



# GitHub

## Manual

The Essentials of GitHub for Beginners

## Efnisyfirlit

<b>Grunnatriði og Skipanir</b>	<b>2</b>
Hvað er Github?	2
Uppsetning reikninga	2
Repositories	3
Branches	3
Merge	4
Clone	4
Commit	4
Push	4
Pull	5
Status	5
Log	5
Fetch	5
Add	6
Pull Request	6
Code Review	6
<b>Git Flow</b>	<b>7</b>
<b>GitHub Desktop</b>	<b>9</b>
<b>Algeng vandamál og lausnir</b>	<b>10</b>
Að hætta við breytingar	10
Árekstur við sameiningu (Merge Conflicts):	10
Vandamál í greinum (Branch Issues):	10
Geymsla (Repository) finnst ekki eða aðgangi hafnað:	10
Vandamál við „Pull Requests”	11

# Grunnatriði og Skipanir

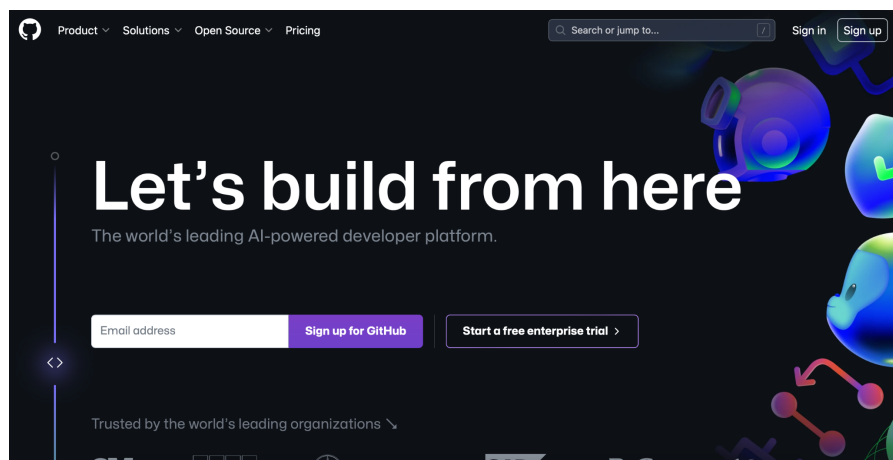
## Hvað er Github?

GitHub, eða <http://github.com>, er notað sem útgáfustýringartól og gerir mörgum kleift að vinna að sama verkefni/kóða í einu. Hægt er að breyta skrá og senda áfram inn á fyrir aðra að sækja og einnig er hægt að rekja þær breytingar sem gerðar hafa verið í skrá. GitHub er gagnlegt tól í hópvinnu þar sem margir eru að vinna að sama verkefni.

Í raun hugsar Git um gögn eins og streymi af ljósmyndum. Í hvert skipti sem þú gerir „commit“ ertu í raun að taka mynd (snapshot) af því hvernig allar skrárnar líta út og geymir vísun í þá ljósmynd.

## Uppsetning reikninga

Til að setja upp reikning á GitHub þarf að byrja á því að finna „Sign-Up“ hnappinn í hægra horninu á vefsíðunni eða að setja inn tölvupóstfangið í gluggan hjá fjólubláa hnappinum.

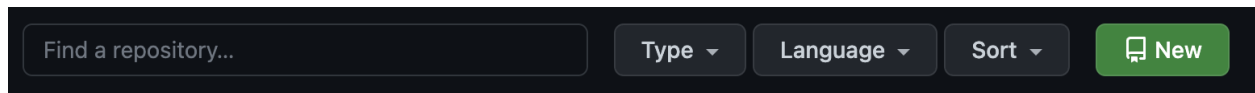


Ef ýtt er á „Sign-Up“ biður síðan um tölvupóstfang og fer með notandann í gegn um uppsetningarferlið. Síðar þarf að staðfesta tölvupóstfangið með því að fara inn á tölvupóstinn sinn og opna link sem verður sendur.

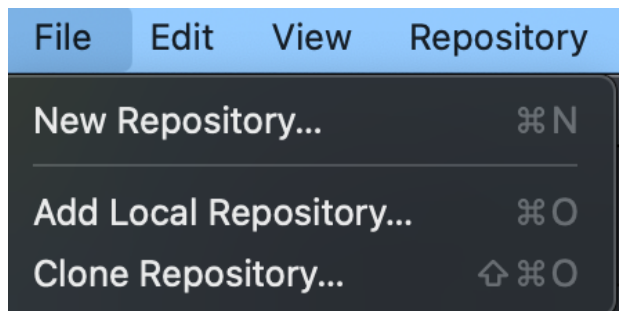
## Repositories

„Repositories” eða „geymslur” eru notaðar til að geyma skrárnar. Hægt er að hugsa um þær sem gagnageymslur. Í þeim eru allar skrárnar sem tengjast verkefninu og einnig hægt að rekja breytingarnar á þeim.

Hægt er að búa til repository á GitHub vefsíðunni:



Í gegnum GitHub Desktop:



eða í command-line:

Búa til directory:	<code>mkdir my_new_project</code>
Færa sig yfir í það directory:	<code>cd my_new_project</code>
Ræsa Git:	<code>git init</code>

## Branches

Í hverri geymslu (repository) eru svokallaðar „branches” eða „greinar”. Notendur geta búið til greinar til að vinna í utan „main” greinarinnar; til dæmis þegar er verið að bæta við einhverjum eiginleikum (features). Þar með er ekki verið að breyta „main” greininni eða aðal forritinu sjálfu.

Búa til nýja grein (branch):	<code>git branch [branch-nafn]</code>
Skipta um grein (branch):	<code>git checkout [branch-nafn]</code>
Til að eyða branch:	<code>git branch -d [branch-nafn]</code>

## Merge

Hægt er að sameina greinar með „merge”. Þar með er hægt að sameina breytingar á einni grein yfir á aðra. Oftast er „merge” notað eftir að „Pull Request” er samþykkt (sjá „Pull Request”).

```
git merge [branch-nafn]
```

## Clone

Að „klóna” í Git þýðir í raun að afrita geymslu (repository) inn á þitt svæði svo þú getir unnið í skránum í geymslunni á þinni tölvu.

```
git clone [URL]
```

## Commit

„Commit” er eins og skjáskot af skránum í geymslunni á ákveðnum tímapunkti. Það skráir breytingar ásamt höfundum og einnig er hægt að bæta við skilaboðum með. Hér er -m notað til að láta skilaboð fylgja með, og stendur fyrir „message”.

```
git commit -m "skilaboð"
```

## Push

Sendir þínar breytingar áfram frá þínu svæði (tölvu) í sameiginlegu geymsluna. Fyrst þarf að gera „commit” (sjá Commit).

```
git push
```

## Pull

Notað til að sækja og samræma breytingar frá sameiginlegu geymslunni yfir á þitt svæði (tölvu).

```
git pull
```

## Status

Sýnir þér núverandi stöðu (status) á skránum. Sýnir þér á hvaða grein (branch) þú ert staddur, ef að þú ert samstilltur sameiginlegu geymslunni, hvaða skrár eru „tracked” eða „untracked”. Ef að skrár eru „tracked” eru þær annaðhvort, committed, modified eða staged.

```
git status
```

## Log

Setur upp söguna af öllum „commits” sem hafa verið gerð í geymslunni.

```
git log
```

## Fetch

Hleður niður einungis nýjustu gögnunum í þína geymslu án þess að sameina það við það sem þú ert að vinna í.

```
git fetch
```

## Add

Þegar þú breytir skrá þá fara þær breytingar ekki sjálfvirkt inn með næsta „commit”, heldur verður að gera „add” til að „stage”-a þær breytingar sem eiga að fara inn með næsta „commit”.

Bæta við einni skrá:	<code>git add nafnskráar.txt</code>
Bæta við mörgum skrá:	<code>git add skrá1.txt skrá2.txt skrá3.txt</code>
Bæta við öllum breytingum:	<code>git add .</code>
Bæta við öllum breytingum fyrir alla geymsluna (repository):	<code>git add -A</code>

## Pull Request

Leið til að leggja til breytinga á geymslu (repository) á GitHub. Þegar „Pull Requests” er notað er hægt að fara yfir kóðann og samþykkja áður en breytingar eru sameinaðar (merge). Eftir að „push” er notað til að breyta einhverju á grein (branch) er hægt að opna „Pull Request” til að koma af stað samræðum um breytingarnar.

## Code Review

Hluti af „Pull Request” (sjá „Pull Request”). Meðlimir í hóp fara yfir, setja athugasemdir við og koma með tillögur að því hvernig hægt sé að bæta tillagðar breytingar. Hægt er að setja athugasemdir við einstaka línur í kóðanum, samþykkja breytingar eða biðja um frekari lagfæringar áður en það er sameinað (merge) main.

# Git Flow

Git Flow er „branching strategy” fyrir Git. Það gerir hópi kleift að hafa ákveðinn ramma utan um stærri verkefni og vinna á skipulagðan hátt. Greinunum (branches) er skipt niður í ákveðna hópa.

## Flæðið á GitFlow er svona:

1. Þróunargrein (Develop Branch) er búin til frá main greininni.  
`git branch develop`  
`git push -u origin develop`
2. Útgáfugrein (Release Branch) er búin til frá þróun.  
`git checkout develop`  
`git checkout -b release/0.1.0`
3. Eiginleikagrein (Feature Branch) er búin til frá þróun.  
`git checkout develop`  
`git checkout -b feature_branch`
4. Þegar eiginleiki (feature) er tilbúinn er hægt að sameina (merge) við þróunargreinina.  
`git checkout develop`  
`git merge feature_branch`
5. Þegar útgáfugrein er tilbúin er hægt að sameina (merge) við þróun og main.  
`git checkout main`  
`git merge release/0.1.0`
6. Ef að vandamál kemur upp í main er hægt að búa til hotfix grein út frá main.  
`git checkout main`  
`git checkout -b hotfix_branch`



7. Þegar hotfix er tilbúið er hægt að sameina (merge) það við bæði þróun og main.

`git checkout main`

`git merge hotfix_branch`

`git checkout develop`

`git merge hotfix_branch`

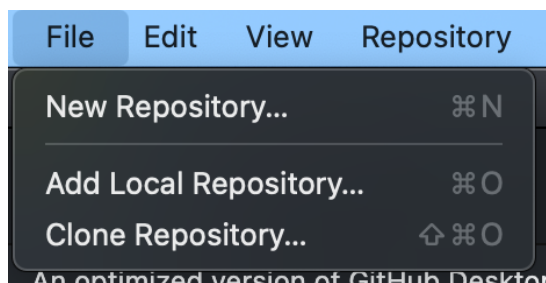
`git branch -D hotfix_branch`

# GitHub Desktop

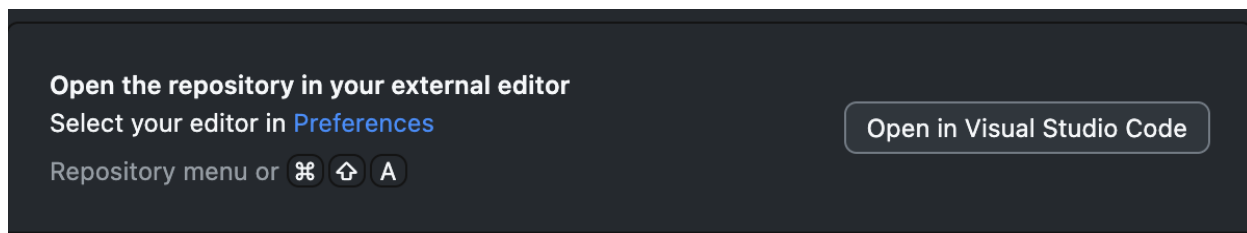
GitHub Desktop er viðmót sem einfaldar Git ferlið töluvert. Venjulega fara aðgerðir eins og pull, push, commit og fleira fram í command-line en með GitHub Desktop er hægt er að hafa umsjón yfir geymslunum (repositories) og senda áfram skipanir án þess að þurfa að nota command-line, sem gerir GitHub Desktop mjög gott fyrir byrjendur.

Hægt er að nálgast GitHub Desktop með því að fara inn á <https://desktop.github.com/>.

Þegar uppsetningu á því er lokið er hægt að tengja GitHub aðganginn sinn við GitHub Desktop. Hægt er að búa til nýjar geymslur og klóna geymslur.



Einnig er hægt að opna geymsluna beint þaðan með VS Code.



GitHub Desktop kemur líka með tillögur að því hvað ætti að gera næst, eins og til dæmis hér:



# Algeng vandamál og lausnir

## Að hætta við breytingar

- Ef þú vilt ekki halda breytingunum sem þú gerðir er hægt að nota:  
git checkout –
- Til að hætta við breytingar í ákveðinni skrá:  
git checkout – [skrá-nafn]
- Til að hætta við allar breytingar sem voru gerðar í directory:  
git checkout –.

## Árekstur við sameiningu (Merge Conflicts):

- Gerist þegar breytingar í mismunandi greinum (branches) stangast á við hvor aðra.
- Lausn: „Pull” síðustu breytingu frá sameiginlegu geymslunni, leysa hnúta handvirkt með text editor, og svo „commit” breytingarnar.

git pull origin main  
git commit

## Vandamál í greinum (Branch Issues):

- Vandí við að breyta eða samhæfa greinar.
- Lausn: Reglulega að sækja („pull”) breytingar frá main greininni og að nota lýsandi nöfn fyrir greinarnar.

## Geymsla (Repository) finnst ekki eða aðgangi hafnað:

- Vitlaus URL fyrir geymslu eða aðgangsheimild ekki rétt stillt.

- Lausn: Finna rétt URL fyrir geymsluna og passa að aðgangsheimildir séu rétt stilltar fyrir alla í hópnum.

## Vandamál við „Pull Requests”

- Árekstrar, sameining tekst ekki eða er ekki samþykkt.
- Lausn: Áður en „Pull Request” er gert, sækja seinustu breytingar frá main greininni, passa upp á að greinin sé uppfærð, og leysa hnútana.

`git pull origin main`