

Regla nr 1 2 og 3

ALDREI pusha ef þú ert ekki búinn að pulla frá main eða merga við main.

Að búa til GitHub account

Fyrsta skrefið er að búa til GitHub reikning, ferð inn á þennan hlekk: <https://github.com/join> og ferð eftir leiðbeiningunum sem koma upp á skjáinn. Hægt er að niðurhala GithubDesktop hérna: <https://desktop.github.com>.

Github Desktop

Tengjast repoinu (cloning)

Það fyrsta sem þarf að gera er að tengjast repo'inu. Í Github

Desktop velurðu takkan eftst í vinstra horninu, velur síðan "add", og velja "Clone repository". Það sem þú vilt setja inn í URL path er repository URL sem er inn á github. T.d "https://github.com/matthias1412/VNL1_19".

Í local path velurðu þá staðsetningu sem þú vilt hafa verkefnið í tölvunni þinni, getur þessvegna verið inn á Desktop. Þegar þú pullar, þá fara breytingarnar sem eru búnar að vera gerðar af öðrum þarna inn, þegar þú gerir breytingar og vilt pusha, er það það sem er í þessu folder það sem fer inná repo'ð. Þú getur séð allar breytingar sem þú hefur gert á folderinum undir "Changes" í Github Desktop.

Pulla

Þegar þú vilt uppfæra folderinn þinn með því sem er í repo'inu á Github, velurðu "Fetch Origin" takkann, ef það eru breytingar á repo'inu kemur takkinn sem "Pull" þegar þú ýtir á hann uppfærist folderinn

þinn. Breytingarnar sem aðrir eru búnir að pusha er komið í file'inn þinn.

Commit & Push

Nokkuð svipað og að pulla, þegar Github Desktop sér að þú hefur gert breytingar lýsist upp takki „commit changes“ þú vilt byrja á því að skrifa í „Commit summary“ stutta lýsingu á breytingunum sem þú ert að commita, og hafa lýsinguna meira ítarlegri í stóra boxinu fyrir neðan. Þegar það er komið velurðu „Commit changes“ – breytingin er ekki ennþá komin inn í repo'ð fyrr en þú ýtir á „push“.

Branches

Branch er hægt að horfa á sem önnur týpa af main, ef ég vil fara bæta við feature'inu að geta geymt starfsmenn, myndi ég brancha frá main og nefna branchið „Employee database branch“. Þaðan myndi ég vinna í öllum þeim kóða sem þarf, vanalega er main aldrei breytt nema þegar þessi feature-branch eru merguð inn í main. Þ.e það er aldrei gert beina breytingu á main.

Að búa til branch

Áður en þú branchar frá main, viltu alltaf pulla allar breytingar á main. Til að búa til branch í Github Desktop velurðu „current branch“ takkann. Velur síðan nafn fyrir branchið. Þú ert núna búinn að búa til annað main, nema þar sem þú ert bara að vinna. Enginn annar ruglar í þínu branchi. Þú getur flakkað á milli brancha með því að fara aftur inn í „current branch“. T.a.m ef þú vilt fara aftur í main, myndirðu velja main og velja að pulla. Þegar þú vilt síðan fara aftur á þitt branch gerirðu það sama.

Merging branches (Git Flow)

Þegar það kemur að því að feature-branchið þitt er fullþróað, þ.e. þú ert búinn að gera gagnagrunn fyrir starfsmenn og vilt setja það inn í main. Viltu merge þessi branches, feature-branchið þitt verður ekki lengur til, heldur mergast það inn í main.

Það fyrsta sem þú vilt gera er að merge feature branchið þitt við main branchið. Gott er að byrja á því að velja „fetch origin“ upp á það að feature branchið þitt er örugglega up-to-date eins og á að vera. Með feature branchið valið velurðu síðan „Branch“, og þá „Merge into current branch“, og velur main. Branch takkinn er staðesttur efst á skjánum eða í forritinu, fer eftir því hvort á macOS eða windows. Ef það eru einhver conflicts, Github Desktop mun þá láta þig vita, hægt er að leysa þessi conflict með því að opna Vscode og leita að conflict merkjum „<<<<<<, =====, >>>>>>“ og laga mismuninn (sjá conflict kaflann líka). Eftir að leysa þessi conflict vista skjalið og pusha breytingarnar í branchið.

Þá viltu loks fara í það að merge við main, til að gera það ferðu yfir í main branchið í Github Desktop, velur „Branch“, á macca er það efst á skjánum > og síðan „Merge into current branch“, í windows er það efst í forritinu sjálfu. Þar inni velurðu þitt feature-branch og velur að búa til merge commitið. Þá viltu fara inn á github.com og ýta á „Create pull request“.

Núna þarf a.m.k. einn hópfélagi að fara yfir þetta feature branch, þ.e. fara á branchið í sinni tölvu, keyra allt og reviewa, bara vera viss um að allt sé í lagi og að allt meiki sens. Þegar einn approvar merge requestið er hægt að lokum merge branchið inn í main.

Pull request & review

Eftir að merge feature-branchinu þínu viltu fara inn á Github.com og ýta á „Create pull request“, og biðja einhvern hópfélaga um að fara yfir kóðann. Sá hópfélagi ýtir á „Add review“, eftir að fara yfir kóðann og breytingarnar, velurðu síðan „Approve“ og hægt er að merge branchið við main, með því að fara í pull requestið og ýta á „merge pull request“

Conflicts

Það er alltaf að fara gerast að einhver conflicts koma upp, að laga það er ekki jafn flókið og það getur litið út. En það kemur vanalega upp þegar þú ert að reyna merge feature-branchið þitt við main branchið. Github Desktop sýnir hvaða filear eru með merge conflict, þú getur þá farið inn í Vscode og opnað þessa file sem eru með conflictið. Þar sem conflictið á sér stað eru þrír valmöguleikar Accepta breytinguna sem einhver annar gerði, accepta bara þína breytingu, eða hafa báðar breytingarnar inni. Þetta eru bara takkar sem þú ýtir á og eftir því sem á við. Þegar conflictið er búið að leysa er síðan aftur hægt að reyna merge'á og ef conflictin voru leyst þá ætti það að ganga í gegn.

Algeng vandamál

.gitignore

Stundum gæti gerst er að talvan ykkar býr til ósýnilegar skrár og github vil láta pusha því inn í repo'íð. Við viljum ALLS ekki fá endalaust af tilgangslausum skráum sem allir eru síðan að hala niður í tölvuna sína. Til að leysa þetta uppfærum við .gitignore. Segjum t.a.m. að

GithubDesktop sér ._MACOS sem skrá til að pusha, þú bætir þá einni línu í .gitignore ._, það sem þetta segir okkur er að allar skrár sem byrja á ._ eiga ekki að vera pushaðar inn í repo'ið.

Gleyma að committa á feature-branchið þitt

Þú vilt alltaf committa þínar breytingar áður en þú ferð yfir í nýja branch. Annars geta breytingar flakkað á milli brancha og jafnvel misst breytingarnar.

Aldrei vinna í main

Bara þú vilt aldrei vinna í main, vilt alltaf vera í þínu feature-branchi og síðan merge main inn í það (því aðrir gætu hafa pushað inn í main áður en þú branchaðir frá main, þannig þegar þú mergar við main án þess að merge main inn í feature branchið týnast þær breytingar sem annar gerði) og síðan merge við main.

Reyna halda breytingum innan einnar kröfu

Ekki reyna að gera margar kröfur í einu feature branchi, best að hafa commitin með sem minnst af breytingum. Þetta hjálpar þeim sem eru að fara yfir kóðann, einnig ef mistök koma upp, þá er hægt að restore'a main upp að nýjustu mistökunum, í staðinn fyrir að kannski taka út kóða sem má alveg vera inni og virkar.

Github í terminal (þarf ekki að kunna, en þægilegra)

Flækjustigið hækkar aðeins hérna, en í endann er þetta það þægilegasta þegar verið er að vinna með

mörg branches og þess háttar. En skiljanlegt ef einhver nennir ekki að setja þetta upp.

Að tengjast repo'inu í terminal/command line

1. Opnar terminal
2. Byrð til ssh encryption key: „ssh-keygen -t ed25519 -C your_email@gmail.com“
3. „ssh-add ~/.ssh/id_ed25519“
4. Copy'ar lykilinn, með því að gera
 - i. cat ~/.ssh/id_ed25519.pub
 - ii. copyar ALLT sem kemur, ss.: „ssh-ed25519 mjoglangurkoði your_email@gmail.com“
5. ferð þá inn á GitHub með þennan lykil
6. Velur myndina þína í efra hægra horninu > undir access velja „SSH and GPG keys“ > velja „new SSH key“
7. Copya lykilinn inn, setja nafn t.d „Local Laptop“
8. og síðast, inn í directoryinu þar sem þú vilt hafa repo'ið í terminal: „git clone [git@github.com:matthias1412/VNL1_19.git](https://github.com/matthias1412/VNL1_19.git)“

Allt er síðan með svipaða virkni og í github desktop, fyrir utan það að þú notar commands „git pull“, „git push“, „git status“, „git commit“, „git branch“, „git merge“, „git checkout“ etc.