# Ramverk1

## Kmom01

### Gör din egen kunskapsinventering baserat på PHP The Right Way, berätta om dina styrkor och svagheter som du vill förstärka under kursen och det kommande året.

Jag tror mina styrkor skulle vara att jag förstår att jag hela tiden håller på att lära mig, och att jag med det helt enkelt inte kan mycket. Varje gång jag tror jag kan något, så visar det sig att jag inte kan det. Varje gång jag säger jag ska göra eller hålla något, så visar det sig att det aldrig händer. Kanske min styrka under huvudvärket Xanax, (host, host), skulle vara en controller. Jag gillar stuket på en controller, men jag kan inte påstå att detta skulle vara någon speciell stor styrka i mig. Jag tror inte jag kan påstå att jag känner mig säker i någonting, för jag är ju här för att lära mig.

Detta leder till mina svagheter. Vilket är bland annat objekt orienterad programmering. Jag vill lära mig mera om abstrakta klasser, traits och kanske interface. Jag skulle vilja se mer av detta, då det känns som oop är ganska stort i den vida världen och vanligt att programmera på det sättet. Meta programmering, om jag inte kommit i kontakt med detta i Python kursen, är något jag skulle vilja se hur det fungerar i PHP. Xdebug är något vi kanske har kommit i kontakt med under förra PHP kursen; det skulle vara kul att kunna debugga i sin terminal om det gick med ett ganska bra sätt som om man gjorde det i Visual Studio som ett grafiskt gränssnitt.

### Vilket blev resultatet från din mini-undersökning om vilka ramverk som för närvarande är mest populära inom PHP (ange källa var du fann informationen)?

Min första undersökning baseras på sidan <https://www.cloudways.com/blog/best-php-frameworks/>. Där har vi för året 2019 som vi befinner oss i, på första plats Laravel; andra Symfony. I följande nedstigande ordning kommer Yii 2, CakePHP, och på femte plats CodeIgniter. Resterande på listan är bland annat Zend Framework, Phalcon, Slim Framework och FuelPHP. På en annan sida <https://raygun.com/blog/top-php-frameworks/>, förutom Laravel på första plats, verkar ordningen lite annorlunda. Nummer två är CodeIgniter och plats tre Symfony. En tredje sida (<https://hackernoon.com/top-8-php-frameworks-in-2019-b6be163605c8>) har samma ordning som den förra, dock utan en viss numrering: Laravel, CodeIgniter och Symfony.

Jag tror topp tre verkar vara Laravel, CodeIgniter och Symfony.

### Berätta om din syn/erfarenhet generellt kring communities och specifikt communities inom opensource och programmeringsdomänen.

Det här är just sådana scenarion som kan uppstå när man ska försöka upprätthålla något, dividerat på flera aktörer inblandade; aktörer som inte är verkliga aktörer såsom en myndighet som alltid har stöd ifrån sin stat. Och även om sista aktören är självaste författaren eller skaparen bakom ett visst språk, vad händer om den personen dör eller skiftar till något annat? Det är väldigt svårt att få alla komponenter att fungera hela tiden som i ett kugghjul med ett konstant flöde av tuggande kugge för kugge för att få saker att röra sig kring axeln. Som videon nämnde, vissa communities (user groups) i USA träffades inte på flera månaders tid; vissa försvann. Andra utfall sker, man kan tycka om något annat, går över ifrån ljuvliga PHP till något snarliknande som Drupal; snarliknande då Drupal är skrivet i PHP. Rykten är något som också kan göra att någon som läser något illa om PHP eller dess community, speciellt på sidor såsom sociala medier, får sin uppfattning att inte gå med i rörelsen.

Det verkar som kärnan av comunityn (PHP gemenskapet), har en demokratisk rörelse om sig. Detta i mina ögon är fel. Även om språket i någons ögon är det språket som har mest open source om sig, så är det ändå farligt att bygga en rörelse eller gemenskap på fria röster utan en slutlig röst eller, som många darrar för att höra i dessa politiskt korrekta tider, auktoritet. Utan auktoritet (som även finns till viss del inom demokratin, som ofta byggs då på ens erfarenheter, beroendes på i vilket sammanhang denna gemenskap kretsar kring) kan man kanske aldrig få en slutlig röst, om inte det sker inom, som vi har i många länder, omröstning. Men omröstning betyder inte att det alltid är det bästa som kommer fram, även om majoriteten verkar för det. Nej, man måste ha mer än så, man måste vila på någons axlar som, när de faller ihop pga. tex. ålder, går vidare till en annan. Denna auktoritet har slutrösten eller det slutliga ordet inom vissa saker som också gör att, hur dåliga axlar (erfarenhet eller annat), ändå håller lågan vid liv. Detta gör att man alltid har en slutlig stomme som alltid finns där, en pelare som lyser, hur än lite denna låga i vissa tider må lysa, som en ledsagare för den som vill ro till det stora språket PHP bland alla stormar och vågor (programmeringsspråk och tekniker).

Desto flera i en gemenskap (community) desto fler kontakter knyts. Det blir enklare att hitta ett jobb om man skulle bli arbetslös, för alltid så är det nästan att någon känner någon annan.

Därför, den negativa stora saken väger alltid mer än de flera positiva sakerna. Om den negativa saken säger, ”om detta fallerar fallerar allting”, så finns den alltid där som en stötesten, oavsett hur bra något går eller hur flera goda positiva saker finns inblandade. Igen, se till att PHP gemenskapen, eller språket får någon form av myndig eller statlig organisation bakom sig, och även ifall många positiva saker därutav fallerar, så finns alltid grundstommen där som håller upp hela gemenskapen.

### Vad tror du om begreppet “en ramverkslös värld” som framfördes i videon?

Jag tror jag kan relatera mig ganska mycket till Michael Cullum. Varför? för jag har ett sådant slags sätt i mig. Jag föredrar att, som utvecklare, och givetvis, jag är en minst sagt nybörjare inom det ordet, att köra mycket från scratch, mycket från att bygga så långt jag kan från grunden något än att ständigt vara beroende av externa komponenter. För det är något sämre att kalla sig en viss ramverksutvecklare än självaste utvecklare. Det låter som man är mer beroende av något ramverk än att kunna bemästra flera genom att använda sig av externa bibliotek och ta ifrån de olika ramverken vad man kan än att köra med ett ramverk i sin applikation. Jag kan ha fel, girighet och oerfarenhet leder till ramverk på arbetsplatserna. Få utvecklare tror jag kan göra vad tex. ramverken gör, tex. en dropdown meny i ren CSS eller med ren JavaScript. Istället blir det bootstrap. Ibland får man använda sig av ett ramverk om det är bästa sätt för att bygga sin applikation, för tid är något man också måste ta i beaktning.

Om man håller sig till vad den mannen sade i sin applikation: ”business logic”, ”libraries” och ”glue” så kan man komponera ihop sin applikation till ett ramverkslöst sådant. För det gäller att använda sig av vad man har till sitt förfogande och kunna göra något med det än att vara för beroende av något man aldrig egentligen behöver, speciellt om man kan sina saker som utvecklare och hur de sitter ihop och hur man kan få dessa olika komponenter att samarbeta.

Jag är helt för för att bli en ”framework agnostic”. En sådan som kanske helst slipper att komma i kontakt med många ramverk genom att kunna bättre än ramverken och snickra ihop egen kod som gör vad de flesta ramverken gör. Så när någon frågar om jag känner till ett visst ramverk, måste jag ursäkta mig och säga, att jag inte känner till det, för att jag är en ramverksagnostiker.

Till viss del bra information i videon. Jag trodde inte att det verkligen fanns sådana människor där ute. Min uppfattning på en arbetsplats är, ramverk efter ramverk efter ramverk, att man blir rädd att tappa sin fina kodstil och kunskap för att bara skriva mera någon annans kod av ett grundspråk, vilket är hemskt.