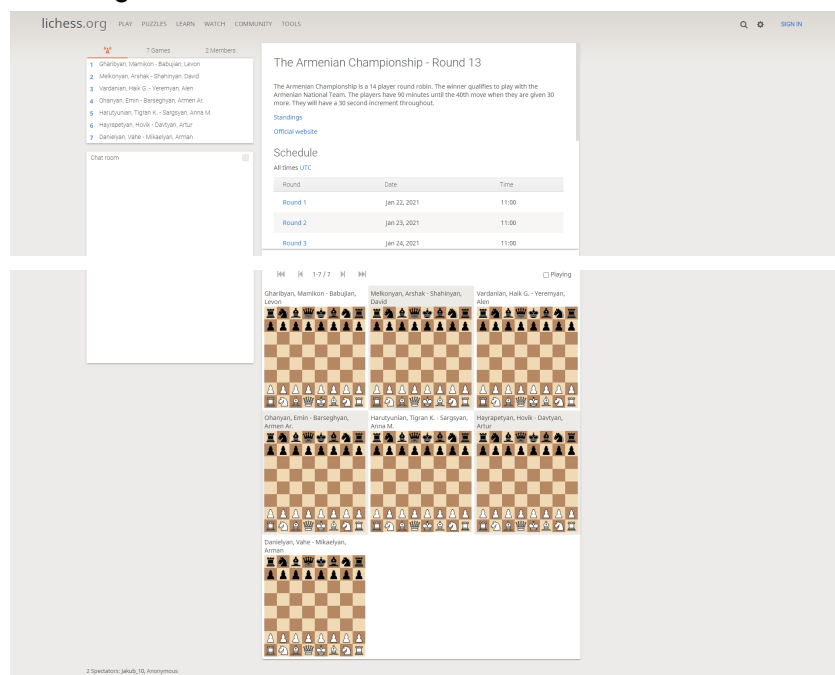
Struktur

Hovedside for turnering



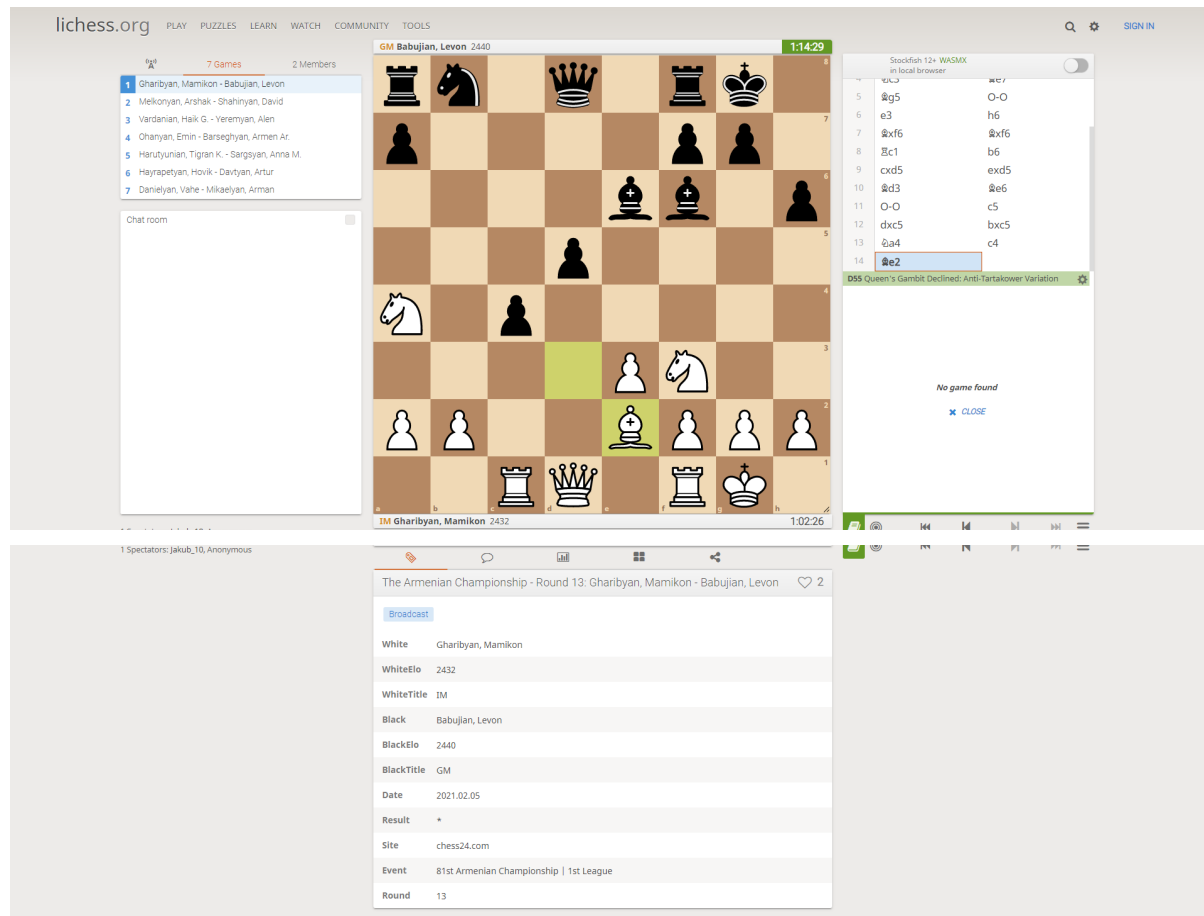
Liste over turneringer fordelt på pågående, kommende og tidligere
 Ny for hver runde
 Navn på turnering, runde og starttid

Turneringsoversikt



- Navn på turnering
- Beskrivelse
- Lenke til resultat
 - <https://i.imgur.com/uKRTgVr.png> - chess-results.com
- Tidsskjema (runde, dato og tid).
- Liste over kamper

Kampoversikt



- Venstre
 - Liste over kamper
- Midten
 - Visualisering av brettet
 - Over og under brettet inneholder informasjonen til spilleren med den fargen
 - Tittel (GM o.l.)
 - Navn
 - Rating
 - Gjenstående tid (farget grønn om det er gitt person sin tur)
- Høyre
- Liste over trekk gjort på gitt turn

Nedenfor er det en liste som inneholder litt mer info.

- Hvite brikker...
 - Navn
 - Rating
 - Tittel
- Sorte brikker
 - Navn

- Rating
- Tittel
- Dato
- Resultat
- Side
- Turnering
- Runde

Utfordringer

- Ingenting å nevne på kostnader, her er siden helt gratis å bruke med alle sine features.
- Kan være litt kluss med “raskt parti” seksjonen som tar opp meste av startskjermen. Bortsett fra det har man en relativt grei navbar å følge på toppen av siden og “quick buttons” for enkle spill mot AI eller mennesker.
- Siden ser relativt enkel/simpel ut, som ikke nødvendigvis er dårlig. Oppgave seksjonen som på lichess kalles “sjakknøtter” forklarer ikke hvorfor dårlige trekk er dårlige, slik som de gjør på chess.com (mest sannsynlig pga gratis adgang). Mangler “nyheter”.

Chess24.com

Igjen, samme som dei to tidligere tjenestene, men har flere muligheter til å følge med på turneringer i sanntid. Dette er forsøkt illustrert med bildene nedenfor. I tillegg har tjenesten en mulighet for chat med andre som følger en turnering. Da man undersøkte foregikk en turnering, Estonian Masters. Denne turneringen blir utført gjennom <https://home.tornelo.com/>. Tornelo er en online plattform for spilling av online turneringer, og er den offisielle turneringorganiserings plattformen for det europeiske sjakkforbundet. Dette bekrefter at Chess24.com støtter netturneringer. Om og hvordan dei følger med faktiske turneringer er uvisst.

◀ Volodin, Aleksandr - Kanep, Meelis ▶

Chessboard position for Volodin, Aleksandr (White) vs Kanep, Meelis (Black). The board shows a complex mid-game position. White pieces are on a1, b1, c1, d1, e1, f1, g1, h1, b2, c2, d2, e2, f2, g2, h2, b3, c3, d3, e3, f3, g3, h3, b4, c4, d4, e4, f4, g4, h4, b5, c5, d5, e5, f5, g5, h5, b6, c6, d6, e6, f6, g6, h6, b7, c7, d7, e7, f7, g7, h7, b8, c8, d8, e8, f8, g8, h8. Black pieces are on a8, b8, c8, d8, e8, f8, g8, h8, b7, c7, d7, e7, f7, g7, h7, b6, c6, d6, e6, f6, g6, h6, b5, c5, d5, e5, f5, g5, h5, b4, c4, d4, e4, f4, g4, h4, b3, c3, d3, e3, f3, g3, h3, b2, c2, d2, e2, f2, g2, h2, b1, c1, d1, e1, f1, g1, h1. A blue arrow points to the black king on g8.

Kanep, Meelis
GM 2465

Volodin, Aleksandr
GM 2378

Move Time: 00:12

Your Moves ?

2:43 min sec

2:32 min sec

Chessboard position for Narva, Mai (White) vs Ehlvest, Jaan (Black). The board shows a complex mid-game position. White pieces are on a1, b1, c1, d1, e1, f1, g1, h1, b2, c2, d2, e2, f2, g2, h2, b3, c3, d3, e3, f3, g3, h3, b4, c4, d4, e4, f4, g4, h4, b5, c5, d5, e5, f5, g5, h5, b6, c6, d6, e6, f6, g6, h6, b7, c7, d7, e7, f7, g7, h7, b8, c8, d8, e8, f8, g8, h8. Black pieces are on a8, b8, c8, d8, e8, f8, g8, h8, b7, c7, d7, e7, f7, g7, h7, b6, c6, d6, e6, f6, g6, h6, b5, c5, d5, e5, f5, g5, h5, b4, c4, d4, e4, f4, g4, h4, b3, c3, d3, e3, f3, g3, h3, b2, c2, d2, e2, f2, g2, h2, b1, c1, d1, e1, f1, g1, h1. A blue arrow points to the black king on g8.

Narva, Mai
WIM 2220

Ehlvest, Jaan
GM 2554

Move Time: 00:05

Your Moves ?

0:32 min sec

4:17 min sec

Struktur

Hovedside for liveturneringer

Live Tournaments | Live broadcasts from top events

Search tournament or player

81st Armenian Championship | 1st League

13 current round 6 games in play 85 finished games

The 1st League of the 81st Armenian Championship is a 14-player round-robin taking place in Yerevan from 22 January to 5 February 2021. The top 3 qualify for the main Championship and a chance to play on the Armenian team.

Standings

Rank	Name	Score	Rating
1	Babujian, Levon	8½/12	2440
2	Shahinyan, David	7½/12	2452
3	Gharibyan, Mam...	7½/12	2432
4	Harutyunian, Ti...	7½/12	2549
5	Davtyan, Artur	7/12	2466
6	Mikaelyan, Arman	7/12	2485

14 players Average rating: 2388

Games 6 running games only This round started at Feb 5, 2021 | 12:00

White	Evaluation	Black
1 IM Gharibyan, M. (7½) 2432		GM Babujian, L. (8½) 2440
2 Melkonyan, A. (4) 2079		IM Shahinyan, D. (7½) 2452
3 IM Vardanian, H.. (4) 2389	○ 1 - 0 ●	Yeremyan, A. (2½) 2274
4 FM Ohanyan, Emin (6) 2255		FM Barseghyan, (7) 2362
5 GM Harutyunian. (7½) 2549		WGM Sargsyan, A.. (5½) 2402
6 GM Hayrapetyan (6½) 2488		IM Davtyan, A. (7) 2466

Featured

- Feb 6-8 Opera Euro Rapld | Prelims
- Jan 16-31 Tata Steel Masters

Watch Live

- LIVE NOW** 81st Armenian Championship | 1st League Round 13
- today at 14:00 Estonian Masters Round 1.1
- today at 16:00 University of Salamanca Masters Round 5
- today at 20:00 Banter Blitz z Mateuszem
- tomorrow at 17:00 Opera Euro Rapld | Prelims Round 1
- tomorrow at 21:30 Banter Blitz with Lefong and Aye (170)
- Feb 7, 2021 | 22:00

Inneholder liste over turneringer nedover siden.

Øverst i turneringene sine rammer:

- Oversikt over antall runder (Nåværende runde)
- Kamper i spill for øyeblikket
- Tidligere spilte kamper

I dette tilfellet ser vi på eksempelet fra bilde ovenfor så ser vi at det nedenfor er to forskjellige tabeller.

Den til venstre er en resultatliste som inneholder informasjon om

- Plassering
- Spillernavn
- Score
- Rating

Til høyre for dette er nåværende spilte kamper, den inneholder følgende informasjon

- Spillernavn sorte brikker
- Spillernavn hvite brikker
- Evaluering av hvor kampen lener seg (svart vs. hvit)

The Eliteserien is a 10-team round-robin tournament taking place from 1 November 2019 to 8 March 2020 in Norway. Players receive 90 minutes for 40 moves, followed by 30 minutes to the end of the game, plus a 30-second increment starting from move one. Official website: www.sjakk.no/seriesjakken/eliteserien/elliteserien-2019-2020

Team		Result		Team		Team		Result		Team	
1 Alta		2 4		OSS		2 SOSS		½ 5½		Vålerenga	
FM Galchenko, M. 2418		○ ½-½ ●		GM Hammer, J. L. 2614		GM Lie, Kjetil A. 2523		○ ½-½ ●		IM Holm, K. S. 2484	
IM Mikalsen, E. 2367		● ½-½ ○		GM Agdestein, S. 2560		IM Ostmo, G. S. 2467		● 0-1 ○		GM Howell, D. W. 2653	
FM Klukin, K. 2293		○ 0-1 ●		IM Hauge, L. O. 2493		FM Byklum, B. 2338		○ 0-1 ●		GM Predojevic 2609	
FM Nielsen, A. 2213		● ½-½ ○		GM Johannessen 2437		FM Harestad, H. 2205		● 0-1 ○		IM Kaasen, T. F. 2382	
FM Solberg, J. 2220		○ 0-1 ●		IM Getz, Nicolai 2368		Astrup, K. C.. 2211		○ 0-1 ●		FM Hobber, A. 2357	
Mikalsen, H. 1988		● ½-½ ○		FM Ingebretsen 2255		Grebstad, K. 2065		● 0-1 ○		Fossum, O. B. 2291	
3 SK 96		2 4		Tromsø		4 Asker		4½ 1½		Bergens	
IM Bern, Ivar 2309		○ 1-0 ●		GM Notkevich, B. 2485		GM Hillarp Pers. 2536		○ ½-½ ●		IM Haldorsen, B. 2463	
FM Mork, K. A. 2259		● 0-1 ○		FM Vestby-Ellin. 2329		GM Ostenstad, B. 2448		● 1-0 ○		IM Houska, J. 2401	
FM Stigar, P. 2217		○ ½-½ ●		FM Lund, Gunnar 2289		GM Tisdall, J. 2412		○ 1-0 ●		IM Lovik, L. O. 2368	
Saevareid, O. 2136		● 0-1 ○		Royset, Pal 2149		IM Haubro, M. 2372		● ½-½ ○		IM Gullaksen, E. 2332	
Kvamme, J. A. 2096		○ ½-½ ●		Machlik, E. 2181		IM Sahl, Bjarke 2328		○ 1-0 ●		FM Gabrielsen 2304	
Moe-Nilssen 2124		● 0-1 ○		Machlik, M. 2094		FM Aulin-Jansson 2248		● ½-½ ○		FM Merkesvik, S. 2170	
5 SK 1911		3 3		Nordstrand							
GM Urkedal, F. 2528		○ ½-½ ●		GM Blomqvist, E. 2539							
FM Pantzar, M. 2406		● ½-½ ○		GM Salomon, J. 2507							
FM Risting, E. 2306		○ 1-0 ●		GM Djurhuus, R. 2460							
FM Storme, Isak 2284		● ½-½ ○		IM Haug, J. 2407							
FM Vitenberg, N. 2255		○ ½-½ ●		FM Tryggestad 2384							
Lindbol, A. 2208		● 0-1 ○		FM Nordquelle 2268							
<div>« < ... 3 4 5 6 7 8 9 > »</div>											

En annen visualisering er for de for det ikke er noen kamper i spill for øyeblikket. I eksempelet ovenfor er det oversikt over kampene til eliteserien. Her er det litt annerledes visualisert da vi ser mer en form av resultater fra gitt runde, og ikke totalt sett. Og turneringen er lagbasert, ikke individuelt basert

I disse tabellene ser vi følgende informasjon:

- Lagnavn
- Totalresultat samlet fra kampene
- Spillernavn m/ rating.
- Resultat fra individuelle kamper

Kamp Visualisering

81st Armenian Championship | 1st League
Round 13

Tournaments
81st Armenian Championship | 1st League x

◀
Melkonyan, Arshak - Shahinyan, David
▶

Shahinyan, David
IM 2452

Your Moves
?

1:11
hr min
1:14
hr min

Melkonyan, Arshak
2079
Move Time: 00:57

9... Nbd7

5 4. c3 b6 5. Bg2 Bb7 6. O-O Nf6 7. Nbd2 O-O 8. Re1 d5 9. Ne5 Nbd7

64crusher
its over for white
Jan 29

weisswurst
X
Jan 29

clockwisedragon
whites king to g4
come on anna, you can win this!
Feb 1
Feb 2

KwarentaDysenuyveOtsentaydos
game
Feb 2

BobbyLockwood
hello
Feb 2

SerkanGrlolu
fil f6
Feb 2

KwarentaDysenuyveOtsentaydos
haahahahaha
Feb 2

Raphael020
queen e2
queen f2
Feb 3
Feb 3

Register or login to chat

0.00 Equal
Stockfish 11 64 POPCNT | Cloud | Depth: 20 | Go Premium for the same deep analysis of any move

0.00 10. Nxd7 Nxd7 11. e4 cxd4 12. cxd4 e5 13. dxe5 Nxe5 14. exd5 Bxd5 15. Ne4 Qd7 16. Bh6 Rad8 17. Bxg7 Kxg7 18. Qd4 Qf5 19. f4 Bxe4 20. Qxe4 Qxe...

-0.41 10. Ndf3 Nxe5 11. Nxe5 Ne4 12. a4 a5 13. Ng4 h5 14. Ne5 Qc7 15. Nf3 e6 16. Bf4 Qe7 17. Ne5 Rfc8 18. h4 c4 19. f3 Nd6 20. e4 dxe4

-0.70 10. Nd3 e6

Her er selve visualiseringen for kampene som blir spilt.









































Informasjon:

- Brett med stilling
- Siste trekk visualisert med blå pil
- Gjenstående tid til høyre for brettet
- Spillernavn m/ rating og brukt tid på nåværende trekk på tilhørende side avhengig av spiller farge (svart eller hvit).
- Man kan nedenfor brettet gå frem og tilbake i blant trekkene samt se hva forrige var.
- Nedenfor der igjen ser man en oversikt over analyse av antatte mulige trekk
- En chat



De har en multiboard visning

- Spillernavn, rating og gjenstående tid vises for alle spillere.

Games and standings		Cross table		Multiboard		Analysis		Opening tree		Database		PGN		Information			
Rank	Player	1	2	3	4	5	6	7	8	9	10	11	12	13	14	R	P
1	 GM Babujian, Levon (2440)		0		0	1	1	1	½	1	1	½	1	½	1	12	8½
2	 IM Shahinyan, David (2452)	1		1	½	1	½	0	½	0	0	1		1	1	12	7½
3	 IM Gharibyan, Mamikon (2432)		0		1	½	½	1	1	½	0	1	1	0	1	12	7½
4	 GM Harutyunian, Tigran K. (2549)	1	½	0		0	1	1	0	½		½	1	1	1	12	7½
5	 IM Davtyan, Artur (2466)	0	0	½	1		½	½		½	1	1	1	1	0	12	7
6	 GM Mikaelyan, Arman (2485)	0	½	½	0	½		½	½	½	1	1	1		1	12	7
7	 FM Barseghyan, Armen Ar. (2362)	0	1	0	0	½	½		1		1	½	1	½	1	12	7
8	 GM Hayrapetyan, Hovik (2488)	½	½	0	1		½	0		½	0	1	½	1	1	12	6½
9	 FM Ohanyan, Emin (2255)	0	1	½	½	½	½		½		0	1	½	1	0	12	6
10	 WGM Sargsyan, Anna M. (2402)	0	1	1		0	0	0	1	1		0	½	0	1	12	5½
11	 IM Vardanian, Haik G. (2389)	½	0	0	½	0	0	½	0	0	1		½	1	1	13	5
12	 Melkonyan, Arshak (2079)	0		0	0	0	0	0	½	½	½	½		1	1	12	4
13	 IM Danielyan, Vahe (2357)	½	0	1	0	0		½	0	0	1	0	0		½	12	3½
14	 Yeremyan, Alen (2274)	0	0	0	0	1	0	0	0	1	0	0	0	½		13	2½

Krysstabell med oversikt over resultatene fra oppgjør på kryss av turneringen

Utfordringer

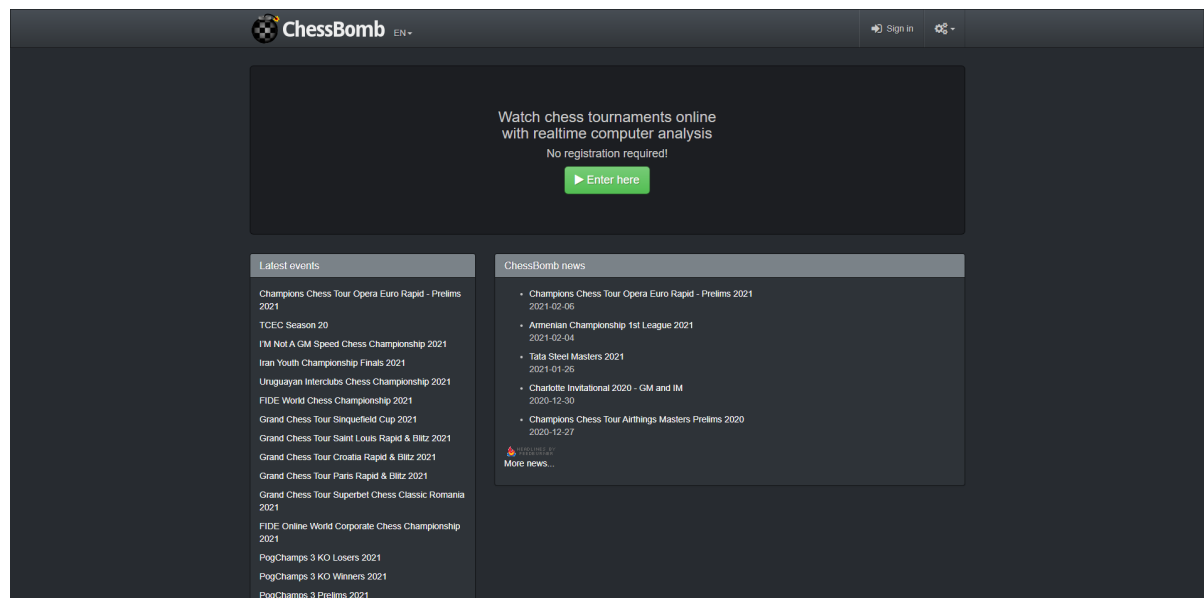
- Litt likt som Chess.com så har Chess24.com et premium abonnement. Dette ene abonnementet kan betales månedlig eller årlig og åpner også opp alle features på nettstedet. For 128 kr/mnd er dette det dyreste premium-abonnementet av de tre største sjakknettstedene, noe som kanskje ikke er så attraktivt for nye medlemmer.
- Chess24.com ser ikke umiddelbart vanskelig ut å bruke. Nettstedet har en veldig oversiktlig navbar, med forklarende tekst til hver underkategori. Dette er kanskje mer oversiktlig enn Chess.com og gjør det muligens lettere å bruke.
- Ser ut til å ha det meste som de andre sjakknettstedene har men "oppgaver/puzzles", samt det meste av undervisning/lesestoff er fullstendig gjemt bak premium-abonnementet og puzzles er kanskje ikke like nybegynnervennlig.

Chessbomb.com

Chessbomb.com er en ren turneringsoversiktside. Den tilbyr ikke tjenester for å spille partier selv.

Struktur

Hovedside



Liste over turneringer og nyheter

Turneringsoversikt

ChessBomb

EN

Sign in

Rooms

Name	Status
Champions Chess Tour Opera Euro Rapid - Prelims 2021	active
TCEC Season 20	active
IM Not A GM Speed Chess Championship 2021	active
Iran Youth Championship Finals 2021	active
Uruguayan Interclubs Chess Championship 2021	active
FIDE World Chess Championship 2021	upcoming
Grand Chess Tour Sinquefeld Cup 2021	upcoming
Grand Chess Tour Saint Louis Rapid & Blitz 2021	upcoming
Grand Chess Tour Croatia Rapid & Blitz 2021	upcoming
Grand Chess Tour Paris Rapid & Blitz 2021	upcoming
Grand Chess Tour Superbet Chess Classic Romania 2021	upcoming
FIDE Online World Corporate Chess Championship 2021	upcoming
PogChamps 3 KO Losers 2021	upcoming
PogChamps 3 KO Winners 2021	upcoming
PogChamps 3 Prelims 2021	upcoming
Dvoretsky Memorial 2021	upcoming
Armenian Championship 1st League 2021	finished
Magistral de Salamanca 2021	finished
Tata Steel Masters 2021	finished
Iran Youth Chessmech 2021	finished

© ChessBomb All rights reserved

[Contact](#)
[Premium](#)
[Blog](#)
[Facebook](#)
[Twitter](#)
[Privacy](#)

- Navn på turnering
- Status

Oversikt innad i turnering

ChessBomb

EN

Sign in

Champions Chess Tour Opera Euro Rapid Prelims '21

Carlsen - Nakamura

Room Info

Champions Chess Tour Opera Euro Rapid - Prelims 2021

2021-02-06 - 2021-02-08

Tweet

Like

Share

Embed

Official site

The Opera Euro Rapid is the 3rd stage of the Champions Chess Tour. 16 participants will play a single round robin and 8 of them will advance to a knockout event.

Time control is 15+10.

Top3 seeds:

Carlsen, Magnus

Vachier-Lagrave, Maxime

Ding, Liren

Chat

Commercial for every United of just soccer make a quarterback does love a woman and make her happy, and there were other things, I think

Wastrel787: I'm sure it was an old quote. It didn't mention fixing the plumbing.

dimat471: GM Norway commercial was great

dimat471: https://www.youtube.com/watch?v=yQy-a_E2I3s

Wastrel787: Yes I saw part of that but missed the punch line

Wastrel787: The one about getting knocked down and knocked up just facepalmed me

cafe8team: good night guys

cafe8team: want to sleep for 2 hours

cafe8team: at least

Only registered members can post.

Games

Round	White	Black	Result
10	Aronian, Levon	Grischuk, Alexander	½-½
	Blumberg, Matthias	Duda, Jan-Krzysztof	0-1
	Carlsen, Magnus	Nakamura, Hikaru	1-0
	Ding, Liren	Vidit, Santosh Gujathi	1-0
	Dubov, Daniil	Radjabov, Teimour	½-½
	Shankland, Sam	Dominguez Perez, Leinier	½-½
	So, Wesley	Giri, Anish	½-½
	Vachier-Lagrave, Maxime	Nepomniachtchi, Ian	½-½
09			
08			
07			
06			
05			
04			
03			
02			
01			

© ChessBomb All rights reserved

[Contact](#)
[Premium](#)
[Blog](#)
[Facebook](#)
[Twitter](#)
[Privacy](#)

- Listeoversikt over
 - Runde nr
 - Navn på spiller med hvite
 - Navn på spiller med sorte
 - Resultat

Innad i kamp



Venstre:

- Chat
- Analyse

Midten:

- Brettet
- Viser info om spiller over og under brettet avhengig av farge
 - Navn
 - Farge
 - Gjenstående tid
 - Resultat

Høyre:

- Liste over trekk gjort og resultatet av trekket

Utfordringer

Features - gjengående

Features som går igjen på alle nettstedene er:

- Online spill
 - Sjakkpartier mot bots, venner eller tilfeldige spillere
- Leksjoner
 - Artikler, e-bøker, videoer og interaktive spill-tutorials.
 - Analyse av spillpartier, samt mulighet til å få dine spilte sjakkparti analysert av AI
- Oppgaver
 - En feature som ligger litt i mellom leksjoner og spill. Kan være informativt og morsomt/interessant å holde på med. Kalles enten oppgaver eller "Puzzles" på de fleste nettsted og varierer med feedback til bruker.

- “Daglige nøtter” eller daglige puzzles.
- Forum
 - Brukes som alle andre forum, der folk har diskusjoner om forskjellige tema innenfor sjakk
- Livestreams
 - Kan være en relativt nylig populær feature, begrunnet pandemien som fører til at alt av sjakk foregår online (Turneringer, Twitch livestreams osv.).
- Nyheter
 - Tillater brukere å få “sjakknyheter” i form av nyhetsartikler lest på nettstedet de også spiller sjakk. Lichess har ikke en slik feature som jeg kan finne.
- Turneringsoversikt
 - Oversikt over hvilke turneringer som kommer opp og informasjon rundt de.

Drøfting

Egenskaper

De aller meste av funksjoner og egenskaper disse sidene tilbyr er overflødig for oss. Samtidig er det ingen av sidene som oppfyller behovet til organisatorer av norsk amatørsjakk. Ingen av sidene har ingen enkel mulighet til å kunne vise hva som foregår på “in-person” turneringer i norske sjakkmiljøer.

Struktur

Det er grafiske elementer og en potensiell sidestruktur vi kan se for oss å ta med oss videre.

Fremside

Vårt produkt er et lettvektsprodukt og vil ikke tilby mye av de egenskapene sidene tar for seg. Vi vil ha en enkel oversikt over pågående, oppkommende og kanskje ferdigspilte turneringer. Vi fant oversikt fra Lichess Broadcast til å være en mulighet.

All Rounds		
Round 11 - Feb 8, 2021		
1	GM Dominguez Perez, Leinier (2758)	-
	GM Dubov, Daniil (2710)	-
2	GM Duda, Jan-Krzysztof (2743)	-
	GM Shankland, Sam (2691)	-
3	GM Grischuk, Alexander (2777)	-
	GM Bluebaum, Matthias (2670)	-
4	GM Giri, Anish (2764)	-
	GM Aronian, Levon (2781)	-
5	GM Nepomniachtchi, Ian (2789)	-
	GM So, Wesley (2770)	-
6	GM Nakamura, Hikaru (2736)	-
	GM Vachier-Lagrave, Maxime (2784)	-
7	GM Vidit, Santosh Gujrathi (2726)	-
	GM Carlsen, Magnus (2862)	-
8	GM Radjabov, Teimour (2765)	-
	GM Ding, Liren (2791)	-
Round 10 - Feb 7, 2021		
1	GM Dubov, Daniil (2710)	½
	GM Radjabov, Teimour (2765)	½
2	GM Ding, Liren (2791)	1
	GM Vidit, Santosh Gujrathi (2726)	0
3	GM Carlsen, Magnus (2862)	1
	GM Nakamura, Hikaru (2736)	0
4	GM Vachier-Lagrave, Maxime (2784)	½
	GM Nepomniachtchi, Ian (2789)	½
5	GM So, Wesley (2770)	½
	GM Giri, Anish (2764)	½
6	GM Aronian, Levon (2781)	½
	GM Grischuk, Alexander (2777)	½
7	GM Bluebaum, Matthias (2670)	0
	GM Duda, Jan-Krzysztof (2743)	1
8	GM Shankland, Sam (2691)	½
	GM Dominguez Perez, Leinier (2758)	½

Konklusjon

Vi analyserte sidene Chess.com, Lichess.org, Chess24.com og Chessbomb.com. Alle utenom Chessbomb gir mulighet for å spille partier. Dette er mer omfattende produkter enn det vi skal tilby. Alle sidene tilbyr en form for turneringsoversikt for profesjonelle turneringer, hvor man kan følge med på stilling i partiet og stillingen i turneringer. Disse har så å si samme utforming. Dette er noe som mangler i norsk amatørsjakk. Vårt produkt skal prøve å tilby dette for norsk amatørsjakk. Vi har funnet en mulig noen sidestrukturelementer vi kan ta med oss videre.

Ressurser

API'er

Chess.com:

<https://www.chess.com/club/chess-com-developer-community>

Lichess.org:

<https://lichess.org/api>

Chess24.com

<https://github.com/chess24com>

Styling

Chess.com

Chess.com tilbyr mange andre brett kombinasjoner enn denne. Nedenfor viser default. Ser ut til å ha den mest uvanlige default fargekombinasjonen. Brikkene er mer unike da dei har skygger. Etter litt snoking ser det ut som brettet og brikkene er en SVG som blir tegnet.





Farger - Default

Brikker

Brikker	HEX	Eksempel
Sort	#565352	
Hvit	#f8f8f8	

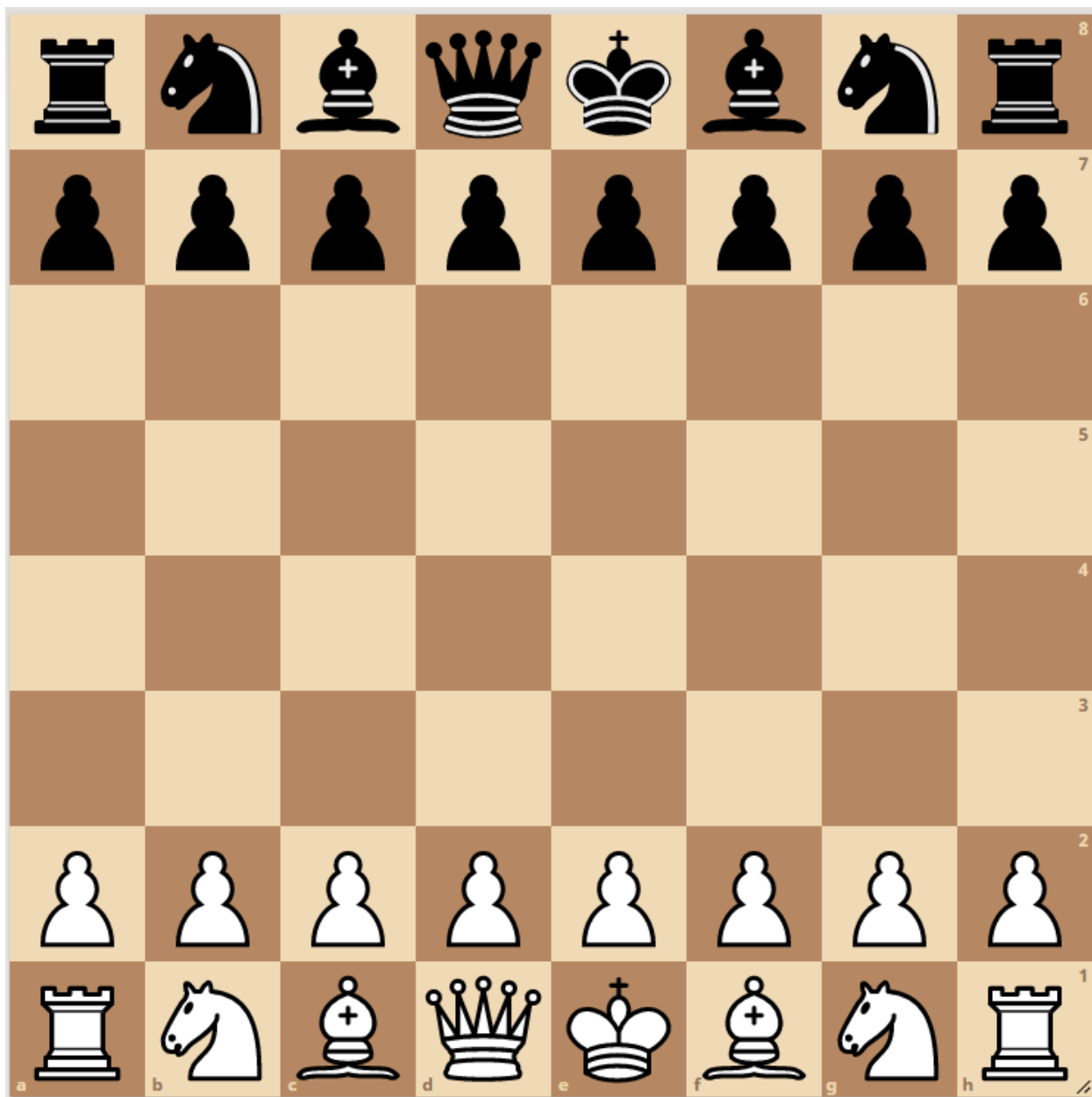
Brett

Plassering	Beskrivelse	HEX	Eksempel
------------	-------------	-----	----------

1a	Grønn	#769656	
1b	Gul	#eeeeed2	


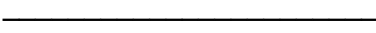
Lichess.com

Lichess tilbyr mange forskjellige mulige fargekombinasjoner for brikker og brett. Nedenfor er default. Denne ser ut til å være lik det samme brettet som blir produsert av Chessboard.js. Uvisst hvilket stammer fra hva.





Farger - Default

Brikker

Brikker	HEX	Eksempel
Sort	#000000	
Hvit	#fff	

Brett:

Plassering	Beskrivelse	HEX	Eksempel
1a	Brun	#b58863	
1b	Oransje	#f0d9b5	


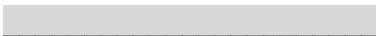
Chess24.com

Denne stylingen fraviker fra dei to tidligere tjenestene. Her har brettet ein treverktekstur. Fargene nedanfor vil derfor berre være ein av nyansane i teksturen.





Farger - Default

Brikker:

Brikker	HEX	Eksempel
Sort	#232323	
Hvit	#d8d8d8	

Brett:

Plassering	Beskrivelse	HEX	Eksempel
1a	Mørke Brun	#572f23	
1b	Lys Brun	#936b54	

Domenetermonologi

Ordbok:

<https://chess24.com/glossary>

Research: Programmeringsspråk, rammeverker og webserver.

Innledning

Vi vil undersøke programmeringsspråk, rammeverk og webservere vi kan bruke i utviklingen av SjakkSkjerm. Disse tre komponentene må undersøkes da SjakkSkjerm skal være en nettside.

Bakgrunn

SjakkSkjerm vil være en nettside. Ett ønske er å ha en skjerm hvor elementer blir oppdatert, uten at hele siden blir lastet inn på nytt. En av delen, som vil bli hyppig oppdatert er en visning av sjakkbrett. Oppdragsgiver har gitt gruppen informasjon om to JavaScript biblioteker som kan vise sjakkbrett. En annen del er en resultatoversikt basert på siste spilte runde. Denne tjenesten bruker sammen Java-rammeverket Spring Boot for utrekning og Javascript-rammeverket Vue.js for presentasjon. Samtidig har oppdragsgiver foreslått JavaScript bibliotekene ChessBoard.js og Chess.js for visning av sjakkbrett. Det er naturlig å ta med seg dette videre. Utenom det er det ingen begrensinger på hva vi kan utforske for backend. Undersøkelse av webserver må naturlig følge hva som er støttet i rammeverker.

Det er begrenset med tid, så vi gruppen vil begrense fokus til Javascript for frontend og Java for backend.. Derav vil vi ikke fokuserer på språk som kan kompilere eller transpilere til JavaScript, foruten språk som er en spesialisering av JavaScript. Vi fokuserer ei heller på språk som er såkalte "server-side" nett-språk, slik som PHP, ASP.net, osv. Vi har derimot undersøkt litt om andre backend-språk for å kunne ha litt kjennskap til senere.

Resulateter

Frontend

Brukergrensesnittet mot SjakkSkjerm blir en nettside. Nettsider består i dag av tre komponenter, som sammen former en nettside. Komponentene er HTML, CSS og Javascript. HTML strukturerer innholdet semantisk. CSS tar seg av presentasjonen av innholdet og presentasjonen av strukturen på innholdet. HTML og CSS er statiske komponenter, altså etter nettsiden er tegnet vil ikke innholdet og presentasjonen kunne dynamisk endres. JavaScript gjør det mulig å legge til oppførsel og funksjonalitet til elementer på en nettside. Det gjør dette ved utføre endringer på en nettlesers (som modelrepresentasjon av en nettside, kalt Document Object Model. Det finnes og andre programmeringsspråk som er en spesialisering av

JavaScript. Det mest kjente er TypeScript. JavaScript er et dynamisk typet programmeringsspråk. TypeScript er et statisk typet programmeringsspråk. Vi vil videre diskutere JavaScript, JavaScript-rammeverker, og TypeScript.

JavaScript

JavaScript er et dynamisk typet, tolket (istedet for kompilert), multiparadigmet språk. JavaScript lar oss dynamisk oppdatere innhald. Det lar oss interagtere med en nettlesers API, og derav Document Object Model (DOM) API'en.

Alt JavaScript-rammeverker gjør er mulig å gjøre i ren JavaScript. JavaScript-rammeverker tilbyr derimot en del biblioteker som tilbyr nyttig funksjonalitet som er krevende å etterligne og tidsmessig krevende å produsere i vanlig JavaScript. Særlig er logikk som skal dynamisk endre innholdet basert på events på en nettside mer komplisert å gjøre selv.

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Rammeverker

React

Beskrivelse:

Et JavaScript bibliotek for å bygge JavaScript brukergrensesnitt. Er produsert av Facebook. Det tar mål som å være et lettvektsbibliotek, og har ingen annen funksjonalitet enn å tegne og kontrollere brukergrensesnittet for en tjeneste.

Fordeler og ulemper

Pros: <ul style="list-style-type: none">• Easy to learn and use• Component-based: reusable code• Performant and fast• Large community	Cons: <ul style="list-style-type: none">• JSX DSL is required• Poor documentation
---	---

Forklaringer

JSX er ein syntax utvidelse av JavaScript, som er utviklet for React. Det produserer React "elementer". Er ment til å gjere det lettere å enkapsulere "bekymringer"/concerns, i React komponenter. JSX gjør det mulig at komponentene kan innehalde både logikk, og markup av elementer.

Bruksområder og eksempler på bruk

Brukergrensesnitt

Er brukt hos blant andre Facebook.com, Uber, AirBnb, Instagram.com.

Eksempel syntax:

```
const names = ['John', 'Sarah', 'Kevin', 'Alice'];
const displayNewHires = (
  <ul>
    {names.map(name => <li>{name}</li> )}
  </ul>
);
ReactDOM.render(
  displayNewHires,
  document.getElementById('root')
);
```

Linker:

<https://www.freecodecamp.org/news/reacts-jsx-vs-vue-s-templates-a-showdown-on-the-front-end-b00a70470409/#.ycvoyji7a>

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

Vue

Beskrivelse:

Et progressivt rammeverk for å bygge brukergrensesnitt. Det er progressivt i den forstand at man trenger bare kjernebiblioteket for i View-layeret, men man kan legge til flere moduler etter behov, helt opp til at man kan produsere SPA (Single Page Applications).
Bruker, som React, en Virtuel DOM.

Fordeler og ulemper:

<p>Pros:</p> <ul style="list-style-type: none">• Performant and fast• Component-based: reusable code• Easy to learn and use• Good and intuitive documentation• Liten bibliotekstørrelse, 18kb• Stort utviklersamfunn	<p>Cons:</p> <ul style="list-style-type: none">• Fewer resources compared to a framework like React• Over flexibility at times• Template DSL• Stor del av utviklersamfunnet snakker ikke engelsk.
---	--

Forklaringer

Template er eit HTML-liknande DSL. Det blir plugget inn i HTML filene. Det har noen iterative funksjonaliteter, slik som for-løkker.

Bruksområder og eksempler på bruk

Web-brukergrensesnitt og SPA.

Er i bruk hos blant annet 9GAG, Gitlab, Buzzfeed, Nintendo, Alibaba

Eksempel syntax:

```
const names = ['John', 'Sarah', 'Kevin', 'Alice'];
const displayNewHires = (
  <ul>
    {names.map(name => <li>{name}</li> )}
  </ul>
);
ReactDOM.render(
  displayNewHires,
  document.getElementById('root')
);
```

Linker:

<https://www.freecodecamp.org/news/reacts-jsx-vs-vue-s-templates-a-showdown-on-the-front-end-b00a70470409/#.ycvoyji7a>

<https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

Angular

Beskrivelse:

En utviklingsplattform for å bygge mobile og desktop web applikasjoner. Tillatter JavaScript, TypeScript og andre språk. Er utviklet av Google og er ett av de mest brukte JavaScript rammeverkene. Er et ganske omfattende rammeverk, som tar for seg mange aspekter, slik som kontroll av brukergrensesnittet, reaksjoner basert på hva brukeren gjør, validering, routing, sending av HTTP requests, testing og bygging av applikasjoner og sammenkobling på tvers av flere. Har en egen CLI for oppretting av Angular prosjekter.

Fordeler og ulemper:

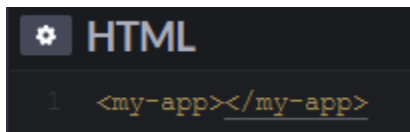
<p>Pros:</p> <ul style="list-style-type: none">• Fast server performance• MVC Architecture implementation• Component-based: reusable code• Good and intuitive documentation• Er vedlikeholdt av Google.• Har god støtte for testing og debugging.	<p>Cons:</p> <ul style="list-style-type: none">• Steep learning curve• Angular is very complex
--	---

Bruksområder og eksempler på bruk

Webapplikasjoner

Er i bruk hos Visa.com, Delta Airlines.

Eksempel syntax:

A screenshot of a code editor with a dark background. At the top, there is a tab labeled 'HTML' with a gear icon to its left. Below the tab, the code line is: 1 <my-app></my-app>. The code is written in a light blue font, and the line number '1' is in a lighter blue font.


```

1 // app.js
2 const { Component, VERSION } = ng.core;
3
4 * @Component({
5     selector: 'my-app',
6     template: `
7         <h1>{{title}} </h1>
8         TODO: Define your Angular
9         ${VERSION.major} component.
10     `
11 })
12 * class AppComponent {
13     title="hello world angular 6";
14 *   constructor() {
15       // TODO: Define your Angular
16       component implementation
17   }
18 }
19 |
20 // main.js
21 const { BrowserModule } =
22     ng.platformBrowser;
23 const { NgModule } = ng.core;
24 const { CommonModule } = ng.common;
25
26 * @NgModule({
27 *   imports: [
28       BrowserModule,
29       CommonModule,
30   ],
31   declarations: [AppComponent],
32   bootstrap: [AppComponent],
33   providers: []
34 })
35 * class AppModule {}
36
37 const { platformBrowserDynamic } =
38     ng.platformBrowserDynamic;
39
40 platformBrowserDynamic()
41     .bootstrapModule(AppModule)
42     .catch(err => console.error(err));
43
44 console.log('hola');

```

Linker:

<https://jaxenter.com/the-pros-and-cons-of-choosing-angularjs-124850.html>

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

Svelte

Beskrivelse:

En ny måte å bygge web applikasjoner. Bruker en deklarativ syntax, hvor du enkapsulerer koden din i komponenter. Den deklorative koden blir compilert til JavaScript kode. Denne JavaScript koden updaterer DOM'en direkte.

Fordeler og ulemper

Pros: <ul style="list-style-type: none">• No virtual DOM• Truly reactive• Easy to learn and use• Component-based: reusable code	Cons: <ul style="list-style-type: none">• Small community• Confusion in variable names and syntax
---	---

Bruksområder og eksempler på bruk

Webapplikasjoner.

Er brukt hos Noppo.pro, men fant ingen kjente bedrifter som bruker det.

Eksempel syntax:

REPL App.html

```
<h1>Hello {name}!</h1>
```

main.js

```
import App from './App.html';

const app = new App({
  target: document.querySelector('main'),
  data: { name: 'world' }
});

// change the data associated with the template
app.set({ name: 'everybody' });

// detach the component and clean everything up
app.destroy();
```

Linker:

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

Sammenligninger

Ytelse

Sammenlikninger gjort for Chrome versjon 88.0.4323.96. Sammenligner Svelte versjon 3.29.4, Vue versjon 3.0.2, Angular versjon 8.2.14, React versjon v17.0.1.

Sammenligningen ble funnet på denne nettsiden:

https://krausest.github.io/js-framework-benchmark/2020/table_chrome_87.0.4280.66.html

Duration in milliseconds \pm 95% confidence interval (Slowdown = Duration / Fastest)

Name Duration for...	svelte- v3.29.4	vue-v3.0.2	angular- v8.2.14	react- v17.0.1
Implementation notes				
create rows creating 1,000 rows	125.0 \pm 1.3 (1.00)	131.8 \pm 1.6 (1.06)	145.4 \pm 4.9 (1.16)	174.4 \pm 2.1 (1.40)
replace all rows updating all 1,000 rows (5 warmup runs).	127.3 \pm 0.4 (1.10)	116.1 \pm 0.7 (1.00)	129.8 \pm 1.2 (1.12)	136.8 \pm 0.8 (1.18)
partial update updating every 10th row for 1,000 rows (3 warmup runs). 16x CPU slowdown.	162.7 \pm 3.4 (1.15)	171.6 \pm 7.1 (1.21)	141.5 \pm 3.8 (1.00)	240.7 \pm 1.3 (1.70)
select row highlighting a selected row. (no warmup runs). 16x CPU slowdown.	34.7 \pm 1.4 (1.00)	161.9 \pm 4.1 (4.66)	73.6 \pm 1.0 (2.12)	133.3 \pm 3.8 (3.84)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4x CPU slowdown.	49.5 \pm 0.4 (1.00)	54.2 \pm 1.2 (1.09)	415.2 \pm 1.8 (8.38)	422.9 \pm 2.9 (8.54)
remove row removing one row. (5 warmup runs).	21.9 \pm 0.2 (1.00)	23.2 \pm 0.4 (1.06)	25.4 \pm 0.4 (1.16)	25.3 \pm 0.5 (1.15)
create many rows creating 10,000 rows	1,192.6 \pm 25.4 (1.08)	1,103.9 \pm 3.9 (1.00)	1,229.9 \pm 17.7 (1.11)	1,624.5 \pm 53.7 (1.47)
append rows to large table appending 1,000 to a table of 10,000 rows. 2x CPU slowdown	259.2 \pm 2.0 (1.01)	256.6 \pm 4.0 (1.00)	269.1 \pm 4.1 (1.05)	317.4 \pm 2.5 (1.24)
clear rows clearing a table with 1,000 rows. 8x CPU slowdown	139.7 \pm 0.8 (1.09)	127.7 \pm 0.9 (1.00)	233.0 \pm 1.7 (1.83)	146.3 \pm 0.7 (1.15)
geometric mean of all factors in the table	1.05	1.24	1.57	1.82
compare: Green means significantly faster, red significantly slower	compare	compare	compare	compare

Startup metrics (lighthouse with mobile simulation)

Name	svelte-v3.29.4	vue-v3.0.2	angular-v8.2.14	react-v17.0.1
consistently interactive a pessimistic TTI - when the CPU and network are both definitely very idle. (no more CPU tasks over 50ms)	1,954.2 ± 0.3 (1.00)	2,105.5 ± 9.8 (1.08)	2,847.4 ± 1.3 (1.46)	2,583.6 ± 2.5 (1.32)
script bootup time the total ms required to parse/compile /evaluate all the page's scripts	16.0 ± 0.0 (1.00)	16.0 ± 0.0 (1.00)	85.7 ± 1.1 (5.36)	35.4 ± 22.4 (2.22)
total kilobyte weight network transfer cost (post-compression) of all the resources loaded into the page.	145.9 ± 0.0 (1.00)	196.5 ± 0.0 (1.35)	302.1 ± 0.0 (2.07)	272.5 ± 0.0 (1.87)
geometric mean of all factors in the table	1.00	1.13	2.53	1.76

Memory allocation in MBs \pm 95% confidence interval

Name	svelte-v3.29.4	vue-v3.0.2	angular-v8.2.14	react-v17.0.1
ready memory Memory usage after page load.	1.1 (1.00)	1.2 (1.14)	2.7 (2.49)	1.3 (1.22)
run memory Memory usage after adding 1000 rows.	2.7 (1.00)	3.5 (1.32)	5.1 (1.89)	4.4 (1.64)
update eatch 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	3.0 (1.00)	3.7 (1.23)	5.4 (1.81)	5.2 (1.72)
replace 1k rows (5 cycles) Memory usage after clicking create 1000 rows 5 times	3.2 (1.00)	4.1 (1.28)	5.9 (1.83)	5.2 (1.60)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	2.4 (1.00)	2.6 (1.10)	4.2 (1.77)	3.0 (1.23)
geometric mean of all factors in the table	1.00	1.21	1.94	1.47

Keyed results.

Med keyed menes det at data og eit dom element er bundet med en nøkkel (key). Altså mellom view og model. Forandring av data vil forandre key, og vil altså være en operasjon i tillegg til å endre dataen. Fordelen er at får en 1-1 forhold mellom data og eit dom-element. Ikkje-keyed vil nødvendigvis være raskere. Derimot har man ikke den garantien for ett 1-1 forhold og man kan bruke opp igjen dom-elementer. Kan skape tilstandsproblemer for eit dom-element. Ikkje-keyed sammenligninger blir ikke tatt med.

Geometric mean

Et gjennomsnittstall for alle målinger for de ulike operasjonene. Lavere tall betyr bedre ytelse. Ytelsen er også vist frem gjennom farge. Jo grønnere farge, jo bedre ytelse. Jo røder farge, jo verre ytelse.

Resulater

Jevnt over er Svelte det raskeste rammeverket. Dette er naturlig da det deklorative språket man skriver blir compilert om til rein JavaScript. De tre andre rammeverkene har en form for virtuell DOM og vil nødvendigvis utføre flere operasjoner og bruke mer minne. Merk, disse resultatene gjelder bare Chrome. Flere nettlesere bruker sine egne JavaScript-motorer, så resultatene her vil nødvendigvis ikke være overførbare.

Læring og dokumentasjon

State Of Js er en årlig undersøkelse for å identifisere trender innen utvikling med JavaScript. De har noe forskjellig fokus vært år, basert på hva som er nytt for JavaScript og trendene i industrien. I rapporten for 2018 presenterte de en oversikt over profesjonelle utvikleres synspunkter for blant annet React, Vue.js og Angular. Svelte var ikke tatt med i sammenligningen. De har ikke produsert en tilsvarende sammenligning siden denne rapporten.

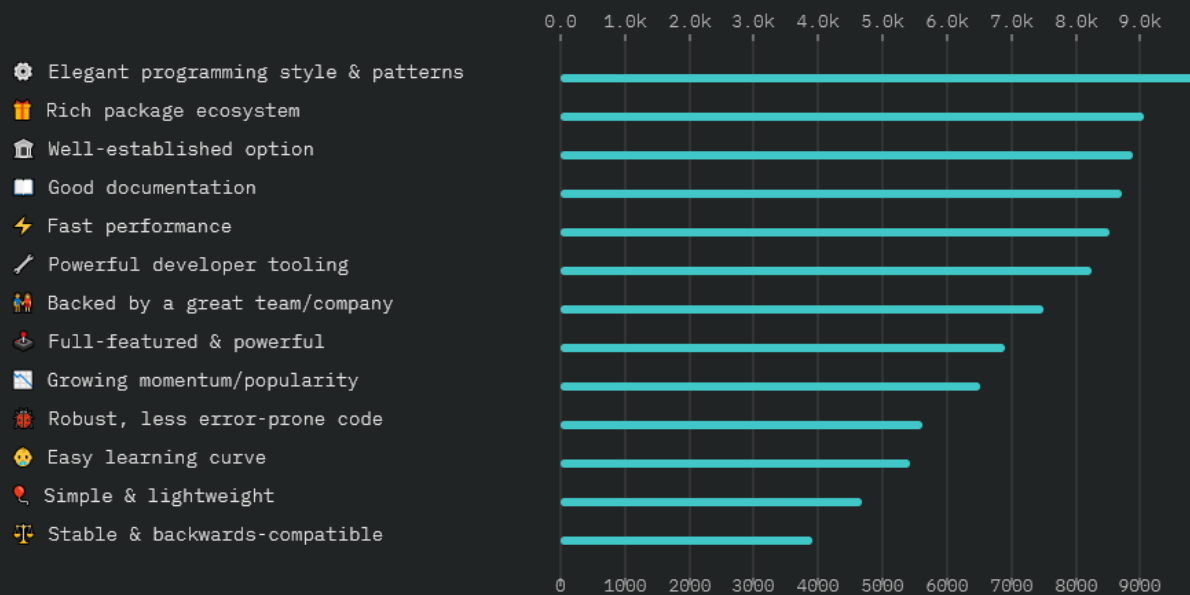
Link til State Of Js:

<https://stateofjs.com/>

React

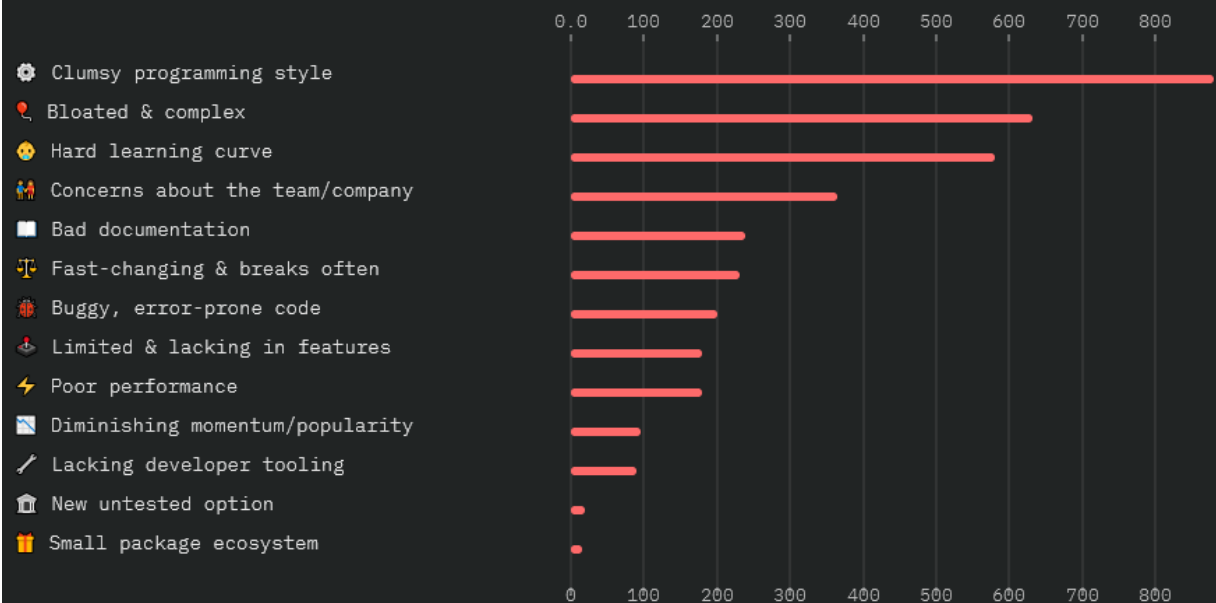
Mest likte aspekter rundt React for utviklere som bruker språket og vil bruke det igjen.

Most liked aspects of React among developers who picked "used it and would use again".



Minst likte aspekter rundt react for utviklere som har brukt språket og ikke vil bruke det igjen.

Most disliked aspects of React among developers who picked "used it and would *not* use again".

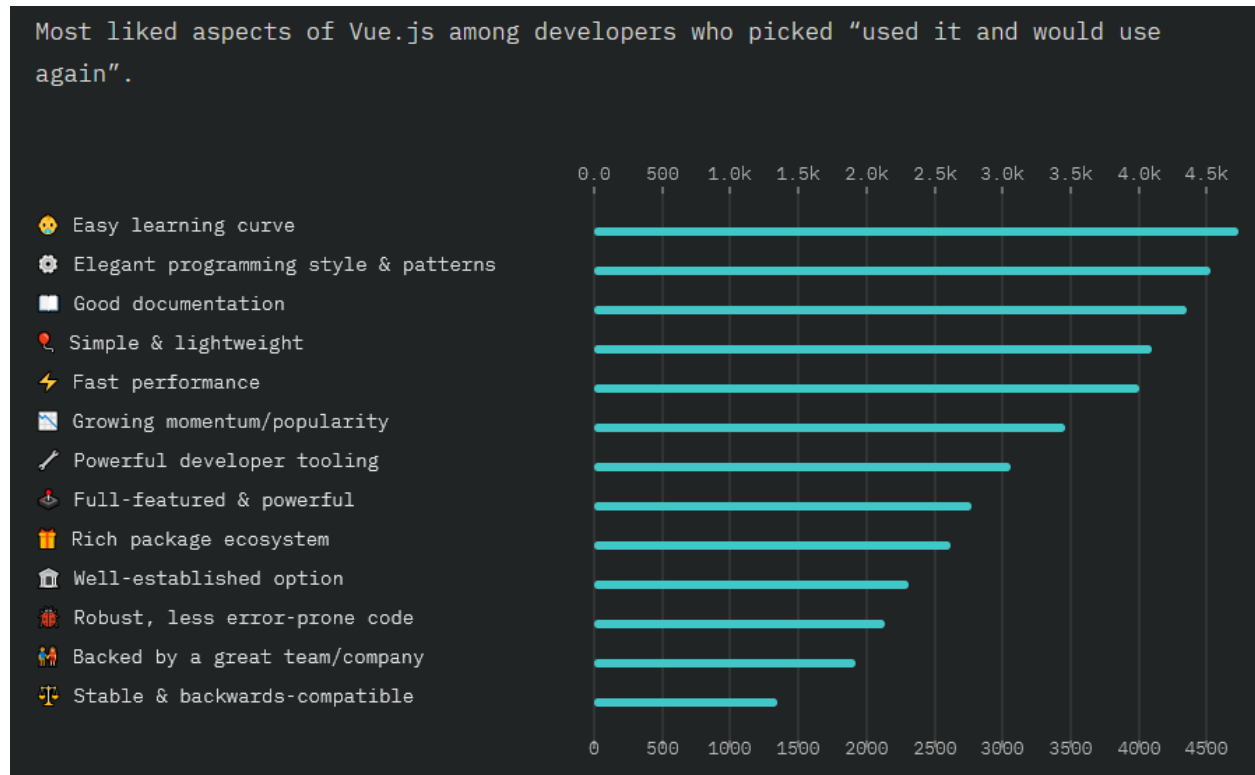


Link til undersøkelse:

<https://2018.stateofjs.com/front-end-frameworks/react/>

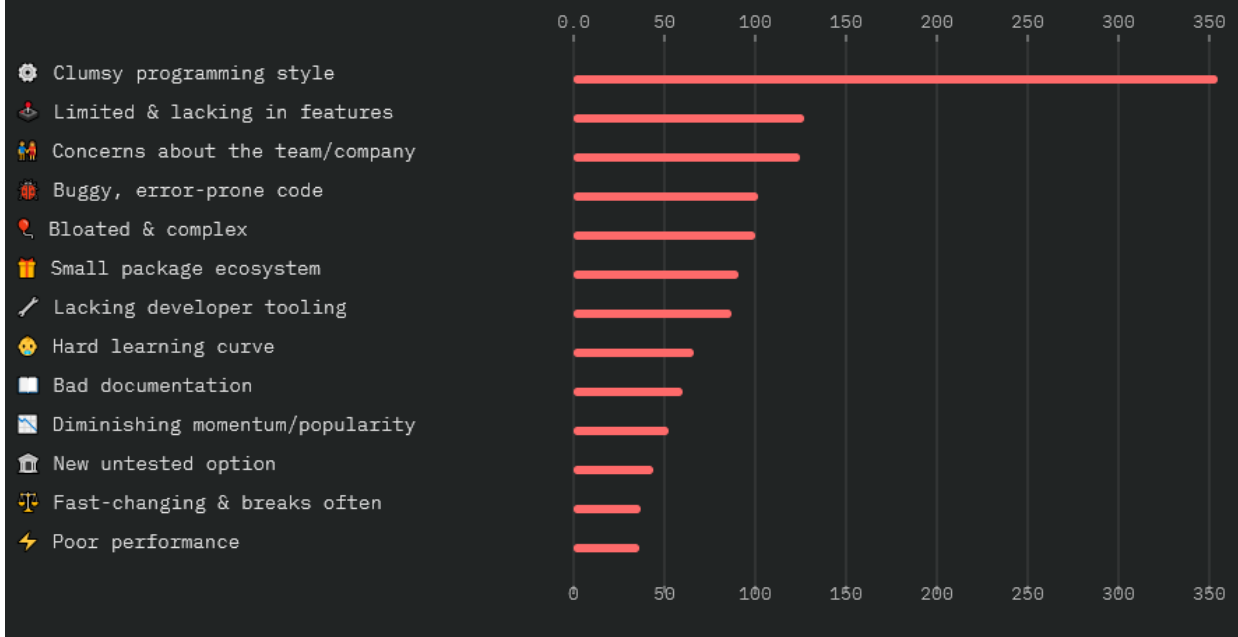
Vue.js

Mest likte aspekter rundt React for utviklere som bruker språket og vil bruke det igjen.



Minst likte aspekter rundt react for utviklere som har brukt språket og ikke vil bruke det igjen.

Most disliked aspects of Vue.js among developers who picked "used it and would *not* use again".



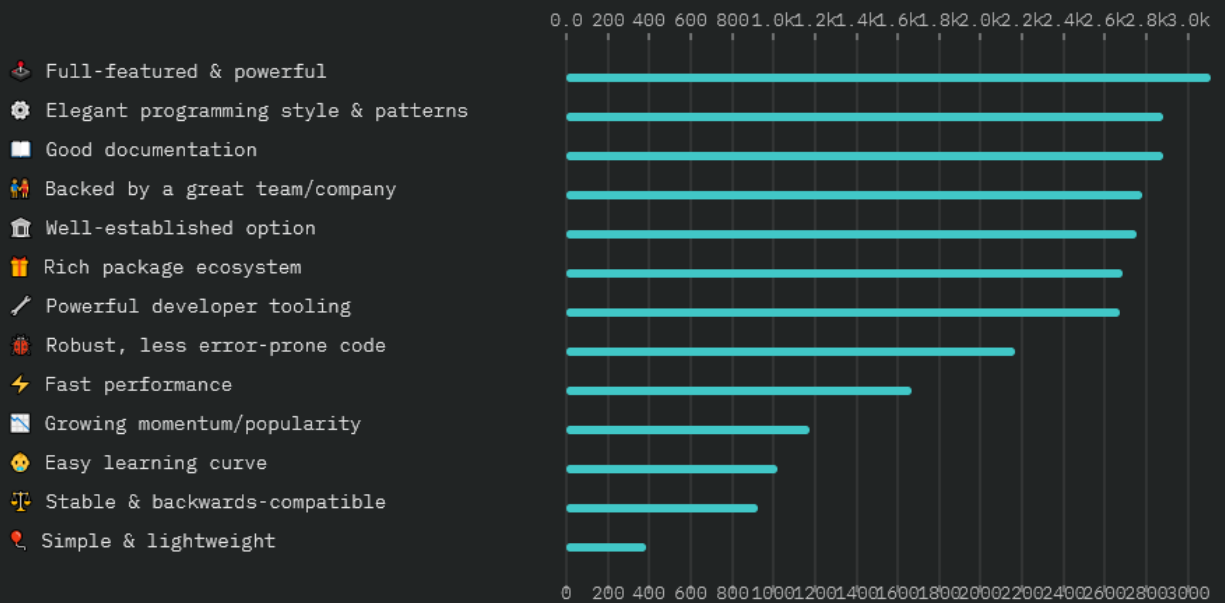
Link til undersøkelse:

<https://2018.stateofjs.com/front-end-frameworks/vuejs/>

Angular

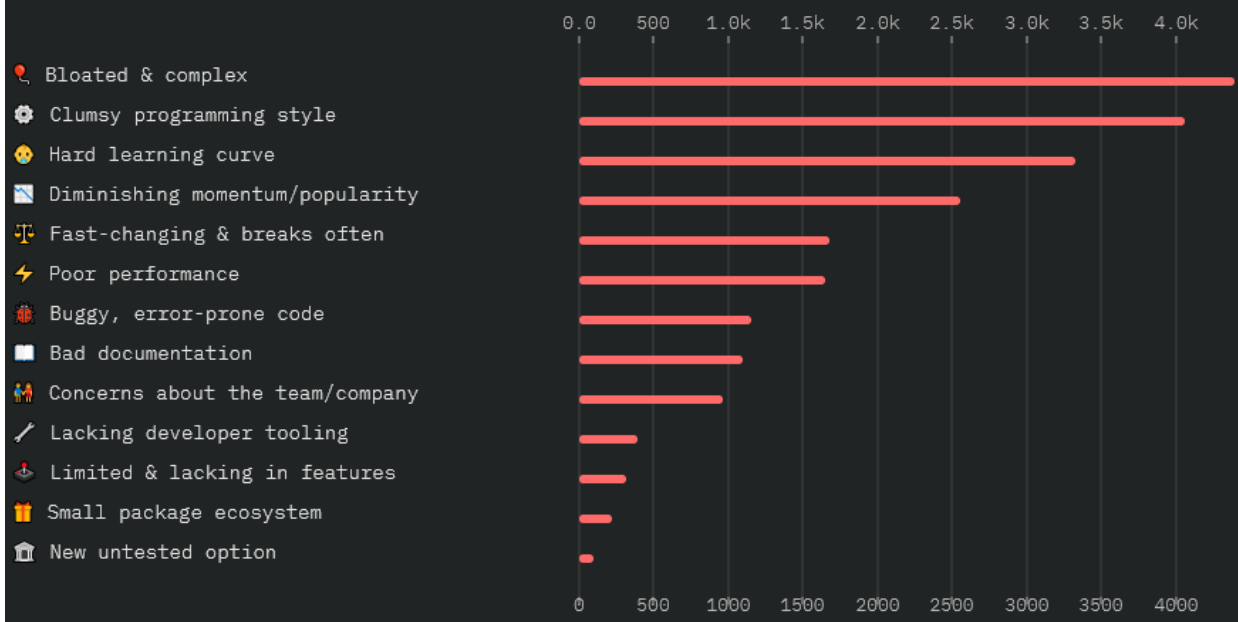
Mest likte aspekter rundt React for utviklere som bruker språket og vil bruke det igjen.

Most liked aspects of Angular among developers who picked "used it and would use again".



Minst likte aspekter rundt react for utviklere som har brukt språket og ikke vil bruke det igjen.

Most disliked aspects of Angular among developers who picked "used it and would *not* use again".



Link til undersøkelse:

<https://2018.stateofjs.com/front-end-frameworks/angular/>

Resultater

Undersøkelsene tok for seg mange aspekter som ikke er relevante for læring og dokumentasjon. Vi vil først og fremst ta for oss aspektene "Good Documentation"/"Bad Documentation" og "Easy learning curve"/"Hard learning curve". Noen andre kan nemnes for kontekst.

Vi ser i resultatene i bildene over at dette er absolutte tall, og at React er det språket som klart flest utviklere har prøvd, med over 9000 besvarelser på det best likte aspektet rundt rammeverket. Derav er det mindre verdi å sammenligne rene tall for samme aspekt. Det har mer for seg å sammenligne posisjonene til de ulike aspektene for hvert av rammeverkene.

React:

God dokumentasjon er rangert som 4 øverste aspektet, mens enkel læringskurve er helt nede på 11. rangering av 13 av hva utviklere liker.

En hard læringskurve er det 3. Mest mislikte aspektet rundt React, mens dårlig dokumentasjon er det 5.

Vue.js:

Enkel læringskurve er rangert øverst, mens god dokumentasjon er rangert på 3. Rangering. Hard læringskurve og dårlig dokumentasjon er nede på 8. og 9. Rangering over mislikte aspekter.

Angular:

God dokumentasjon er rangert 3. Øverst, mens enkel læringskurve er helt nede på 11.plass for aspekter som utviklere liker.

Hard læringskurve er helt oppe på 3. Plass over mislikte aspekter, mens dårlig dokumentasjon er 8.plass.

Det er verdt å merke seg at Angular har 3 besvarelser for hva som utviklere liker minst som har flere besvarelser enn alle besvarelser om hva utviklere liker best for det samme rammverket. I tillegg har det fleste aspektene for hva utviklere misliker med Angular fått flere besvarelser enn de samme aspektene for de andre rammeverkene. De er helt klart det minst likte rammeverket.

Ut i fra disse sammenligningene er det Vue.Js som kommer best ut i kategorien om læring og dokumentasjon.

Backend:

For backend er det egentlig utallige språk man kan bruke. Dagens web-apper har som oftest en REST API backend, og det aller fleste språk har biblioteker inkludert i språket for dette eller gode rammeverker for funksjonaliteten. Deler av vårt produkt vil bruke en REST-API arkitektur. I dag er de mest populære språkene for backend for webapplikasjoner er Java, Python, Ruby, JavaScript (Node.js). I og med gruppen har liten erfaring med JavaScript og dets rammeverker vi kommer til å bruke, så vil vi fokusere på det kjente, Java. Her er det derimot rammeverker vi ikke har brukt før vi kan sammenligne, hvor de aller fleste følger JAX-RS spesifikasjonen.

Java:

REST-Rammeverker

Informasjon hentet fra denne siden:

<https://rapidapi.com/blog/top-java-rest-frameworks/>

Apache CFX

Følger JAX-RS spesifikasjonen

Jersey

Følger JAX-RS spesifikasjonen. Brukes av Weblogic og Glassfish applikasjonsservere.

RESTEasy

Følger JAX-RS spesifikasjonen. Brukes av JBOSS EAP og Wildfly applikasjonsservere.

Spring MVC

Spring rammeverket sin REST modul fraviker fra de aller fleste web rammeverker. De følger ikke JAX-RS spesifikasjonen, men har heller sine egne annoteringer for REST endepunkter. Funksjonaliteten er derimot temmelig lik. Det er i tillegg en mulighet å bruke Jersey-rammeverket sammen med Spring Boot, og slik få JAX-RS endepunkter. I følgende link er det en sammenligning mellom annoteringene og en forklaring på forskjellen mellom JAX-RS (spesifikasjonen) og Spring rammeverket.

<https://stackoverflow.com/questions/42944777/difference-between-jax-rs-and-spring-rest>

Spring MVC (Model View Controller) er Spring sin modul for REST-applikasjoner.

Generelle applikasjonsrammeverker og spesifikasjoner

Spring.io

Beskrivelse

Spring rammeverket i sin helhet er det mest brukte Java rammeverket generelt sett. Det har mange moduler for ulike formål. Det har og sin egen tjeneste for opprette prosjekter kalt Spring Initializr. Tjenesten lar deg spesifisere om det er ett Maven eller Gradle prosjekt, om hvilket språk du utvikler i (Java, Kotlin og Groovy er støttet), om skal produseres en JAR (Java Application Resource) eller WAR (Web Application Resource) artifakt, hvilke avhengigheter prosjektet har (JDBC API, Lombok, osv). Den produserer grunnklassene du trenger, prosjektfiler, gitignore filer osv.

Link Spring Initializr:

<https://start.spring.io/>

Fordeler og ulemper

Fordeler	Ulemper
<ul style="list-style-type: none">- Stort utviklersamfunn, både i Norge og i utlandet.- Verktøy og rammeverkfunksjonaliteter som lar deg komme raskt i gang med å utvikle en applikasjon.- Modulært, Vil du feks ha en REST-applikasjon, trenger du bare Spring MVC modulen.- Har innebygd støtte for minst 3 applikasjonsservere.- God støtte for produsere enhets- og intergrasjonstester- Åpen kildekode	<ul style="list-style-type: none">- Moduler kan være overflødig store da de er laget for generelle formål.- Mange avhengigheter man nødvendigvis ikke har bruk for. Gir større minneavtrykk.- Manglende kvalitet for applikasjonsovervåking- og logging-verktøy- Gammel Spring kode kan være vanskelig å oppdatere.

Bruksområder og eksempler på bruk

Omtrent hva som helst.

Blir brukt av blant andre Walmart, Trivago, Netflix.

Jakarta EE

Beskrivelse

Jakarta EE er et sett med spesifikasjoner for bedriftsapplikasjoner. Jakarta EE (Enterprise Edition) startet som Java EE. Java EE spesifikasjonen og rammeverk som fulgte spesifikasjonen var da utviklet av Oracle. Java EE ble så utfaset av Java, for så å bli videreført av Eclipse Foundation, da under navnet Jakarta EE. Eclipse GlassFish 6.0, utgitt desember 2020 er så langt eneste fulle implementasjonen av Jakarta EE spesifikasjonen.

Mye av spesifikasjonen er som sagt innebygd i eldre versjoner av Java, under navnet Java EE. En av Java sine nåværende LTS-versjoner (Long Time Support), versjon 8, har denne spesifikasjonen innebygd, og Oracle tilbyr støtte mot betaling frem til 2030. Siste versjon som inneholdt noe av spesifikasjonen var Java 9.

Fordeler og ulemper

Fordeler	Ulemper
<ul style="list-style-type: none">- Åpen kildekode- Flere prosjekter, Payara, Wildfly, Quarkus, m.f. har som mål å følge deler av spesifikasjonen- God støtte for mikrotjenester gjennom Eclipse Microprofile standarden- Løser de samme problemene som Spring	<ul style="list-style-type: none">- Moduler kan være overflødig store da de er laget for generelle formål.- Mange avhengigheter man nødvendigvis ikke har bruk for. Gir større minneavtrykk.- Blir sett på som tungvektig og mindre utviklervennlig.

Bruksområder og eksempler på bruk

RedHat, Payara, Wildfly med flere, utvikler rammeverker som følger spesifikasjonene. Deres løsninger brukes av mange bedrifter.

Under følger en liste over Payara sine kunder per februar 2021:

<https://www.payara.fish/about/our-customers/>

Applikasjonsservere med Webserver funksjonalitet

Tomcat

Beskrivelse

Følger ikke Java/Jakarta EE spesifikasjonen, men har en klar webserver vinkling. Er den mest brukte applikasjonsserveren for Java.

Er default inkludert om man bruker "spring-boot-starter-web" artifact id'en for Maven prosjekter. for å initialisere prosjekter. Kan og initialiseres med "spring-boot-starter-tomcat" templat.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Fordeler og ulemper

Fordeler	Ulemper
----------	---------

<ul style="list-style-type: none"> - God integrasjon med Spring - Større utviklersamfunn med erfaring med Tomcat. - Enklere for bruk for nett-tjenester - God sikkerhet - Åpen kildekode 	<ul style="list-style-type: none"> - Krever ekstra oppsett om man vil bruke den som en mer generell applikasjonsserver (Spring kan ta seg av dette?) - Mangler mer avansert funksjonalitet - Mindre godt brukergrensesnitt for driftadministrasjon
---	---

Mer info: <https://www.trustradius.com/products/apache-tomcat/reviews?qs=pros-and-cons>

Bruksområder og eksempler på bruk

Mange steder hvor Spring brukes sammen med Webserver.

Link under gir et innblikk i omfanget av bruk:

<https://enlyft.com/tech/products/apache-tomcat>

Jetty

Beskrivelse

Følger Java EE/Jakarta EE spesifikasjonen, men er følger JAX-RS spesifikasjonen.

Kan initialiseres for Spring med “spring-boot-starter-jetty” templatene.

Fordeler og ulemper

Fordeler	Ulemper
<ul style="list-style-type: none"> - God integrasjon med Spring - Driftet av Eclipse, innebygd støtte i Eclipse IDE - Lite ressursavtrykk - Åpen kildekode 	<ul style="list-style-type: none"> - Krever ekstra oppsett om man vil bruke den som en mer generell applikasjonsserver (Spring kan ta seg av dette?) - Mangler mer avansert funksjonalitet - Mindre godt brukergrensesnitt for driftadministrasjon

Mer info: <https://stackshare.io/stackups/jetty-vs-tomcat>

Bruksområder og eksempler på bruk

GlassFish

Følger Java EE spesifikasjonen.

WildFly

Følger Java EE spesifikasjonen.

Payara

Beskrivelse

Følger Jakarta EE spesifikasjonen. Er en avskyter av GlassFish.

Fordeler og ulemper

Fordeler	Ulemper
<ul style="list-style-type: none">- Kompatibel med Jakarta EE<ul style="list-style-type: none">- Støtter derav så å si "alt"- God ytelse- Micro-versjonen har lite minneavtrykk	<ul style="list-style-type: none">- Da den følger Jakarta EE spesifikasjonen har den ikke god støtte i Spring- Mindre god i bruk for mikrotjenester.- Tungvingt å sette opp og drifte

Bruksområder og eksempler på bruk

Webserver.

Brukt av blant andre BMW gruppen, Papa Johns og SWISSCOM

Mere info: <https://www.trustradius.com/reviews/payara-server-2019-06-19-06-18-17>

Andre Språk:

Python

Beskrivelse

Python er et dynamisk typet, ikke-kompilert, multiparadigmet språk. Det har to store nettrammeverker, Django og Flask. Django er et stort rammeverk, som tilbyr det meste man forventer av et nettrammeverk. Flask har noe mindre funksjonaliteter enn Django, men er mer modulært oppbygd

Linker:

<https://testdriven.io/blog/django-vs-flask/>

<https://hackr.io/blog/flask-vs-django>

Fordeler og Ulemper

Fordeler	Ulemper
<ul style="list-style-type: none">- Språket har en lett syntax- Gode rammeverker i Flask og Django- Stort utviklingsmiljø	<ul style="list-style-type: none">- Dynamisk typet, kan gi utfordringer for OO programmering.- Ikke kompilert, vanskeligere å luke vekk typefeiler.

	- Kan være tregere
--	--------------------

Bruksområder og eksempler på bruk

Maskinlæring, Nettjenester

Django, og derav Python er brukt av blant andre Instagram, Spotify, Mozilla.

<https://djangostars.com/blog/why-we-use-django-framework/>

Go

Beskrivelse

Statisk typet, compilert språk utviklet hos Google. Har sterkt minnesikkerhet og søppelsamling.

Fordeler og Ulemper

Fordeler	Ulemper
<ul style="list-style-type: none"> - Raskt - Lett syntax - Laget for moderne programvareutvikling - Sterk minnesikkerhet - Rutiner gjør multitrådprogrammering enklere 	<ul style="list-style-type: none"> - Ungt og mindre modent språk <ul style="list-style-type: none"> - Mangler noen abstraksjonsteknikker slik som Generics - Mindre utviklingsmiljø - Mangler GUI biblioteker

Link:

<https://hackernoon.com/should-i-go-the-pros-and-cons-of-using-go-programming-language-8c1daf711e46>

<https://www.toptal.com/go/4-go-language-criticisms>

Bruksområder og eksempler på bruk

Brukt hos Google for nettverksservere og distribuerte applikasjoner.

Brukt av blant andre Docker, Uber, Medium

Link:

<https://hackernoon.com/should-i-go-the-pros-and-cons-of-using-go-programming-language-8c1daf711e46>

JavaScript

Beskrivelse

Det er mulig å bruke JavaScript som et backendspråk gjennom å bruke Node.js. Node.js tilbyr et lokal JavaScript kjøretidmiljø. Dette gjør at man kan kjøre JavaScript utenfor en nettleser som har en JavaScript motor. Spesifikt bruker Node.js Chrome sin V8 JavaScript motor.

Fordeler og ulemper

Fordeler	Ulemper
<ul style="list-style-type: none">- Gjør det mulig å bruke samme utviklingsspråk både for server og for klient.- Stort utviklingsmiljø- JavaScript har god støtte for å utvikle asynkron programkode.- Lettvektig	<ul style="list-style-type: none">- V8 JavaScript motoren har bare tilgang på en enkel CPU-kjerne, gjør den mindre skalerbar.- Mindre støtte for programmering mot relasjonsdatabaser.

Bruksområder og eksempler på bruk

WebSocket servere

Brukt av blant andre: Netflix, Uber, Walmart, LinkedIn.

Link: <https://www.simplytechnologies.net/blog/2018/4/18/8-top-companies-that-rely-on-nodejs>

Sammenligninger

Vi sammenligner REST-rammeverk, Generelle applikasjonsrammeverker, Applikasjonsservere for Java.

Ytelse

Ser bort fra ytelse, da det ikke er mange sammenligninger å finne for noen av kategoriene.

Læring og dokumentasjon

Rammeverker

Spring.io er helt klart det rammeverket som har størst brukerbase individuelt. Det følger derav at det finnes mye ressurser på internett. Samtidig sier profesjonelle utviklere at dokumentasjonen er god. For JAX-RS spesifikasjonen finnes det flere rammeverk som implementerer den. Ressurser om spesifikasjonen blir mer fragmentert da det finner flere som implementerer den og noen er spesifikt om den ene implementasjonen. I tillegg, siden Spring er så mye mer populært nå i dag enn et enkelt JAX-RS-implementerende rammeverk, så har ressursene for JAX-RS en tendens til å være en del eldre og i noen tilfeller utdaterte. Gruppen har erfaring med at når vi leter etter ressurser for JAX-RS så finner man ofte tilsvarende ressurs for Spring før man finner den for JAX-RS.

Applikasjonsservere

Tomcat følger med Spring Boot som default applikasjonsserver. Tomcat og Jetty er det som er mest naturlig med å bruke for Spring. Jetty lar man følge JAX-RS spesifikasjonen for REST-api'er.

Når det kommer til å følge Java EE/Jakarta EE spesifikasjonen, er GlassFish, Payara og Wildfly valgene. Alle tre følger den fullt ut, noe som betyr at de inneholder alt man forventer av en Java EE/Jakarta EE applikasjonsserver. Tomcat og Jetty blir sett på som "lettere" enn disse tre, da de ikke støtter all den funksjonaliteten.

Alle har god dokumentasjon, men siden Tomcat er default for Spring, og Spring er det mest brukte rammeverket, så finner man mer ressurser angående Tomcat enn de andre.

Drøfting

Frontend - JavaScript og rammeverk

Funksjonalitet

Når man sammenligner hva som er mulig i JavaScript rammeverkene virker det som Angular selvsagt er den definitive vinneren fordi den inneholder så masse tjenester som KAN være brukbare for et prosjekt. Det er ikke dermed sagt at det største rammeverket alltid det beste for et prosjekt. Det rammeverket man bør bruke er selvsagt det som støtter det man trenger for det gitte prosjektet. React har som nevnt mindre støtte enn Angular, men om man ser kun på oppsettet av brukergrensesnittet så kan React faktisk være et bedre valg enn Angular. Årsaken til dette er at React er bygd opp på en slik måte at den styrke er nettopp brukergrensesnitt, og i motsetning til Angular trenger den ikke en del koblinger mellom de forskjellige komponentene for å sørge for at alt sitter sammen godt og trygt. Flere av manglene alle disse rammeverkene har, f.eks. React sin mangel på routing kan enkelt løses da samfunnet rundt React har laget eksterne pakker som legger til denne tjenesten.

Både Vue og Angular har tjenester innebygde som fjerner unødvendige deler av prosjektet for å gjøre den mer optimalisert. F.eks. dersom du ikke trenger en gitt validering eller lignende, vil ikke denne pakken være med i sluttproduktet.

Læring

Vue er oftest sett på som det rammeverket som er enklest å lære seg. Noe som er en fordel for oss som er relativt uerfarne innenfor JS. I tillegg sier andre utviklere at dokumentasjonen er god. Dette er også et stort pluss for å kjappere komme i gang og for å finne løsninger på problemer.

Samlet vurdering

Gruppen vil bruke Vue.js for å bygge deler av et brukergrensesnitt for siden. Det er flere begrunnelser for dette. Gjennom denne undersøkningen viser det at rammeverket er blant det med best ytelse, profesjonelle utviklere synes det er enkelt å lære og har god dokumentasjon. Det er og et modulært rammeverk hvor man kan legge til moduler for mer funksjonalitet om man

trenger det. Det gjør og at man ikke trenger å ta med seg mange unødvendige avhengigheter og store biblioteker. I tillegg finnes det en allerede utviklet komponent med Vue.js frontend vi kan ta med oss inn i prosjektet, som er utviklet av oppdragsgiver. Oppdragsgiver er og tilgjengelig for å kunne hjelpe oss med rammeverket.

Særlig det at rammeverket er enkelt å lære og har god dokumentasjon er særdeles viktig for gruppen da gruppen har minimal erfaring med JavaScript.

Vi ser ikke for oss en utstrakt bruk av Vue.js, men steder som krever delvis tegning av nettsiden basert på hendelser på server kan vi få bruk for det.

Backend - Java, rammeverker og applikasjonsservere.

Ytelse

Tok ikke med sammenligning.

Læring og dokumentasjon.

Det finnes mye mer ressurser for Spring.io rammeverker enn for JAX-RS spesifikasjonen. Dokumentasjon for Spring.io er god. For applikasjonsservere finner man flest ressurser om Tomcat.

Samlet vurdering

Gruppen ser bort fra andre språk og deres rammeverker da de sees på som en for stor oppgave å lære seg et nytt programmeringsspråk, samtidig som tjenesten kan bli utfordrende å utvikle. I andre situasjoner enn en Bacheloroppgave med begrenset tid og andre obligasjoner, ville det falt naturlig å prøve ut disse språkene, utvikle noen prototyper for hvert av dem og ta vurderinger derfra.

Spring MVC følger ikke JAX-RS spesifikasjonen. Dette sees på som en ulempe, da gruppen har kjennskap til denne spesifikasjonen. Spring Boot + MVC har sine egne tilsvarende biblioteker, metoder og annoteringer. I tillegg tillater Spring Boot at man integrerer Jersey rammeverket, sammen med andre Spring moduler. På den måten er det mulig å følge JAX-RS spesifikasjonen. Gruppen ser for seg at overgangen til gjøremåtene til Spring ikke er så stor.

Det finnes og en komponent som allerede er utviklet som bruker kombinasjonen Spring Boot + Tomcat. En annen fordel for Spring er deres Spring Initializr. Denne tjenesten gjør det veldig simpelt og raskt for å komme i gang med utvikling, samtidig som den har tilgang til mange konfigurasjoner.

Vi ser for oss at deler av serverdelen må ha kunne kommunisere med flere nettlesere for å distribuere events fra ulike turneringer. Dette vil måtte involvere å kjøre kode på flere tråder. Java har god støtte for flertrådprogrammering, men er ikke det enkleste å jobbe med asynkron programmering. Ser ikke for oss at det blir så mye av det, så Java passer godt.

For dokumentasjon og læring, er Spring.io og Tomcat det fremste valget.

Sett under ett vil gruppen gå for en stack med Spring Boot + MVC og Tomcat. Denne kombinasjonen er naturlig da Spring Boot bruker Tomcat som sin default applikasjonsserver.

Konklusjon

Gruppen går for JavaScript + Vue.js for frontend og Java + Spring Boot/MVC + Tomcat for backend.

Ressurser

URL'er for artikler brukt for denne rapporten. Er i rekkefølge etter hvor de er i teksten, fra topp til bunn.

Frontend

JavaScript

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

<https://www.freecodecamp.org/news/reacts-jsx-vs-vue-s-templates-a-showdown-on-the-front-end-b00a70470409/#.ycvoyji7a>

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

<https://www.freecodecamp.org/news/reacts-jsx-vs-vue-s-templates-a-showdown-on-the-front-end-b00a70470409/#.ycvoyji7a>

<https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

<https://jaxenter.com/the-pros-and-cons-of-choosing-angularjs-124850.html>

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

<https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>

https://krausest.github.io/js-framework-benchmark/2020/table_chrome_87.0.4280.66.html

<https://stateofjs.com/>

<https://2018.stateofjs.com/front-end-frameworks/react/>

<https://2018.stateofjs.com/front-end-frameworks/vuejs/>

<https://2018.stateofjs.com/front-end-frameworks/angular/>

Backend

Java

Rammeverker

<https://rapidapi.com/blog/top-java-rest-frameworks/>

<https://stackoverflow.com/questions/42944777/difference-between-jax-rs-and-spring-rest>

<https://start.spring.io/>

<https://www.payara.fish/about/our-customers/>

Applikasjonsservere

<https://www.trustradius.com/products/apache-tomcat/reviews?qs=pros-and-cons>

<https://enlyft.com/tech/products/apache-tomcat>

<https://stackshare.io/stackups/jetty-vs-tomcat>

<https://www.trustradius.com/reviews/payara-server-2019-06-19-06-18-17>

Python

<https://testdriven.io/blog/django-vs-flask/>

<https://hackr.io/blog/flask-vs-django>

<https://djangostars.com/blog/why-we-use-django-framework/>

Go

<https://hackernoon.com/should-i-go-the-pros-and-cons-of-using-go-programming-language-8c1daf711e46>

<https://www.toptal.com/go/4-go-language-criticisms>

<https://hackernoon.com/should-i-go-the-pros-and-cons-of-using-go-programming-language-8c1daf711e46>

JavaScript

<https://www.simplytechnologies.net/blog/2018/4/18/8-top-companies-that-rely-on-nodejs>