

Extra

- [VSC en Git](#)

VSC en Git

Voor je examen moet je in staat zijn om een versie controle systeem te gebruiken. Met een versie controle systeem kun je makkelijk terug naar een oude versie van je code.

Ben je dus bezig met een nieuwe opdracht en je hele code werkt niet meer, dan ga je gewoon terug naar de vorige versie. Hoe dit werkt wordt heit uitgelegd.

Tip: framework -> gebruik Git!

Gebruik je **React**, **Yii**, **Laravel** of een ander **framework** dan is het aan te raden om Git te gebruiken.

Hiermee zorg je dat je telkens terug kan naar de vorige werkende versie.

Hoe installeer je Git en stel je alles goed in, in VSC?

Hieronder wordt dit in 4 stappen uitgelegd.

Stap 1 Git voor Windows

Je hebt [Git voor Windows](#) geïnstalleerd? Nee, doe dat dan eerst. Tijdens de installatie worden veel dingen gevraagd, als je het niet weet dan kun je gewoon alle standaard opties aan laten staan.

Stap 2 configureer Git voor Windows

Ga naar een **terminal in VCS** en stel je naam en email in voor het gebruik van Git hub.

```
git config --global user.name "Issrae Marqui"
git config --global user.email "issramarqui@student.com"
```

Stap 3 Initialize Repository

Open een project in VSC.

Klik op het derde icoontje aan de rechterkant (met die drie bolletjes) en druk dan op *Initialize Repository*.

[image.1674318159168.jpg](#)

Stap 4 Commit

Na de init moet je je project files "*commiten*". Je maakt als het ware een versie (waar je naar terug kan).



Je ziet in dit voorbeeld dat er twee bestanden zijn gewijzigd.

Vul een korte 'titel' bij Message in voor deze versie en druk op commit.

Je bent nu klaar.

Elke keer als je je code hebt aangepast en alles werkt, doe je een commit. Je kunt dan altijd terug.

Wijzigingen bekijken

Stel je hebt een bestand gewijzigd en je wilt zien wat er is gewijzigd.

Installeer hiervoor een plug-in:



Druk op het versie beheer-icoontje (rechts, derde van boven). Kies dan het bestand waarvan je de wijzigingen wilt zien. Je ziet nu de vorige versie en de nieuwe versie (die nog niet is ge-commit) naast elkaar en de wijzigingen zijn aangegeven.

Met het ronde pijltje achter de bestandsnaam kan je alle wijzigingen in dit bestand in één keer terug draaien.

Versies vergelijken

Wil je oudere wijzigingen zien? Dat kan ook druk dan op het rondje boven de message met het klokje erin.



Je ziet dan rechts alle wijzigingen. Je kunt op ene versie klikken en dan kan je zien hoe die versie er uit zag.

Terug naar een versie

Stel je wilt de **file** index.php terug zetten.

git checkout versionid file_name

Wil je **alles** terug zetten gebruik dan:

```
git checkout versionid
```

en dan weer **terug** naar de laatste versie.

```
git checkout -
```

--