

Lektion 9/11

Hemmuppgift

*Som vanligt har ni en hemuppgift till nästa vecka, ni arbetar förstås vidare på ert tennissystem. Eftersom klubbens ordförande är van vid IT så vill hen ha en sida med en **horisontell meny** och beställningen är en **responsiv sida**, som fungerar **både på stor och liten skärm, bred eller smal**. Börja med att göra menyn. Se sedan till att sidan blir responsiv, W3CSS brukar vara ett bra sätt att komma igång med den saken.*

Känns inte detta nog utmanande skulle du dels kunna se om du kan använda React för att göra menyn, räcker inte det kanske du bör tänka på koola animationer värdig en bollklubb.

Nästa vecka vill jag se hur det blev på git.

Inledning

För att göra inlämnandet av uppgifter här på kursen smidigare vill jag att ni i fortsättningen använder git. Detta sammanfaller väl med kursplanen som även den säger att ni skall lära er git. Därför flyttar vi kursens git-avsnitt till idag och till första lektionen nästa vecka.

Det bör även påpekas att era individuella projekt fortsätter och byggs ut kursen igenom. Det är era respektive färdiga projekt jag kommer att betygsätta, men för att ingen skall fastna på något strul i början av kursen vill jag att ni lämnar in varje vecka så att ni får en puff i rätt riktning varje vecka och så att jag får chans att se att ni verkar vara på banan allihopa.

Personligen använder jag github för den här typen av uppdrag och ni hittar kursfilerna som jag lägger upp under:

<https://github.com/Rnqst/WebutvHT22>

Skicka adresser till era git-repositorier angivande via epost eller teams så att jag kan se vad ni pysslar med.

Introduktion git

För det första hittar ni den officiella guiden till git här:

<https://git-scm.com/book/en/v2>

Den kan både läsas via nätet eller tankas ned i lämpligt format och den är tillgänglig på många språk. Wikipedia kan vara ett bra ställe att få en överblick:

<https://en.wikipedia.org/wiki/Git>

Git är ett revisionshanteringssystem, ett system för att hjälpa oss att hålla reda på källkod och andra dokument och hur dessa förändras med tiden (och av vem). Förutom att git är ett av de mer kompletta applikationerna på det här området, häng inte upp dig på att det saknas GUI, diverse kopplingar till dokumenthanteringssystem och diverse annat som tillverkarna av motsvarande kommersiella system använder för att sälja in sina produkter. Sådana funktioner finns som tillhör i väldigt många varianter så att även när vi räknar kringfunktioner så är git ovanligt komplett. Men

själva grundsystemet, kodarkivet, har fler funktioner än de flesta motsvarande system och till detta så är det en rad saker där git helt enkelt överträffar system rent tekniskt. Med en sådan inledning förstår du att vi även har baksidan där den tekniska nivån helt enkelt ger ett mer komplicerat arbetsflöde, i vissa speciella fall. Till vardags förenklas arbetsflödet för de flesta.

Vad gör ett revisionshanteringsystem, naturligtvis håller det reda på alla förändringar över alla filer i ett mjukvaruprojekt, sett över tiden. Med de flesta sådana system är det enkelt att grena av en ny gren på utvecklingsträdet. Det ovanliga med git är att det oftast är enkel och snabb operation att kopiera tillbaka förändringarna från en gren till en annan.

Den andra viktiga saken där git skiljer sig från de flesta är att vi får ett komplett arkiv när vi gör en lokal kopia på vår egen dator. Därmed fungerar git väldigt bra i vår mobila värld där alla springer runt med laptoppar som ibland har nätkoppling och ibland körs helt separat. På det viset kan alla på ett projekt arbeta när och var de vill för att sedan synkronisera de som de skapats när det är dags att dela med andra.

Det går utmärkt att köra git utan att använda Github och vill du ha en tjänst liknande Github så finns många alternativ, exempelvis Gitlab. På samma sätt är det enkelt att sätta upp en egen lokal gittjänst, både med "bara" git, men även med "hub"-liknande funktioner.

<https://www.youtube.com/watch?v=mJ-qvsxPHpY>

Installation

Du kan dels köra git direkt från kommandoraden, som upphovsmannen tänkte var det enklaste sättet att arbeta på. Eller så kör du något grafiskt verktyg som exempelvis Github Desktop:

<https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/installing-and-authenticating-to-github-desktop/installing-github-desktop>

Finns inte inbyggd git-stöd i din utvecklingsmiljö så finns git ofta som plugin, så även till Visual Studio Code. Ett smidigt alternativ för den som ändå sitter i sin editor hela dagarna. Kör du windows hittar du git här:

<https://git-scm.com/>

Linuxanvändare har git via sin pakethanterare, en väg som brukar vara enklast för mac-användarna också, men där finns det fler pakethanteringsalternativ. Homebrew är ett populärt alternativ:

<https://brew.sh/>

Master och Main, vad är problemet?

Vi har sedan några år en nätkampanj, eller kanske snarare folkrörelse, där de inblandade försöker upprätta olika grupper anseende och eliminera begrepp som kan kopplas till tidigare förtryck. När det kommer till git så är systemet i grunden flexibelt på många sätt som inte alla liknande system är, men sedan git kom har de flesta kallat stammen i sina repositorer för "master" för att markera att det är denna som projektet utgår från och att detta, så att säga, är "original-grenen". Men många förknippar termen "master" mer med vad de stackars slaverna i USA var tvungna att kalla sina så kallade "ägare", i stället för att se det som ett generiskt ord vars flesta användningar inte har någonting att göra med slavhantering över huvud taget. Oavsett den aspekten så krockar den populära synonymen "main" med huvudfunktionen i program från flera populära

programmeringsspråk vilket många tycker är förvirrande; tänk dig frågan "Var finns XYZ i programmet?" om någon svarar "i main" så blir plötsligt inte det svaret entydigt. Kanske hittar någon på ett tredje alternativ som inte retar någon.

Responsiv web

Responsiva sidor är webbsidor som själva anpassar sig så att de ser vettiga ut och så att de fortfarande är smidiga att använda när skärmupplösningen ändras. Förutom rena formateringsfaktorer, av mer estetisk natur, så brukar menyer och liknande strula till det när exempelvis skärmens bredd ändras. Många har ju idag skrivbordsdatorer med skärmbredd (ibland över flera fysiska skärmar) med 4-5000 pixlar eller mer. Detta samtidigt som det inte är ovanligt med färre än 1000 pixlar på bredden för den som använder en mobil. En meny som skall fungera i båda dessa lägen kan kräva lite omsorg från utvecklaren.

https://www.w3schools.com/html/html_responsive.asp

https://www.w3schools.com/whatis/whatis_responsive.asp

Observera att W3Schools har en egen CSS för responsiv webb som är mycket smidig att använda. Följ den här guiden:

<https://www.w3schools.com/w3css/default.asp>

W3CSS får användas och kopieras helt fritt.

Kevin Powell

Brukar ha bra videogenomgångar om webb, han har några intressanta saker om responsiv webb, börja med det här projektet, koden hittar du här på 'hubben:

<https://github.com/kevin-powell/responsive-made-easy>

Sedan slår du igång filmen på:

<https://www.youtube.com/watch?v=bn-DQCifeQQ>

Ta dig gärna tid att gå igenom denna, Kevin har som vanligt många guldkorn för den nyfikne. Han har även några andra klipp med småtips:

<https://www.youtube.com/watch?v=VQraviuwbzU>

<https://www.youtube.com/watch?v=0ohtVzCSHqs>

Bootstrap

Söker du på nätet är det stor chans att du hittar Bootstrap som "standardpaket" för att få responsiva sidor. Bootstrap fungerar säkert utmärkt. Observera att Bootstrap egentligen inte ingår i den här kursen, men det är mycket populärt och det kan vara bra att känna till.

[https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

https://www.w3schools.com/bootstrap/bootstrap_ver.asp

<https://getbootstrap.com/>

<https://getbootstrap.com/docs/3.4/css/>

Hur du använder Bootstrap med JavaScript

<https://getbootstrap.com/docs/3.4/javascript/>

De flesta brukar nog ändå tycka W3CSS är betydligt smidigare.

JavaScript

Som vanligt är W3Schools en bra start:

<https://www.w3schools.com/js/default.asp>

När browsern tolkar en sida skapar den ett så kallat DOM träd

https://www.w3schools.com/js/js_htmlDOM.asp

detta använder du för att komma åt och ändra de saker du vill hantera på sidan.
Ytterligare vägledning här:

<https://www.w3schools.com/jsref/default.asp>

Geekforgeeks har en annan bra tutorial:

<https://www.geeksforgeeks.org/javascript/>

FreeCodeCamp har en kurs som fokuserar på just JavaScript:

<https://www.youtube.com/watch?v=PkZNo7MFNFg>

Med materialet här:

<https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/>
<https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/es6/>

Vill du ha en mer kombinerad genomgång för JS i webbläsaren kanske den här kan vara något:
<https://www.youtube.com/watch?v=W6NZfCO5SIk>

React

React är ett JavaScript-paket ursprungligen framtaget av Facebook för att underlätta att göra användargränssnitt på webbsidor med JS. Tillsammans med AngularJS är React de allra populäraste kodbiblioteken för användargränssnitt i JS. I kursplanen står React så det är huvudalternativet här

[https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))

<https://reactjs.org/>

Som vanligt har vi W3School som hjälper oss igång:

<https://www.w3schools.com/react/default.asp>

Mozilla gör inte bara webbläsare, här har de en kurs i React:

[https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side JavaScript frameworks/React getting started](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started)

FreeCodeCamp bygger en väderapp med React:

<https://www.youtube.com/watch?v=Reny0cTTv24>

<https://github.com/bobangajicsm/react-weather-app>

AngularJS

Angular är Googles svar på Metas React. Båda är bra, väldokumenterade och populära verktyg. Jag tar med Angular här så att du får chans att titta på det populäraste alternativet till React här om du är nyfiken. Angular ingår inte i kursen men det kan vara intressant ändå. Angular är liksom React fria att användas för webbutvecklare.

<https://en.wikipedia.org/wiki/AngularJS>

<https://angularjs.org/>

<https://www.w3schools.com/angular/>

En liten tutorial på 'tuben:

<https://www.youtube.com/watch?v=3qBXWUpoPHo>