Git og Github

Hva er Git?

Git kan definerers som et versjonskontrollsystem. Det er laget for å kunne spore endringer i datafiler og strukturere arbied på prosjekter som er utført av flere mennesker. Det er myntet på utviklere for raskt og effektivt kunne lagre å oppdattere prosjektene de jobber med.

Hvordan bruke Git?

Git brukes ofte i terminal, hvor du gjennom kommandoer som for eks. "push" og "pull" enkelt laster opp din nye versjon av arbeidet, eller henter ned en ønsket prosjekt. Du definerer hvor du ønsker å hente data fra, eller hvor du ønsker å laste opp ny data og trykker kommando.

Hva syns jeg om Git?

Jeg syns git virker som et stabilt og fleksibelt verktøy som vil kunne effektivisere skolearbeidet mitt og på sikt også kundearbeidet. Jeg er positiv til bruk av git.

Hva er Github?

Github er verdens ledende plattform for software utvikling. Det er lagringsstedet som du sender data til gjennom git. I github lagres ikke bare gjelldende porsjekt, men også alle tidligere versjoner av samme prosjekt. Dette er en utrolig nytting sikkerhet for utvikliere. I tilfelle disk-kræsj eller tyveri vil du alltid ha backup av prosjektene dine. Denne måten å lagre data på er også en sikkerthet mot deg selv. Er du spontan og "rask på labben" vil du fort angre på at du slettet tidligere deler av arbeidet ditt, til fordel for nye ideer. Med versjonkontroll er det ikke noe problem lenger. Du kan bare "skru tiden tilbake" og forstette å jobbe der du siste gjorde en stor endring. For deg som utvikler blir github en portefølge av alle prosjektene dine. Du kan også benytte det som et privat bibliotek.

Hvordan bruke Github?

Du kan bruke Github på flere forskjellige måter. Github er tilgjengelig i nettleser, gjennom git eller via skrivebordsprogrammet GitHub desktop. Jeg velger å bruke GitHub desktop, da det er oversiktilig, brukervennlig den enkleste metoden hvis man ikke er så kjent med å bruke terminal.

Åpne GitHub – create new repository – legg mappestrukturen for prosjektet lokalt inn I det nye repository – publish repository.

Da pubeliseres første versjon av prosjektet ditt, til din Github-konto. Du kan logge deg inn på dine sider å hente prosjektet senere, eller dele det med andre. Prosjektetne dine vil være offentlige og tilgjengelig for andre så lenge du har "gratisversjonen" av Github.

Hva syns jeg om Github?

Som med git, så syns jeg Github virker som et stabilt og fleksibelt arbiedsverktøy som vil effektivisere arbeidet mitt. Versjonskontroll er vel det jeg liker best, da det er fort gjort å fjerne tidligere kode til fordel for nye og ikke alltid bedre idéer. Jeg tar det i bruk fra og med denne obligen.

Media typer

Hva er media typer?

Media typer er spesifikasjoner for ulike skjermer gjennom CSS. Mobilskjerm, laptopen, TV'en, den stasjonære pc'en eller prosjektoren. Det er en stil som gir deg en spesifik opplevelse av innholdet på nettsiden.

Noen medietyper i CSS:

All: for alle typer enheter.

Handheld: for håndholdte enheterv (ment for små skjermer med begrenset oppløsning). Etter innføringen av responsivt design er det omtrent ikke behov fro denne medietypen lenger.

Print: utskriftsversjon. Dersom du ønsker å skrive ut en nettside er det kun innholdet du er interessert i. Du har ikke behov for menyer eller annet materiale fra siden.

Projecsjon: for prosjektorer. Etter innføringen av responsivt design er det omtrent ikke behov fro denne medietypen lenger.

Screen: for pc-skjermer. Dette er den vanlige medietypen. Det er den du ser på pc-skjermen din og takket være responsivt design kan denne typen også brukes på skjermer til nettbrett, mobiler eller tv. For ettersom skjermbildet minker eller øker vil nettsiden organisere skjermbildet der etter.

Speech: for skjermlesere. Du kan la en skjermleser lese av innholdet og "bestemme selv" hvordan den vil gjøre det, eller ved spesifikke CSS-egenskaper bestemme hvodan tale-opplevelsen skal være. Lyder, stemmer og intonasjoner kan tilpasses den beste lytteopplevelsen.

Tv: for tv-skjermer (elder tv-skjermer, med begrenset oppløsnin og fargegjengivning). Denne har utgått litt med årene, da tv-skjermer stort sett har minimum oppløsning i full HD også opp til 4K. Det er ikke behov for å tilpasse innholdet lenger, da pc-skjerm og tv-skjerm ofte har samme oppløning.

Hvordan bruke ulike medietyper?

Du kan skrive et eget stilark til de ulike medietypene du vil ha tilgjengelige. Der kan du spesifisere hvordan opplevelsen av html-dokumentet skal være. Så linker du til stilarket i html-dokumentet og merker det med media="ønsket type". For eks. <link href=“print.css” rel=“stylesheet” media=“print” />.

Du kan også refferere til en annen mediatype i hoved-stilarket. For eks.

body { background-color: #333333; }

p { color: #ffffff; }

@media print {

body { background-color: #333333; }

p { color: #ffffff; }

\* { font-family: 'Times New Roman', Times, serif; }

}

CSS layouts

Hva er layout?

Layout er måten delene av noe er arrangert på. Vi må planlegge layout i stort sett alt vi lager. Skriver vi en artikkel, lager vi først en stor overskrift, deretter underoverskrifter, avsnitt, paragrafer, kommentarer, bilder, annen media, ren tekst, topptekst og bunntekst. Dette er har vi en visjon om hvordan skal se ut. Vi velger derfor en layout som skal gi brukeren eller leseren den best mulig opplevelsen av vårt arbeid. Dette er spesielt viktig på internett, da personer eller selskaper ofte er avhengige av å selge noe eller markedsføre seg selv på en eller annen måte.

Hva mener med layouts I CSS?

Design og orden er noe de aller fleste mennesker setter pris på, og særlig når de surfer på nettet. Dersom det er behagelige fargekombinasjoner, orden og struktur på nettsiden gir du brukeren den best mulige opplevelse av nettsiden. Dette gjøres med CSS. Gjennom internetts utvikling har også nettsidene blitt "penere" å se på. Dette er takket være ulike metoder og fleksible verktøy til utviklerene som lager strukturen og designet. Blant de nyere og mest moderne metodene, har vi CSS Grid og CSS Flexbox. Dette er to verktøy vi kan bruke til å få akkurat den layouten vi vil.

CSS Grid

CSS Grid er et verktøy hvor vi lager et "grid" på en nettside. Et grid kan sees på som en tabell hvor vi har kolonner og rader, antall etter ønske. Vi skjærer da nettsiden vår inn i celler som vi kan navngi og fortelle hva skal inneholde. Innholdet på nettsiden kan dyttes inn i en eller flere celler etter hvordan du som utvikler vil ha det. CSS Grid er utrolig oversiktlig og fleksibelt, da du også har automatiske funksjoner som til enhver tid vil kunne endre nettsiden etter antall celler du bruker eller har plass til i nettleservinduet ditt. Når nettsiden oppdatterer seg etter hvordan brukeren behandler den, kaller vi responsivt design. Man kan også tenke på det som å søke opp samme nettside på en stasjonær pc, med stor skjerm, og på mobilen sin. Med CSS Grid vil du kunne få plass til like mye informasjon på en tilsvarende oversiktlig måte på mobilskjermen, som på pcskjermen.

CSS Flexbox

CSS Flexbox er et verktøy for å posisjonere innhold. Du setter merkelapp på for eks. en rekke tekstbokser, for så å fortelle området du ønsker å "flexe", hvor du ønsker at merkelappen skal vises. Det kan høres komplisert og vanskelig ut, men er egentlig oversiktlig og enkelt og bruke. På samme måte som CSS Grid er CSS Flexbox også et kraftig og fleksibelt verktøy, da mulighetene for nøyaktig posisjonering oftest er opp til utviklerens kunnskaper og ferdigheter.

CSS Float

CSS Float er et verktøy som er blitt utdattert etterhvert. Etter at Flexbox og Grid har kommet på banen, blir float skjeldet brukt, da det er litt mer tungvindt å bruke. Hva float gjør ligger litt i oversettelsen av ordet. Float = flyt. Det lar deg flyte merkede elementer etter hverandre til høyre eller venstre. Det kan brukes til for eks. å lage må en meny til å se ut som et navigasjonsbanner. Utfordringen med float er at du til enhver tid må ta hennsyn til alle elementene dine rundt. Det er ofte brukt til å "wrappe" (bre rundt) tekst rundt bilder.