



HÁSKÓLINN Í REYKJAVÍK

VERKLEGT NÁMSKEIÐ 1 - HÓPUR 19

## **Github Manual**

28. nóvember 2023

# Efnisyfirlit

<b>1 Inngangur</b>	<b>3</b>
<b>2 Almennar reglur</b>	<b>3</b>
<b>3 Að stofna GitHub reikning</b>	<b>3</b>
<b>4 Notkun á Github Desktop</b>	<b>3</b>
4.1 Cloning repository . . . . .	3
4.2 Pulling Changes . . . . .	3
4.3 Commit and Push . . . . .	4
<b>5 Að vinna með branches</b>	<b>4</b>
5.1 Að búa til branch og skipta á milli brancha . . . . .	4
5.2 Merging Branches (Git Flow) . . . . .	4
<b>6 Pull requests Review</b>	<b>5</b>
<b>7 Að leysa Conflicts</b>	<b>6</b>
<b>8 Algeng vandamál í Git</b>	<b>6</b>
<b>9 GitHub í Terminal (Ekki nauðsynlegt að kunna)</b>	<b>6</b>

# 1 Inngangur

Þetta handbók er hönnuð til að veita notendum grundvallarleiðbeiningar um notkun Git og GitHub, sérstaklega fyrir þá sem eru að byrja sín fyrstu skref í forritunar- og hugbúnaðarþróun. Hér eru skýrar leiðbeiningar um helstu aðgerðir eins og að stofna og nota branches, vinna með .gitignore, klóna repositories, og hvernig á að nota GitHub í terminal. Markmiðið er að veita notendum skilvirkt verkfæri til að stjórna kóða og samvinnu í verkefnum.

## 2 Almennar reglur

1. **regla nr 1, 2 og 3:** ALDREI pusha ef þú ert ekki búinn að pulla frá main eða merge við main.
2. **Regluleg commits:** Haltu breytingum skipulögðum með því að gera reglulega commits. Þetta gerir það auðveldara að fylgjast með þróun verkefnisins og leysa vandamál ef þau koma upp. T.d ef þú ert að vinna í virkni, síðan allt í einu fokkast allt upp og ekkert virkar, getur þá farið til baka í nýjasta commitið þar sem þú veist að verkefnið virkar.
3. **Skýrar Lýsingar á Commits:** Þegar þú gerir commit, skrifaðu skýra og lýsandi texta sem útskýrir breytingarnar. Þetta auðveldar öðrum að skilja tilgang breytinganna og fylgjast með framförum.
4. **Notkun Brancha:** Forðastu að vinna beint í main branchinu. Þróarðu nýja eiginleika í sérstökum feature-branchum. Þetta aðskilur þróun og gerir samþættingu öruggari.

## 3 Að stofna GitHub reikning

- Fyrsta skrefið: Stofna GitHub reikning.
- Til að stofna reikning: <https://github.com/join>.
- Til að hlaða niður Github Desktop: <https://desktop.github.com>.

## 4 Notkun á Github Desktop

### 4.1 Cloning repository

- **Bæta við geymslu (.e repository):** Smelltu á 'Add' hnappinn sem finna má efst í vinstra horninu.
- **Clone repository:** Veldu 'Clone repository'. Settu inn URL slóðina á geymsluna sem þú vilt "klóna", t.d. 'https://github.com/matthias1412/VNLI\_19'.
- **Velja stað fyrir repo'ið á tölvunni:** Ákveða hvar á tölvunni þú vilt hafa verkefnið, getur þess vegna verið á Desktopinu. Velur staðinn í "Local path".
- **Vinnsla með repo'ið:** Eftir að "klóna"repo'ið, munu allar breytingar sem gerðar hafa verið af öðrum í þessu repo'i birtast í folder'inum sem þú valdir. Ef þú gerir breytingar og vilt pusha þeim í repo'ið, munu þær breytingar sem þú gerðir í þessu folder'i vera þær sem fara inn í geymsluna.
- **Sjá breytingar:** Allar breytingar sem þú gerir á verkefninu þínu í folderinum þínum geturðu séð undir 'Changes' í GitHub Desktop.

### 4.2 Pulling Changes

Til að uppfæra folderinn þinn með nýjustu breytingum á repo'inu:

1. **Fetch Origin:** Byrja á því að velja "Fetch Origin"takann til að skoða hvort einhverjar nýjar breytingar hafa verið gerðar á repo'ið.
2. **Pull Changes:** Ef það hafa verið breytingar, mun "Fetch Origin"takinn breytast í "Pull". Veldu þennan "Pull"takka til að uppfæra þitt local folder með nýjustu breytingum.

## 4.3 Commit and Push

Nokkuð svipað og að pulla

1. **Breytingar á local folderinum:** Þegar þú hefur gert einhverjar breytingar, sér Github Desktop það, og sýnir með því að sýna þær breytingar sem þú hefur gert undir "Changes". Aug þess mun "Commit changes"takinn lýsa upp.
2. **Bæta við Commit Summary:** Byrjaðu á því að skrifa stutta lýsingu á breytingunum sem þú gerðir inn í "Commit Summary". Ásamt nánari lýsingu í stærri boxinu fyrir neðan. Jafnvel þó þú sért á þínu eigin branchi, er gott að hafa yfirlit hvenær hverju var bætt við.
3. **Commit'a breytingum:** Þegar það er búið að bæta við lýsingu, og ganga úr skugga að breytingarnar sem á að bæta við eru réttar, velurðu "Commit changes"takann. Eftir það ertu samt ekki búinn að pusha gögnunum í repo'ið
4. **Push'a breytingum:** Til að setja breytingarnar endanlega í repo'ið, þarf að ýta á "Push"takann. Þá er ferlið búið og local breytingar þínar eru komnar á GitHub repo'ið.

## 5 Að vinna með branches

1. **Hvað eru Branches?:** Hægt er að horfa á branch sem önnur gerð/týpa af main branchinu. Sem dæmi, ef þú vilt bæta við nýrri virkni, segjum starfsmanna gagnagrunn, þá myndirðu búa til branch/feature-branch út frá main og nefna það t.d Employee-database-branch"
2. **Þróun inn í Branch:** Þú myndir þá vinna í öllum þeim nauðsynlega kóða innan þessarar nýju greinar. Það er algengt/óskrifuð regla að aldrei gera beinar breytingar á main branchið, nema þegar þessi branches eru sameinaðar inn í main.
3. **Engar beinar breytingar á main branchinu:** Þessi nálgun tryggir að main branchið haldist stöðugt og breytingar eru aðeins mergaðar eftir ítarlega þróun og prófun í feature-bröncunum.

### 5.1 Að búa til branch og skipta á milli brancha

1. **Að pulla áður en þú branchar:** Áður en þú stofnar nýtt branch, viltu tryggja að allar nýjustu breytingarnar séu teknar úr main branchinu með því að pulla. Sem tryggir það að þitt branch sé með nýjustu breytingum úr main.
2. **Að búa til nýtt branch:** Til að búa til branch í Github Desktop, velurðu "Current branch"takann. Þar velurðu síðan nafn fyrir nýja branchið þitt. Þetta branch er eins og búa til nýtt main, nema bara fyrir þína vinnu, einangrað frá breytingum sem aðrir gætu verið að gera.
3. **Enginn ruglar í þínu branchi:** Í þínu branchi vinnurðu að þeirri virkni sem þú ert að setja inn sjálfur, enginn annar ruglar eða gerir neitt við þennan kóða.
4. **Að skipta um branches:** Til að skipta á milli brancha, ferðu aftur í "Current branch"valmöguleikann. Sem dæmi ef þú vilt fara aftur í main branchið, velurðu "main"og pullar frá main, það sama gildir um að skipta aftur yfir í þitt branch, eða hvaða branch sem er.

### 5.2 Merging Branches (Git Flow)

1. **Hvað er að merge?:** Að merge er það ferli að samþætta breytingar frá einni grein (t.d feature-branch) yfir í aðra (oftast main) branch. Þetta er mikilvægt að gera þegar þú hefur klárað að gera virkni, eins og starfsmanna gagnagrunnu, og ert tilbúinn/nn að merge við main branchið/verkefnið.
2. **Update'a Feature Branch:** Áður en þú mergað viltu vera viss um að branchið þitt sé uppfært með því að ýta á "Fetch Origin"og pusha/pulla allt sem þarf.

5. **Að merga pull requestið:** Með eitt approval, er hægt að merga pull requestið inn í main. Til að gera það ferðu aftur á Pull request dálkann inn í github.com, og velur "Merge pull request". Þetta mergar feature branchið inn í main branchið endanlega. Virknin er nú kominn inn í verkefnið.

## 7 Að leysa Conflicts

1. **Hvað eru merge conflicts?:** Merge conflict eru algeng þegar þú ert að merga feature-branch inn í main branchið. Conflict gerast þegar sömu línur af kóða hafa verið breyttar í báðum bröncunum eftir að þær voru stofnaðar.
2. **Conflicts í GitHub Desktop:** GitHub Desktop lætur þig vita hvaða skrár eru með merge conflicts. Þessi conflicts þurfa að vera leyst áður en hægt er að endanlega merge'a. (Þ.e eftir að leysa conflictin þarf að merga aftur.)
3. **Að leysa conflicts í VScode:** Opna skrárnar með conflicti í, í VScode. Þar munt þú sjá merki sem segja til um hvar conflictið á sér stað. Merkin eru ««««<"sem þýðir þínar breytingar, -===== "aðskilur breytingarnar og »»»»> eru komandi breytingar (frá branchinu sem þú ert að reyna merga í).
4. **Valmöguleikar til að leysa conflict:** Fyrir hvert conflict, hefurðu þrjá valmöguleika:
  - a) Accept the incoming changes made by others.
  - b) Keep only your changes.
  - c) Combine both sets of changes into one.VScode gefur þér takka til að velja þessa valmöguleika, þetta er bara eitt klikk sem þú þarft til að laga þetta.
5. **Að lokum:** Eftir að þú hefur leyst öll conflict, vistaðu skrárnar. Þá geturðu reynt að merga aftur. Ef öll conflict voru leyst, mergar þetta.

## 8 Algeng vandamál í Git

1. **.gitignore:** Stundum býr tölvan til ósýnilegar skrár sem GitHub reynir að senda (push) inn í repository-ið. Við viljum forðast að hlaða niður gagnslausum skrá. Til að koma í veg fyrir þetta, uppfærum við .gitignore skránnar. Til dæmis, ef GitHub Desktop sér skrá eins og '.\_MACOS', bætirðu línu í .gitignore með '.\_\*', sem segir að allar skrár sem byrja á '.\_' eigi ekki að vera sendar inn í repository-ið.
2. **Að Gleyma að committa á feature-branch:** Þú ættir alltaf að committa breytingar áður en þú skiptir yfir í nýtt branch. Ef þú gerir það ekki, gætu breytingar farið á milli brancha eða jafnvel tapast.
3. **Aldrei vinna í main:** Forðastu að vinna beint í main branchinu. Þú ættir að vinna í þínu feature-branchi og síðan samþætta (merge) það við main. Þetta er mikilvægt vegna þess að aðrir gætu hafa sent breytingar í main áður en þú byrjaðir að vinna í þínu branchi. Þannig tryggirðu að engar breytingar glatist við samþættingu.
4. **halda breytingum innan einnar kröfu:** Reyndu að takmarka commit í feature-branchinu þínu við sem færstar breytingar. Þetta auðveldar öðrum sem fara yfir kóðann þinn og ef upp koma villur, er auðveldara að leiðrétta þær án þess að þurfa að fjarlægja annan kóða sem virkar og á rétt á sér að vera til staðar.

## 9 GitHub í Terminal (Ekki nauðsynlegt að kunna)

1. **Aukið flækjustig:** Notkun GitHub í terminal/command line getur virðst flóknari en býður upp á þægindi, sérstaklega þegar unnið er með mörg branches. Skiljanlegt er ef einhver kys að nota einfaldari leiðir.
2. **Tengjast Repo í Terminal:**
  - a) **Opna Terminal:** Byrjaðu á því að opna terminal á tölvunni þinni.

- b) **Búa til SSH Encryption Key:** Sláðu inn 'ssh-keygen -t ed25519 -C your\_email@gmail.com' til að búa til lykil.
  - c) **Bæta við SSH Key:** Notaðu skipunina 'ssh-add ~/.ssh/id\_ed25519' til að bæta lyklinum við SSH keyring.
  - d) **Afrita SSH Key:**
    - i. Notaðu 'cat ~/.ssh/id\_ed25519.pub' til að sýna opinbera hluta lykilsins.
    - ii. Afritaðu allt sem birtist, eins og 'ssh-ed25519 mjoglangurkoði your\_email@gmail.com'.
  - e) **Bæta SSH Key við GitHub:**
    - i. Farðu inn á GitHub og veldu myndina þína í efra hægra horninu.
    - ii. Undir 'Access' veldu 'SSH and GPG keys' og síðan 'New SSH Key'.
    - iii. Paste'aðu afritaðan SSH lykil þar inn og gefðu því heiti, t.d. 'Local Laptop'.
  - f) **Klóna Repository:** Að lokum, farðu í viðeigandi directory í terminal og notaðu 'git clone git@github.com:matthias1412/VNL1\_19.git' til að klóna repository-ið.
3. **Notkun Git Skipana:** Þegar búið er að tengjast repository-inu í terminal, geturðu notað skipanir eins og 'git pull', 'git push', 'git status', 'git commit', 'git branch', 'git merge', 'git checkout' o.fl. til að stjórna verkefninu.