

Dodona

# Dodona oefeningen maken



UNIVERSITEIT  
GENT

**Brecht Willems**

[Brecht.Willems@UGent.be](mailto:Brecht.Willems@UGent.be)

Onderwijsbegeleider aan de UGent

# Inhoudsopgave

[ Klikken op onderstaande titels verwijst je door naar het overeenkomstige deel in dit document. ]

<b>1</b>	<b>Inleiding</b> .....	<b>5</b>
<b>2</b>	<b>Keuze tussen online of offline editor</b> .....	<b>6</b>
2.1	Online editor .....	6
2.2	Offline editor .....	6
<b>3</b>	<b>Installatie vooraf</b> .....	<b>6</b>
3.1	Chrome .....	6
3.2	Python interpreter .....	6
3.3	VS Code downloaden en installeren .....	7
3.3.1	<i>VS Code downloaden en installeren voor Windows</i> .....	7
3.3.2	<i>VS Code downloaden en installeren voor Mac</i> .....	8
3.4	Automatisch aanbevolen extensies downloaden en instellingen en sneltoetsen configureren .....	8
3.4.1	<i>Instellen met Welcome scherm</i> .....	8
3.4.2	<i>Wat is er automatisch ingesteld?</i> .....	9
3.5	Dodona extensie configureren .....	10
3.5.1	<i>Registeren in Dodona</i> .....	10
3.5.2	<i>API-token genereren</i> .....	10
3.5.3	<i>Koppelen met VS Code</i> .....	11
<b>4</b>	<b>Repl.it (optioneel)</b> .....	<b>12</b>
<b>5</b>	<b>Papyros</b> .....	<b>12</b>
<b>6</b>	<b>Aan de slag met VS Code</b> .....	<b>13</b>
6.1	VS Code interface overzicht .....	13
6.2	Map/folder openen .....	13
6.3	Bestanden en mappen aanmaken in een workspace .....	14
6.3.1	<i>Bestand aanmaken (.py, .html, .css, .sql ...)</i> .....	14
6.3.2	<i>Map aanmaken</i> .....	15
<b>7</b>	<b>Van start met Dodona als lesgever</b> .....	<b>16</b>
7.1	Lesgeversrechten aanmaken .....	16
7.2	Cursus en reeks aanmaken .....	16
7.3	Oefening toevoegen .....	17
<b>8</b>	<b>Van start met GitHub</b> .....	<b>18</b>
8.1	GitHub account aanmaken .....	18
8.2	GitHub repository aanmaken .....	18
8.3	Repository toevoegen aan Dodona .....	18
8.4	Dodona toegang geven tot repository .....	19
8.5	Webhook instellen .....	19
8.6	Lesmateriaal aanmaken .....	20

<b>9</b>	<b>GitHub Desktop</b> .....	<b>21</b>
9.1	Installatie en configuratie .....	21
9.2	Bestanden opladen naar GitHub .....	22
<b>10</b>	<b>Github.dev (optioneel)</b> .....	<b>24</b>
10.1	Github.dev openen .....	24
10.2	Bestanden opladen naar github.dev .....	24
<b>11</b>	<b>Repository structuur</b> .....	<b>25</b>
11.1	Oefeningmap-structuur .....	25
11.2	Oefeningconfiguratie (config.json of dirconfig.json) .....	26
<b>12</b>	<b>Judges</b> .....	<b>27</b>
12.1	Python.....	27
12.1.1	<i>Kenmerken</i> .....	27
12.1.2	<i>OuputJudge</i> .....	27
12.1.2.1	Voorbeeld: echo.....	27
12.1.3	<i>DoctestJudge</i> .....	27
12.1.3.1	Voorbeeld: echo-function-print.....	27
12.2	Turtle.....	27
12.2.1	<i>Kenmerken</i> .....	27
12.3	HTML/CSS .....	28
12.3.1	<i>Kenmerken</i> .....	28
12.3.2	<i>Nieuwe oefeningen</i> .....	28
12.3.3	<i>Voorbeelden</i> .....	28
12.4	SQL.....	28
12.4.1	<i>Kenmerken</i> .....	28
12.4.2	<i>Features</i> .....	28
12.4.3	<i>Nieuwe SQL-oefeningen</i> .....	28
12.4.4	<i>Voorbeelden</i> .....	29
<b>13</b>	<b>Dodona in VS Code</b> .....	<b>30</b>
13.1	Leesactiviteiten in VS Code.....	30
13.2	Dodona oefening aanmaken in VS Code .....	30
13.3	Dodona oefening indienen via VS Code .....	30
13.4	Dodona Tips .....	31
13.5	Overige functionaliteit in VScode .....	31
<b>14</b>	<b>GitHub in VS Code (optioneel)</b> .....	<b>32</b>
14.1	Configuratie .....	32
14.2	Bestanden opladen naar GitHub .....	33
<b>15</b>	<b>Python code uitvoeren, inlezen van tekst en debuggen (Optioneel)</b> .....	<b>35</b>
15.1	Python code uitvoeren in Terminal .....	35
15.2	Informatie inlezen met Python via Terminal .....	36

15.3	Python code debuggen (optioneel)	36
<b>16</b>	<b>HTML bestand openen in Chrome</b>	<b>38</b>
<b>17</b>	<b>Databanken in VS Code (Optioneel)</b>	<b>39</b>
17.1	Eerste query uitvoeren	39
17.2	SQL script maken en uitvoeren	39
17.3	Databank oefeningen in Dodona	40
<b>18</b>	<b>VS Code sneltoets (Optioneel)</b>	<b>41</b>
18.1	Command Palette openen	41
18.2	Sneltoetsen	41
18.2.1	<i>Belangrijkste sneltoetsen</i>	41
18.2.2	<i>Overige sneltoetsen</i>	42
18.3	Alle sneltoetsen	42
<b>19</b>	<b>Extensies (optioneel)</b>	<b>43</b>
19.1	Extensie manueel installeren	43
19.2	Geïnstalleerde extensies weergeven	44
19.3	Specifieke instellingen voor een extensies weergeven	44
<b>20</b>	<b>Instellingen en sneltoetsen manueel aanpassen (Optioneel)</b>	<b>45</b>
20.1	Instellingen openen en aanpassen	45
20.1.1	<i>Aanbevolen instellingen</i>	45
20.2	Sneltoets aanpassen/toevoegen	46
20.2.1	<i>Aanbevolen sneltoetsen</i>	46
20.3	Kleurthema aanpassen	47
<b>21</b>	<b>Python in Interactive Window (Optioneel)</b>	<b>48</b>
21.1	Python code uitvoeren in Interactive Window	48
21.2	Informatie inlezen met Python via Interactive Window	49
<b>22</b>	<b>Foutoplossing</b>	<b>50</b>
22.1	Al mijn instellingen zijn plots verdwenen?!	50
<b>23</b>	<b>Bronnen</b>	<b>50</b>

[ Gebruik indien nodig de sneltoets Ctr+F om te zoeken in het document.]

# 1 Inleiding

De **drempel** om zelf nieuwe oefeningen te maken voor Dodona is te groot voor sommige leerkrachten. Naast de tijdsinvestering, zijn er redelijk wat installatie- en configuratiestappen die doorlopen moeten worden, die niet altijd even makkelijk zijn.

Deze handleiding hoopt hierop een antwoord te bieden door alle nodige stappen uit te schrijven om te starten met het maken van nieuwe oefeningen. Zo kan je oefeningen maken die afgestemd zijn op de **leefwereld van de leerling** en op de **inhoud van het vak**. Nieuwe oefeningen kunnen ook gebruikt worden tijdens een **test** of **examen**.

Eenmaal alles goed staat, gaat het maken van nieuwe oefeningen relatief vlot.

Voor deze workshop wordt [Visual Studio Code](#) gebruikt. Visual Studio Code, **meestal afgekort tot VS Code**, wordt ontwikkeld door Microsoft en is gedeeltelijk open source. [PyCharm](#) (gratis voor studenten/leerkrachten) van JetBrains is ook een optie. Het is zwaarder/krachtiger, maar wel iets minder gebruiksvriendelijk.

Ook voor leerlingen/studenten geniet VS Code de voorkeur, omdat dat programma veel meer ondersteuning en mogelijkheden (zoals testen) biedt tijdens het programmeren in vergelijking met de online editor van Dodona. De website **Repl.it** kan echter ook goed van pas komen om kleine programma's online te testen (zie verder).

Hieronder lees je hoe je **VS Code** zelf op je eigen computer kunt installeren, en hoe je die kunt configureren om het werken met VS Code nog makkelijker te maken. Op het einde van dit document geven we ook nog enkele handige tips.

In onderstaande handleiding wordt telkens met rode kaders aangegeven welke knoppen er aangeklikt moeten worden of welke tekstvelden er ingevuld moeten worden.

Laat u niet afschrikken door de lengte van deze handleiding. Na sectie 12 Judges, vanaf pagina 30 staat er minder relevante achtergrondinformatie die soms ook van pas kan komen.

*Dodona wordt door het [Dodona team](#) (prof. Bart Mesuere, Charlotte Van Petegem en prof. Peter Dawyndt) en de UGent **gratis** aangeboden aan scholen en onderwijsinstellingen.*

[ Gebruik indien nodig de sneltoets Ctr+L+F om te zoeken in het document.]

## 2 Keuze tussen online of offline editor

### 2.1 Online editor

Ik zou aanraden om eerst de handleiding op <https://docs.dodona.be/nl/guides/exercises/creating-exercises/introduction/> te doorlopen. Mocht je merken dat je toch liever een offline editor gebruikt met meer mogelijkheden, dan kan je de rest van deze handleiding doorlopen.

### 2.2 Offline editor

Ik gok dat je toch 20 minuten moet rekenen tegen alles goed staat. Als je van plan bent om veel/intensief oefeningen te maken, zou ik de offline manier aanraden.

## 3 Installatie vooraf

Indien de installatie verschillend is voor Windows en Mac zal dit in rood aangegeven worden.

### 3.1 Chrome

Om HTML5-webpagina's te bekijken gebruik je [Chrome](#). Installeer Google Chrome mocht dat nog niet het geval zijn.

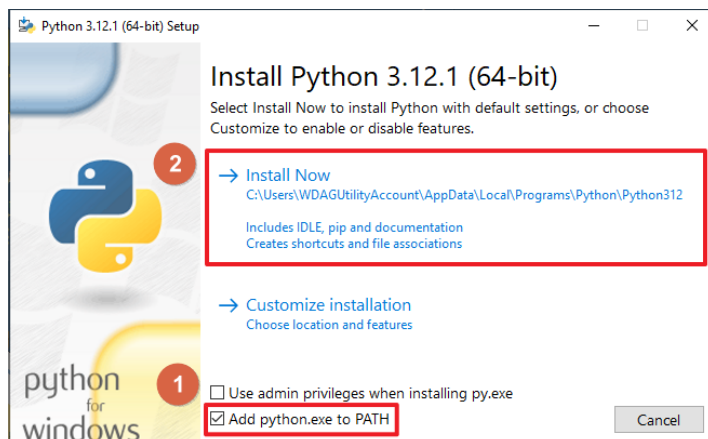
### 3.2 Python interpreter

Een interpreter is letterlijk een programma dat jouw code zal lezen, interpreteren en uitvoeren. Zonder een interpreter kan jouw Python code dus niet uitgevoerd worden! Volg onderstaande stappen indien je nog geen recente Python interpreter hebt.

*Voor de handleiding kiezen we de laatste stabiele versie van Python. Dit is nu 3.12.1. In deze handleiding kunnen er nog screenshots staan van een oudere versie, maar de stappen die je moet doorlopen zijn hetzelfde.*



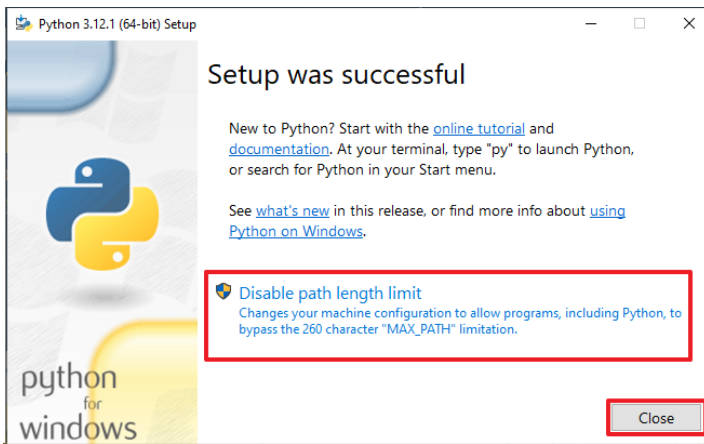
- <https://www.python.org/downloads/>
- **Download Python 3.12.1** (besturingssysteem staat normaal automatisch juist)
- Directe link voor Windows: <https://www.python.org/ftp/python/3.12.1/python-3.12.1-amd64.exe>
- Ergens opslaan en het bestand dat werd gedownload uitvoeren.



**Enkel voor Windows:** Deze stap is belangrijk en moet tijdens de installatie uitgevoerd worden, anders moet je het achteraf manueel Python toevoegen aan het PATH.

→ **"Add python.exe to PATH" aanvinken**

- **Install Now**
- Wilt u toestaan dat deze app wijzigingen aan uw apparaat aanbrengt? **Ja**
- Wachten (± 1 minuut)



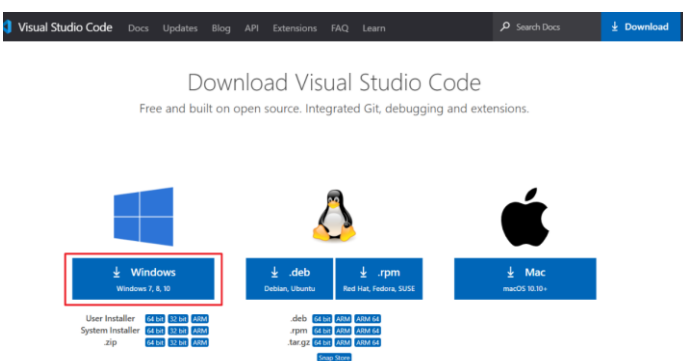
Indien je de melding krijgt:

- **Disable path length limit**
- Wilt u toestaan dat deze app wijzigingen aan uw apparaat aanbrengt? **Ja**
- **Close**

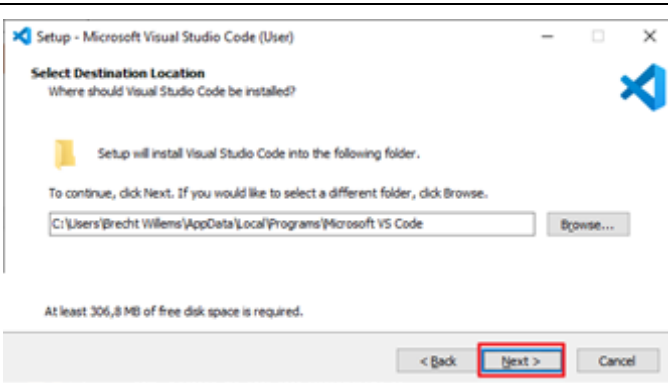
### 3.3 VS Code downloaden en installeren

Visual Studio Code (meestal afgekort tot VS Code) is een programmeeromgeving die wordt ontwikkeld door Microsoft. De installatie is wat verschillend voor Windows en Mac.

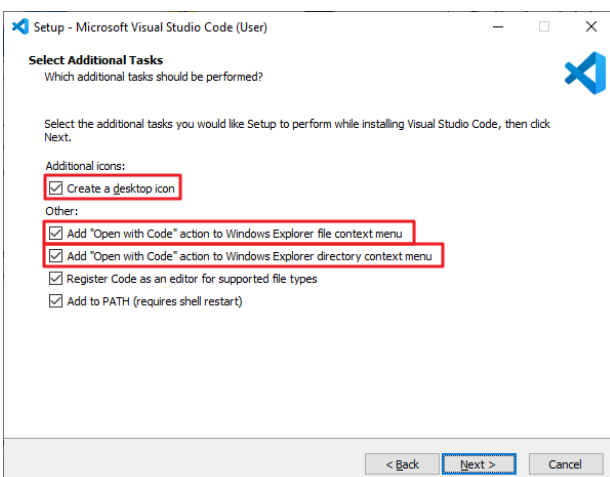
#### 3.3.1 VS Code downloaden en installeren voor Windows



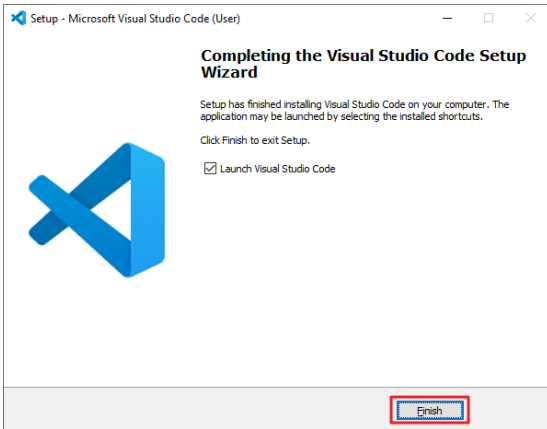
- <https://code.visualstudio.com/Download>
- Kies de eerste kolom of gebruik de directe link voor **Windows User Installer (64-bit)**:  
<https://aka.ms/win32-x64-user-stable>
- Bestand ergens opslaan
- Bestand dat werd gedownload (.exe) openen/uitvoeren
- Wachten



- English > **OK**
- **I accept the agreement > Next >**
- Select Destination Location > **Next >**
- Select Start Menu Folder > **Next >**

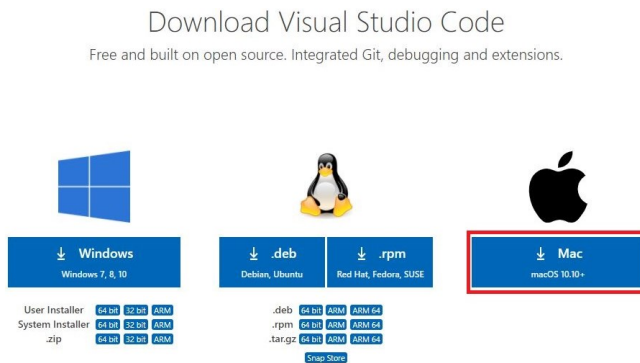


- Alles aanvinken:
- **Create a desktop icon** aanvinken
  - **Add "Open with Code" action to Windows Explorer file context menu** aanvinken
  - **Add "Open with Code" action to Windows Explorer directory context menu** aanvinken
  - **Add to PATH (requires shell restart)** aanvinken mocht dat nog niet zo zijn
  - **Next >**
  - **Install**
  - Wachten (± 1 minuut)



→ Finish

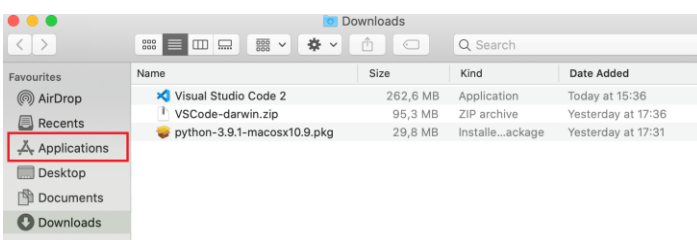
### 3.3.2 VS Code downloaden en installeren voor Mac



→ <https://code.visualstudio.com/Download>

→ Kies de derde kolom – dan wordt het bestand **VSCoDe-darwin.zip** gedownload

→ Wachten



→ In je Downloads – map staat **VSCoDe-darwin.zip**

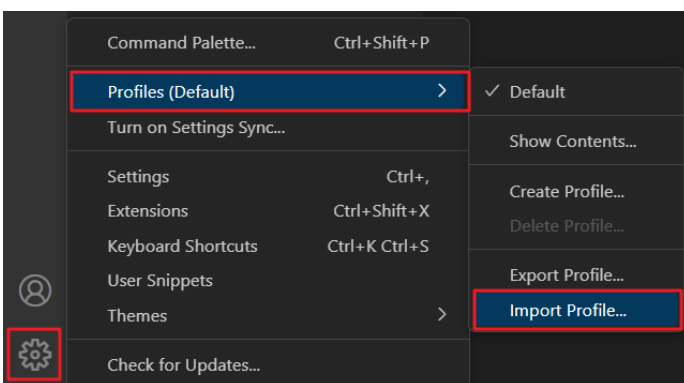
→ Dit bestand wordt ook automatisch unzipped als **Visual Studio Code 2** – indien dit niet gebeurt is klik je op de .zip

→ Versleep nu **Visual Studio Code 2** naar **Applications**

## 3.4 Automatisch aanbevolen extensies downloaden en instellingen en sneltoetsen configureren

In VS Code kan je meerdere profielen gebruiken. Een profiel bevat een aantal instellingen. De instellingen van een goede configuratie zijn beschikbaar in een GitHub gist. Die kan je eenvoudig downloaden als UGent-profiel waardoor alle instellingen onmiddellijk goed staan. Je kan op elk moment terug wisselen naar het default profiel. **Sluit deze stap niet over, anders moet je zelf veel zaken manueel instellen.**

### 3.4.1 Instellen met Welcome scherm

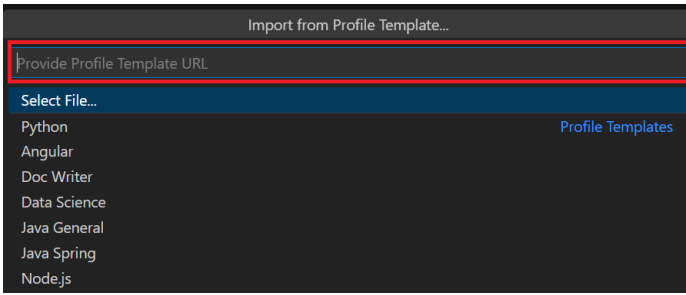


→  (linksonder)

→ Profiles (Default)

→ Import Profile...

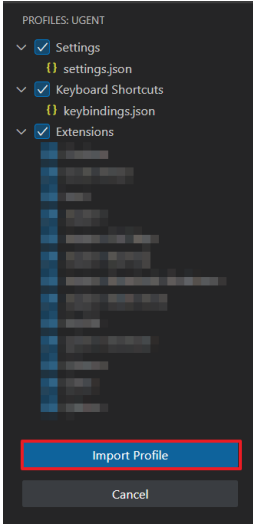




Plak deze link in het tekstvak dat bovenaan verschijnt (rood omkaderd):

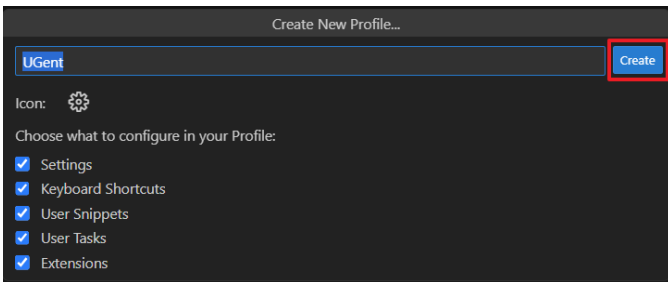
<https://gist.github.com/BTWS2/7ee259a4202e78e6ecd2fbd7b64a3891>

→ Klik **Import from URL** of druk op **Enter**



Hier krijg je een overzicht van de instellingen, sneltoetsen en extensies die geïmporteerd zullen worden.

→ **Import Profile**

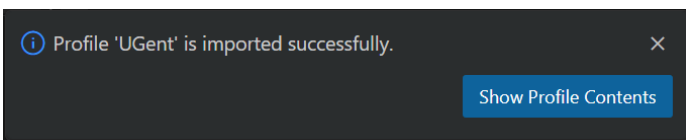


→ **Create**

→ (Eventueel VS Code afsluiten en opnieuw opstarten)

→ ... Wachten ...

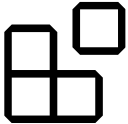

Extensies worden nu automatisch gedownload en geïnstalleerd. Aanbevolen instellingen en sneltoetsen worden automatisch goed ingesteld.



Je krijgt een melding dat het UGent-profiel succesvol geïmporteerd is. Linksonderaan VS Code zou er nu ook "UG" moeten staan bij het tandwiel icoon, om aan te duiden dat je in het UGent-profiel aan het werken bent. Als je "UG" niet ziet staan, werk je in het Default profiel van VS Code. Bekijk [Al mijn instellingen zijn plots verdwenen?! om dit in orde te brengen.](#)

### 3.4.2 Wat is er automatisch ingesteld?

Onderstaande extensies werden automatisch geïnstalleerd door het importeren van het UGent-profiel (in [Geïnstalleerde extensies weergeven](#) vind je hoe je met @enabled de geïnstalleerde extensies kan weergeven).

	Logo	Naam en link	Uitleg
		<a href="#">Python</a> (developer: Microsoft)	<b>Noodzakelijk</b> om Python te kunnen gebruiken in VS Code. Deze extensie bevat automatisch Pylance.

	<b>Black Formatter</b> (developer: Microsoft)	<b>Aanbevolen</b> om Python code goed uit te lijnen.
	<b>Jupyter</b> (developer: Microsoft)	Interactieve manier om Python code te schrijven ( <b>optioneel</b> ).
	<b>Open in Browser</b> (developer: TechER)	Website snel openen in de browser ( <b>optioneel</b> ).
	<b>MySQL</b> (developer: cweijan)	Verbinden met een databank ( <b>noodzakelijk</b> voor SQL-oefeningen)
	<b>Dodona</b> (developer: Pieter De Clercq – Stijn De Clercq)	Deze extensie laat toe om heel gemakkelijk Python oefeningen in te dienen rechtevreeks vanuit VS Code ( <b>sterk aanbevolen</b> ).
	<b>Excel Viewer</b> (developer: GrapeCity)	CSV-bestanden bekijken in tabelvorm met deze knop ( <b>optioneel</b> ). 

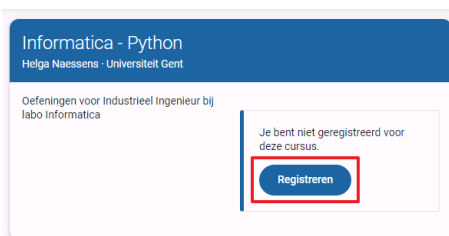
Optioneel: De instellingen en sneltoetsen die automatisch werden ingesteld kan je ook manueel instellen, zie [Instellingen en sneltoetsen manueel aanpassen](#). Hoe je manueel een extensie kan installeren wordt in [Extensie manueel installeren](#) beschreven.

## 3.5 Dodona extensie configureren

De oefeningen worden via Dodona opgegeven en moeten daar ook ingediend worden. Met de [Dodona extensie](#) kan dit eenvoudig en snel in Visual Studio Code. Configureer nu deze extensie in drie stappen.

### 3.5.1 Registeren in Dodona

[Dodona](#) is een online programmeeromgeving ontwikkeld aan de UGent die automatische feedback genereert voor student en docent. De correctheid van de ingediende code wordt bepaald door de uitvoer van testen op de ingediende code en de verwachte uitvoer te vergelijken. Bovendien bevat Dodona een geïntegreerde debugger, waarmee je stap voor stap door je code kan gaan.



- ➔ <https://dodona.ugent.be/nl/courses>
- ➔ **Alle cursussen**
- ➔ Typen in het zoekvak (lesgever Brecht Willems)
- ➔ **Registreren**
- ➔ **Inloggen met onderwijsinstellings account**

### 3.5.2 API-token genereren

Om de Dodona-extensie te kunnen gebruiken, moet een API token aangemaakt worden op de Dodona website. Via dit API-token kan VS Code in jouw naam de code uploaden en indienen voor feedback (zonder dat je je wachtwoord moet delen). Om een token aan te maken volg je onderstaand stappenplan.



→ <https://dodona.ugent.be/profile>

of

→ <https://dodona.ugent.be/>

→ Dropdown onder jouw Voornaam

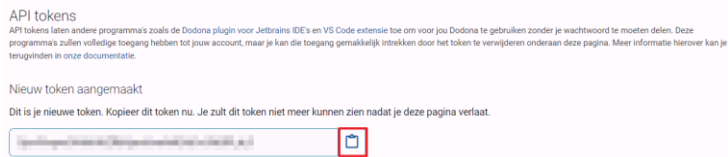
→ Mijn profiel




→ Scroll naar onder

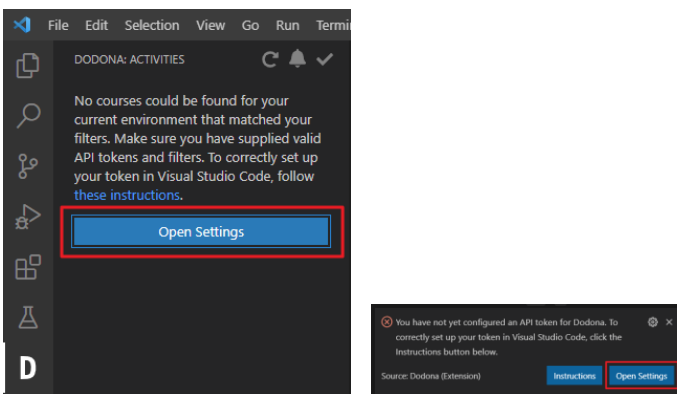
→ Kies een naam (maakt niet uit welke)

→ +-knop



→  Kopieer de gegenereerde token

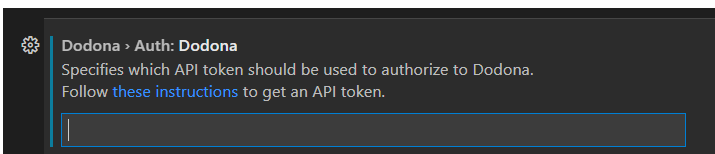
### 3.5.3 Koppelen met VS Code



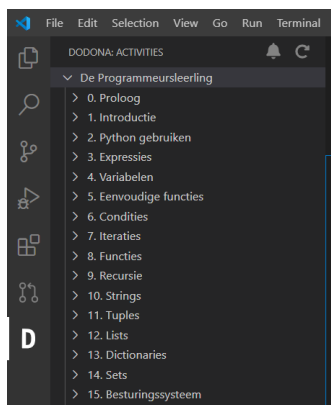
→ **D** (links in activity bar halverwege) of via melding rechtsonder

→ **Open Settings**

Indien je het D-symbool van Dodona niet ziet in de linkerkantlijn, moet je de Dodona extensie nog installeren. Door op deze webpagina [Dodona](#) op de groene **Install** knop te drukken, opent installatiepagina van de extensie in VS Code. Klik op de blauwe **Install** knop in VS Code om te extensie te installeren.



→ **Plak je API token in het tekstvak**



In de linkerkolom krijg je nu een overzicht van de oefeningensreeksen van de cursus (eventueel nog uitklappen)

Je kan nu Dodona in je browser sluiten. Als je een nieuwe token wil, moet je dezelfde stappen doorlopen (en eventueel de oude verwijderen want er kan geen twee keer dezelfde naam gebruikt worden).

## 4 Repl.it (optioneel)

Repl.it is een online tool die volledig draait in de browser. Er is dus geen installatie nodig en kan zinvol zijn voor kleine scripts en oefeningen of wanneer je problemen ondervindt met VS Code.

1. Surf naar [repl.it](https://repl.it) en maak een account aan.
2. Je ontvangt een bevestigingsmail in je mailbox.
3. Log in op [repl.it](https://repl.it) en maak een nieuwe replit met **+ New repl** (linksboven)
4. Vul de gevraagde gegevens in. Kies voor de taal **Python** (niet Python 2.7).
5. In het linkse venster kan je code intikken, vb. `print("Hello world!")`
6. Klik op de **Run**-knop (groene knop bovenaan). In het rechtse venster kies je **Console** om de uitvoer te zien.



## 5 Papyros


UGent heeft een alternatief voor Repl.it gemaakt, genaamd Papyros: een online basic editor:








<https://papyros.dodona.be/>

## 6 Aan de slag met VS Code

### 6.1 VS Code interface overzicht

Aan de linkerkant van het scherm staat de **Activity Bar**. Je kan ook gebruik maken van sneltoetsen (*shortcuts*) (zie derde kolom):

Voor Windows  +  + en voor Mac  +  +

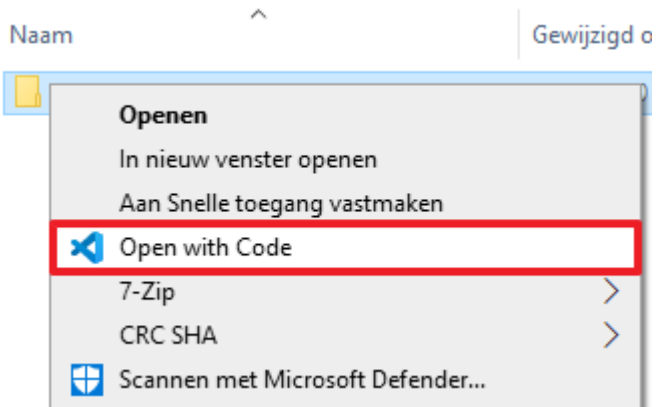
	<b>File Explorer:</b> Overzicht van bestanden en mappen.	Ctrl+Shift+E ( <b>Mac:</b> Command+Shift+E)
	<b>Search:</b> Zoeken (en vervangen) in alle bestanden in de geopende map.	Ctrl+Shift+F ( <b>Mac:</b> Command+Shift+F)
	<b>Run and debug</b>	Ctrl+Shift+D ( <b>Mac:</b> Command+Shift+D)
	<b>Source Control</b>	Ctrl+Shift+G ( <b>Mac:</b> Command+Shift+G)
	<b>Testing</b>	Niet gebruikt
	<b>Extensions:</b> Installeer en beheer je extensies.	Ctrl+Shift+X ( <b>Mac:</b> Command+Shift+X)
<b>D</b>	<b>Dodona:</b> overzicht oefeningen Python.	Geen sneltoets
	<b>Databank:</b> connecteer met een databank.	Geen sneltoets

Onderaan het scherm staat de **Status Bar**, waar je (v.l.n.r.) de programmeertaal, het aantal errors en warnings, het huidige lijnnummer en meldingen kan vinden.

### 6.2 Map/folder openen

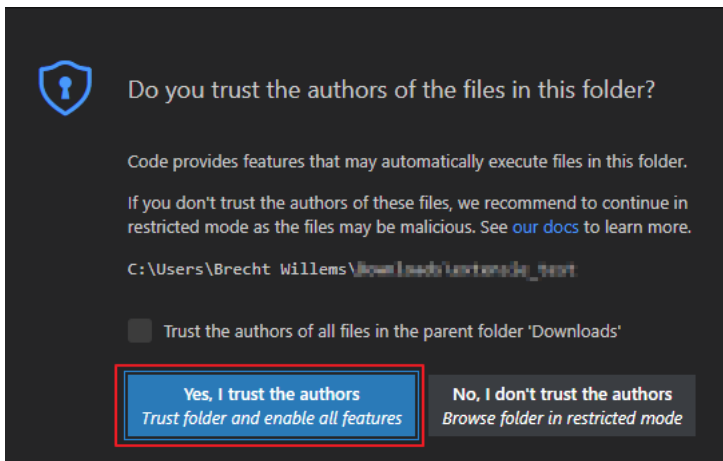
Om een overzicht te bewaren is het best om een nieuwe map/folder (in Windows Verkenner of Mac Finder) te maken. Maak hierin eventueel aparte mappen voor de verschillende programmeertalen (HTML\_CSS/Python/SQL). Start VS Code op vanuit de map waarin je wilt werken.

**VS moet iedere keer vanuit een map geopend worden, dit is de **workspace** waarbinnen VS Code zal werken.**



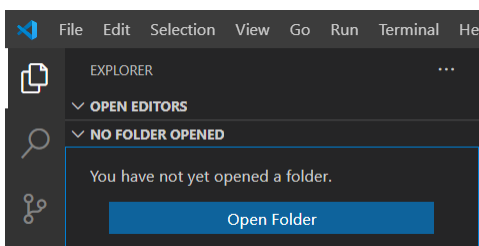
- ➔ Open Windows Verkenner (of Mac Finder)
- ➔ Maak een nieuwe map met als naam bv. **dodona\_oefeningen**
- ➔ Rechtsklik op die map
- ➔ Open with Code

VSCode start op en toont alle bestanden/mappen in die map.



- ➔ Do you trust the authors of the files in this folder?

**Yes, I trust the authors**



Indien er geen folder werd geopend, gebruik dan:



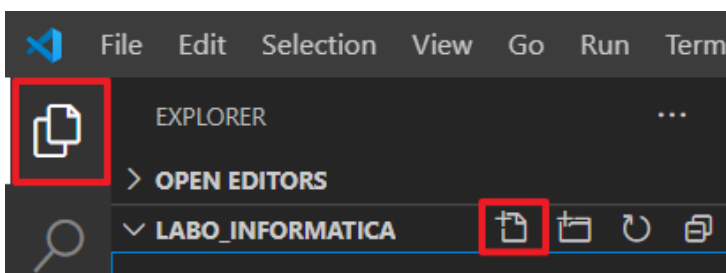
## 6.3 Bestanden en mappen aanmaken in een workspace


Open **File Explorer** (Ctrl+Shift+E). Hier vind je informatie over **OPEN EDITORS** (bestanden die je momenteel geopend hebt en eventueel aan het bewerken bent) en de inhoud van de **geopende map** (bv. **DODONA\_OEFENINGEN**) waarin je werkt. De geopende hoofdmap is nu jouw **workspace**, die waarschijnlijk nog helemaal leeg is.



### 6.3.1 Bestand aanmaken (.py, .html, .css, .sql ...)

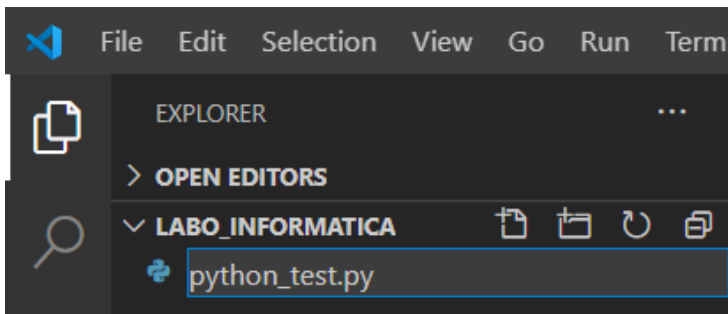
In de map kan je bestanden toevoegen



- ➔  (New File icoon naast mapnaam)

of

- ➔ Rechtsklikken (onder de mapnaam)
- ➔ **New File**



→ Kies een bestandsnaam met de juiste extensie

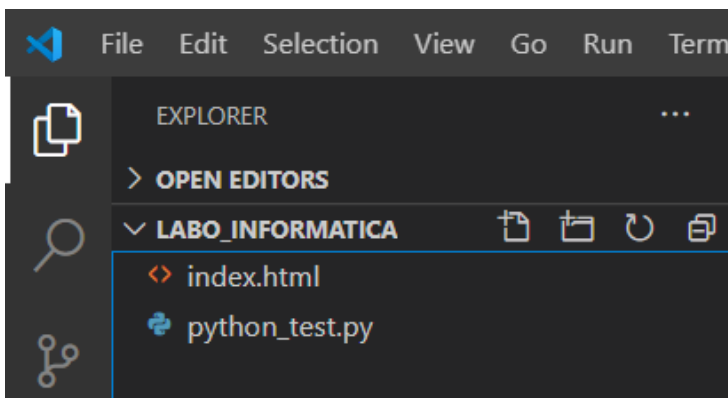
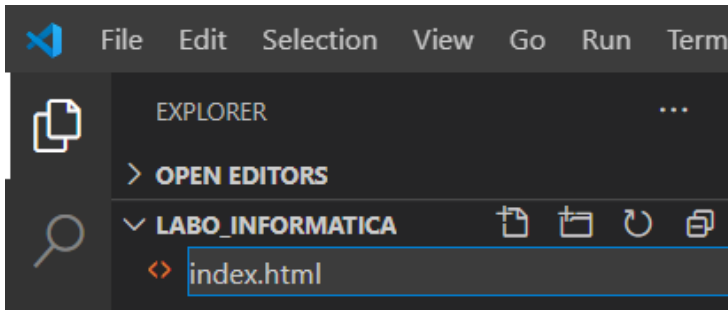
→ Voeg zelf de juiste extensie toe:

- Python File (.py)
- HTML File (.html)
- Stylesheet (.css)
- SQL File (.sql)
- Tekstbestand (.txt)
- CSV-bestand (.csv)

→ Enter

Het bestand wordt nu aangemaakt in de workspace

Later kan je op dezelfde manier een tekstbestand toevoegen in die map.



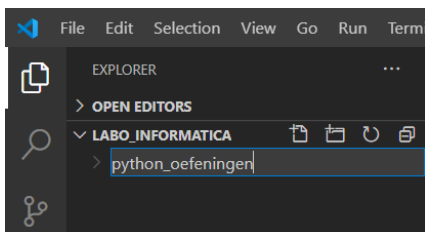
→ Maak twee bestanden: **python\_test.py** en **index.html**


**Tip: Kies altijd een naam ZONDER spaties.**

Eventuele spaties vervang je door liggende streepjes.

### 6.3.2 Map aanmaken

Je kan ook een map toevoegen aan de workspace:



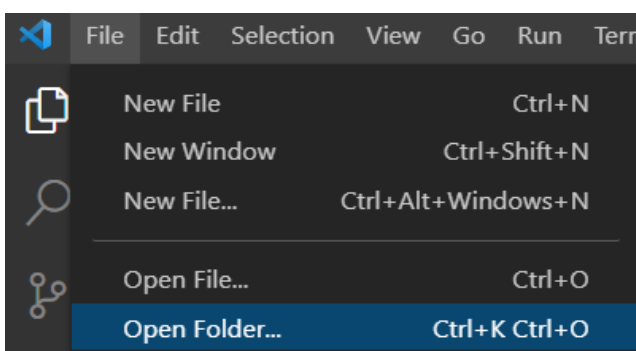
→  (New Folder icoon naast mapnaam)

→ Kies een mapnaam

→ Enter

De map worden nu aangemaakt in de workspace.

Wil je Python-oefeningen in die map plaatsen, dan wijzig je best de workspace (zie hieronder)



Indien je wil werken in een andere folder, open die folder dan met *File / Open Folder* (Ctrl+K Ctrl+O) .

Enkel dan wordt de **workspace** ook correct geopend.

## 7 Van start met Dodona als lesgever

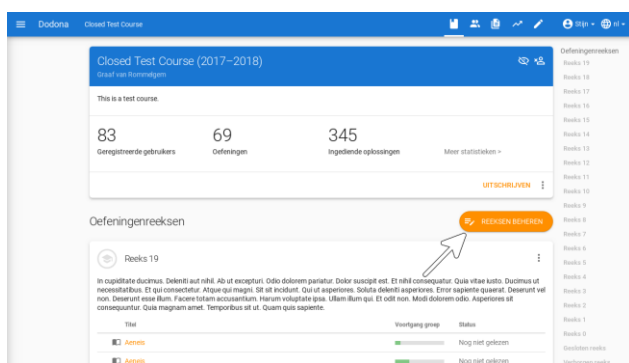
Op <https://docs.dodona.be/nl/> vind je documentatie van Dodona als naslagwerk. De meeste informatie van hieronder vind je ook daar terug. Hieronder is het eerder een gedetailleerd stappenplan uitgewerkt specifiek voor VS Code in combinatie met GitHub.

### 7.1 Lesgeversrechten aanmaken

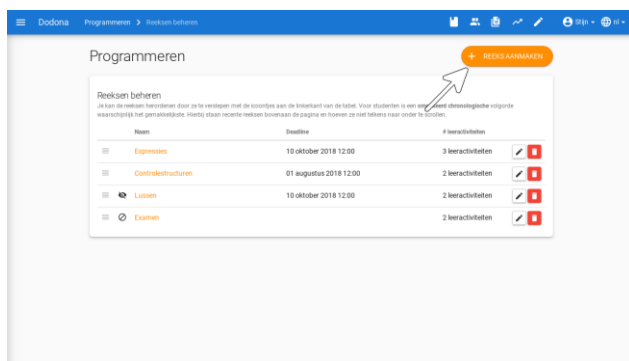
Om zelf oefeningen te kunnen toevoegen, moet je eerst lesgeversrechten aanvragen. Vul dit formulier in [https://dodona.ugent.be/nl/rights\\_requests/new](https://dodona.ugent.be/nl/rights_requests/new).

### 7.2 Cursus en reeks aanmaken

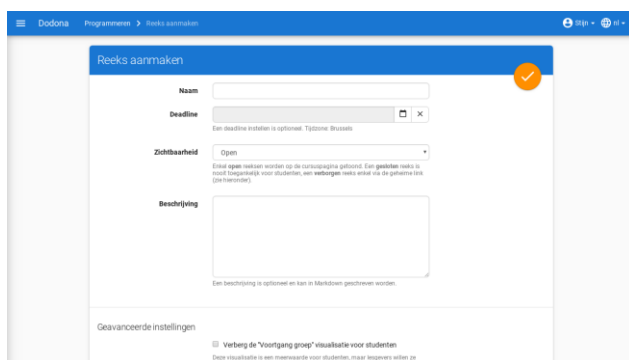
Maak hier een nieuwe lege cursus aan: <https://dodona.ugent.be/nl/courses/new/>. Een overzicht van alle cursussen staat hier <https://dodona.ugent.be/nl/courses/>. Daarna moet er nog een reeks aangemaakt worden, waar later de oefeningen in opgenomen worden.



➔ Reeksen beheren



➔ Reeks aanmaken



➔ Naam invullen

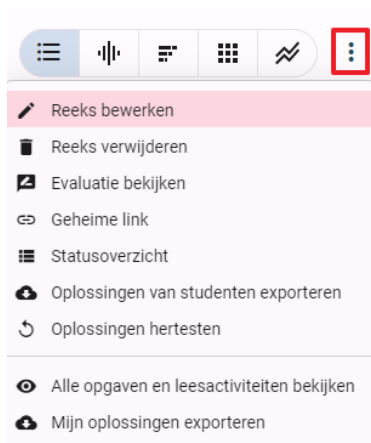
➔ Vinkje rechtsboven

Meer informatie: <https://docs.dodona.be/nl/guides/teachers/exercise-series-management/>



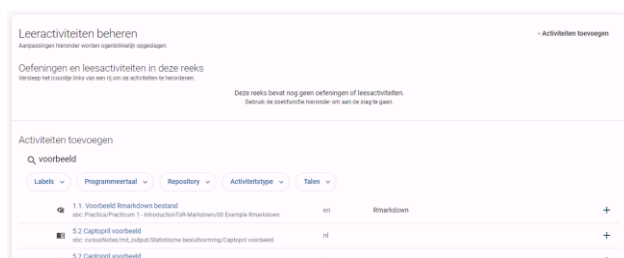
## 7.3 Oefening toevoegen

In Dodona kan je zowel eigen oefeningen als oefeningen van andere lesgevers gebruiken in je cursus.



→ Drie verticale puntjes

→ Reeks bewerken



→ Scroll naar beneden naar de onderste helft.

→ Zoek naar een oefening op basis van de titel. Dit kan een bestaande Dodona-oefening zijn of een eigen nieuwe oefening. Vind je het niet onmiddellijk? Filter dan op basis van de naam van je repository.

→ +-icoon aan rechterkant om de oefening toe te voegen aan de reeks.

## 8 Van start met GitHub

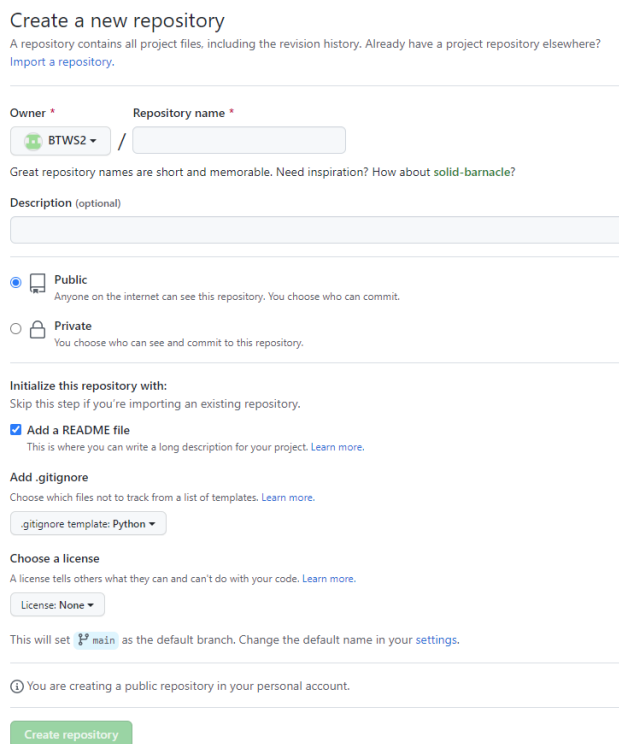
Dodona gebruikt git repositories om oefeningen en ander lesmateriaal te beheren. Een repository kan meerdere oefeningen bevatten en moet aan een bepaalde structuur (zie verder) voldoen om gelezen te kunnen worden door Dodona. De officiële documentatie vind je op [https://docs.dodona.be/nl/guides/teachers/new-exercise-repo/#\\_1-een-git-repository-aanmaken](https://docs.dodona.be/nl/guides/teachers/new-exercise-repo/#_1-een-git-repository-aanmaken).

### 8.1 GitHub account aanmaken

De eerste stap is inloggen/registreren op GitHub via <https://github.com/>.

### 8.2 GitHub repository aanmaken

Maak een GitHub repository aan. Je kan dit eenvoudig doen via <https://github.com/new>.



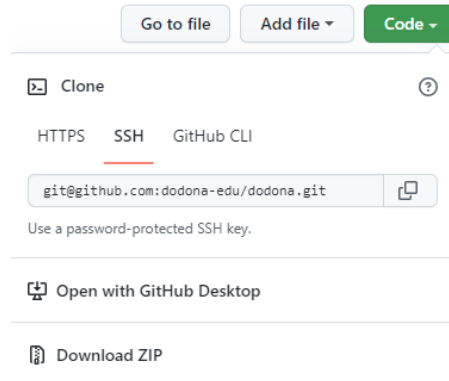
The screenshot shows the GitHub 'Create a new repository' page. It includes fields for 'Owner' (set to BTWS2), 'Repository name', and 'Description (optional)'. There are radio buttons for 'Public' (selected) and 'Private'. Under 'Initialize this repository with:', there is a checked box for 'Add a README file'. The '.gitignore' section has a dropdown menu set to 'Python'. The 'Choose a license' section has a dropdown menu set to 'None'. At the bottom, there is a green 'Create repository' button.

- ➔ Kies een **Repository name**
- ➔ Wil je dat iedereen de inhoud van je repository kan zien of enkel jij? Kies dan voor **Public of Private**.
- ➔ **Add a README file** aanvinken
- ➔ **.gitignore template: Python** Dit zorgt er voor dat enkel de relevante files gesynchroniseerd worden.
- ➔ **License** mag je laten staan op **None**.
- ➔ **Create repository**

### 8.3 Repository toevoegen aan Dodona

Als je lesgeversrechten op Dodona hebt, dan verschijnt in de linker navigatiebalk (te herkennen aan de drie strepen linksboven) een link **Repository's** die je naar een [overzicht van alle repositories op Dodona](#) brengt. Om een nieuwe repository toe te voegen, klik je op de grote plus-knop rechts bovenaan de pagina.

- ➔ Kies een unieke **Repository name**, bij voorkeur dezelfde naam als de repository in GitHub.
- ➔ Clone URL:
  - **Code** knop op GitHub repository
  - **SSH** tabblad
  - **git@...** kopiëren en plakken bij “Clone URL in Dodona”

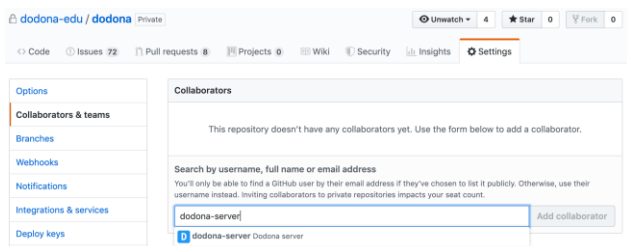


- ➔ Standaard judge: zal meestal **Python** zijn.
- ➔ **Vinkje** rechtsboven

## 8.4 Dodona toegang geven tot repository

Je moet er voor zorgen dat de Dodona server zowel lees- als schrijfrechten heeft op je repository met oefeningen. Dodona heeft deze schrijftoegang nodig om eenvoudig de oefeningen kunnen bewerken via de webinterface.

Hiervoor moet je de gebruiker **dodona-server** aan je repository toe te voegen. Van zodra het Dodona team deze uitnodiging accepteert, kan je van start gaan.



- ➔ **Settings** tab rechtsboven in GitHub repo
- ➔ Collaborators
- ➔ Add people
- ➔ **dodona-server**
- ➔ Add **dodona-server** to this repository

## 8.5 Webhook instellen

Om je oefeningen up-to-date te houden, moet Dodona weten wanneer je wijzigingen aanbrengt. We doen dit met behulp van een *webhook*. Dit is een unieke URL die er voor zorgt dat Dodona je repository opnieuw zal verwerken. Deze URL zou zichtbaar moeten zijn nadat je je repository aan Dodona hebt toegevoegd.

**Repository:** we-programmeren

**Clone URL:** git@github.ugent.be:pythia/programmeren.git

**Path:** we-programmeren

**Default judge :** pythia

**# Exercises:** 732

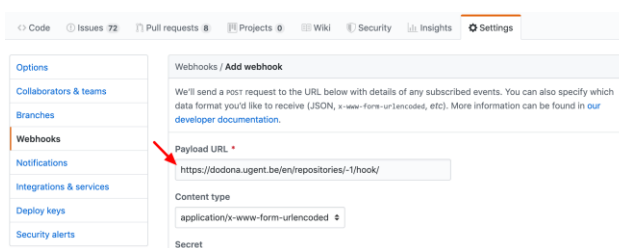
[Reprocess all exercises](#)

## Webhook

If you set up a **webhook** on GitHub, Dodona will automatically update your exercises on each commit:

`https://dodona.ugent.be/en/repositories/1/hook/`

Deze URL moet je vervolgens toevoegen aan GitHub via onderstaande stappen.



- ➔ **Settings** tab rechtsboven in GitHub repo
- ➔ Webhooks (links in navigatiebalk)
- ➔ Add webhook (rechtsboven)
- ➔ **Payload URL:** Geef de **URL** in die je van Dodona gekregen hebt.
- ➔ De andere instellingen mag je zo laten staan.
- ➔ **Add webhook**

## 8.6 Lesmateriaal aanmaken

Je bent nu helemaal klaar om oefeningen en lesmateriaal toe te voegen aan je repository. Als alles goed gaat, dan zouden ze automatisch moeten verschijnen op Dodona. Denk er aan om de oefeningen ook nog toe te voegen aan een reeks in je cursus (zie 7.3 [Oefening toevoegen](#)).

## 9 GitHub Desktop

Wat je met GitHub Desktop kan, kan je ook rechsreeks met VS Code. GitHub Desktop is alleen gebruiksvriendelijker.

### 9.1 Installatie en configuratie



- Download GitHub Desktop  
<https://desktop.github.com/>
- Bestand ergens opslaan
- Bestand dat werd gedownload (.exe) openen/uitvoeren
- Wachten

#### Welcome to GitHub Desktop

Your browser will redirect you back to GitHub Desktop once you've signed in. If your browser asks for your permission to launch GitHub Desktop please allow it to.

[Sign in to GitHub.com](#) [Cancel](#)

#### Welcome to GitHub Desktop

- Sign in to GitHub.com
- This site is trying to open GitHubDesktop > Open

#### Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

- Use my GitHub account name and email address
- Configure manually

Name

B. Willems

Email

[Redacted]

[Finish](#) [Cancel](#)

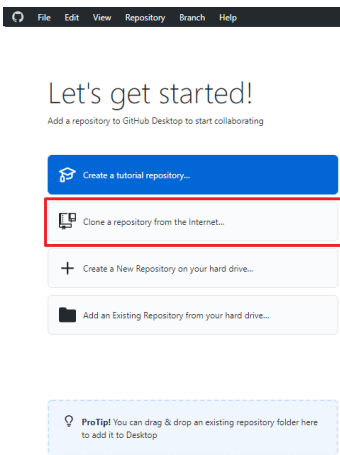
Example commit

**Fix all the things**

B. Willems • 30 minutes ago

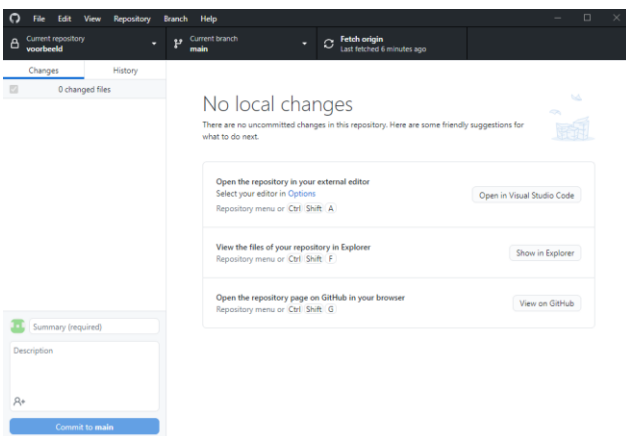
#### Configure Git

- Use my GitHub account name and email address
- Finish



- ➔ Clone a repository from the Internet...
- ➔ Zoek naar je eigen repository
- ➔ Kies waar het geplaatst moet worden, meestal is dit  
C:\Users\jouw\_gebruikersnaam\Documents\GitHub\jouw\_repo\_naam
- ➔ **Clone**

Deze instelling zitten ook onder “File > Clone repository...” linksbovenaan in het menu.

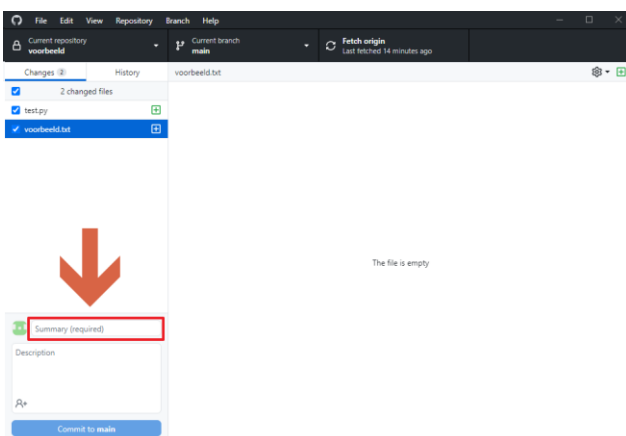


Nu zijn er 3 opties.

- ➔ **Open in Visual Studio Code:** VS Code gebruiken we om wijzigingen aan te brengen. GitHub Desktop op de lokale bestanden te synchroniseren met de online bestanden.
  - Do you trust the authors of the files in this folder > Yes, I trust the authors
- ➔ **Show in Explorer:** Uiteindelijk is dat gewoon een map met achterliggende versiegeschiedenis, die je dus ook kan openen in Windows verkenner.
- ➔ **Show on Github:** Dit is handig om te controleren of de wijzigingen goed zijn doorgekomen naar github.com

Deze instellingen zitten ook onder “Repository” bovenaan in het menu.

## 9.2 Bestanden opladen naar GitHub

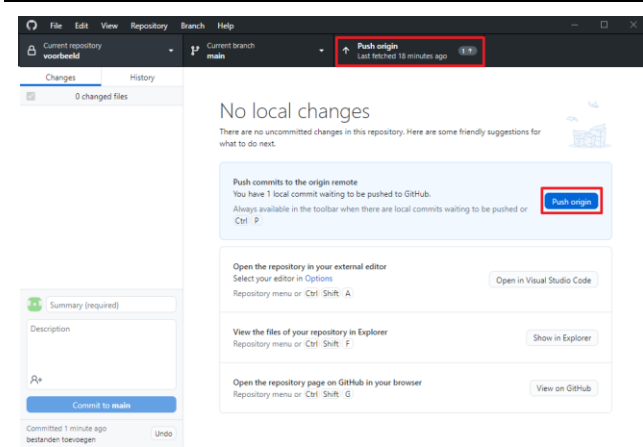


- ➔ Voeg een bestand toe (bijvoorbeeld test.txt) of wijzig de inhoud van een bestand.
- ➔ **Open GitHub Desktop.** Hier krijg je een overzicht van wat er gewijzigd is.
- ➔ Voeg verplicht een eigen bericht toe die kort zegt wat er veranderd is in **Summary**. Dit is puur voor jezelf om achteraf makkelijk dingen terug te vinden.

*Standaard worden alle wijzigingen doorgegeven, wat meestal de bedoeling is.*

- ➔ **Commit to main**

Nu staan de wijzigingen klaar om te versturen naar github.com. Het versturen is de volgende stap.



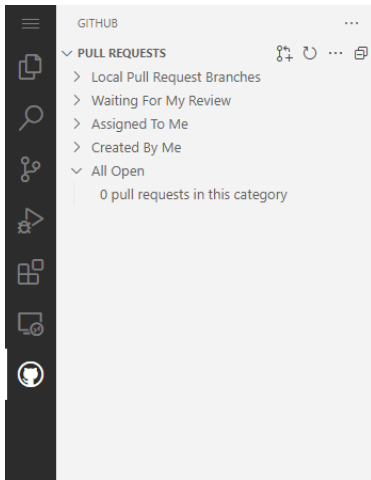
➔ **Push origin** (vrij vertaald: “omhoog duwen naar de cloud”)

➔ **View on GitHub.** Nu zou je je wijzigingen online moeten zien.

## 10 Github.dev (optioneel)

De webgebaseerde editor github.dev is een editor die volledig in de browser wordt uitgevoerd. De editor is heel gelijkaardig aan VS Code, maar biedt minder mogelijkheden. De bestanden die op github.com staan kan je weergeven, wijzigen en “comitten” via github.dev.

### 10.1 Github.dev openen



Er zijn twee manieren om github.dev te openen.

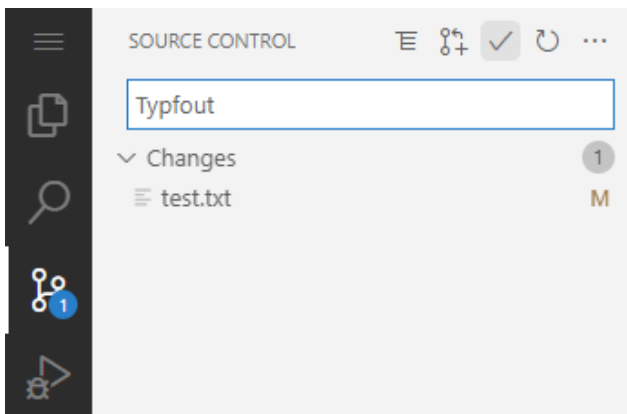
→ Druk op **Shift** en daarna op de de **;**-toets. Je typt dus een punt terwijl de GitHub repository zichtbaar is in je browser.

of


→ Verander **.com** door **.dev** in de URL in de adresbalk. Bijvoorbeeld, deze repository <https://github.com/BTWS2/voorbeeld> wordt <https://github.dev/BTWS2/voorbeeld>

→ Wachten

### 10.2 Bestanden opladen naar github.dev



→ Voeg een bestand toe (bijvoorbeeld test.txt) of wijzig de inhoud van een bestand.

→ **Klik op het  logo.** Hier krijg je een overzicht van wat er gewijzigd is.

→ Voeg verplicht een eigen bericht toe die kort zegt wat er veranderd is in **Message**. Dit is puur voor jezelf om achteraf makkelijk dingen terug te vinden.

*Standaard worden alle wijzigingen doorgegeven, wat meestal de bedoeling is.*

→ ✓ of **Ctrl+Enter** in het tekstvak

Nu staan de wijzigingen op github.com.



# 11 Repository structuur

Het eenvoudigst is als je dezelfde structuur overneemt als een gelijkaardige oefening op <https://github.com/dodona-edu/example-exercises>. Je maakt een map per programmeertaal en zet daarin je oefeningen.

In deze handleiding zal ik enkel de meest relevante items bespreken.

## 11.1 Oefeningmap-structuur

Binnenin een oefeningmap behandelt Dodona elke map met een `config.json`-bestand als een aparte leeractiviteit: dit kan een programmeeroefening zijn of een leesactiviteit. We verwachten dat deze map de volgende structuur heeft:

- **Een `config.json`-bestand:** dit bestand bevat de [oefening-specifieke configuratie](#). Deze configuratie zal samengevoegd worden met alle `dirconfig.json`-bestanden in de bovenliggende map van de oefeningmap. Je kan altijd configuratiewaarden die ingesteld werden door een bovenliggende map overschrijven met het `config.json`-bestand.
- **Een `description-map`:** deze map bevat de volgende bestanden die de oefening beschrijven:
  - **Een `description.en.md`- en/of `description.nl.md`-bestand:** deze bestanden bevatten de Engelse en/of Nederlandse beschrijving van de oefening.
  - **Een optionele `media-map`:** deze map bevat statische bestanden zoals afbeeldingen die gebruikt worden in de oefeningbeschrijving.
  - **Een optionele `boilerplate-map`:** deze map bevat de bestanden `boilerplate.en`, `boilerplate.nl`, en/of `boilerplate` (zonder extensie). De inhoud van deze bestanden worden automatisch ingeladen in het inzendingsvak van de gebruikers. Je kan deze bestanden gebruiken om startcode of structuur te voorzien voor de studenten.
  - **Een `evaluation-map`:** de inhoud van deze map wordt beschikbaar gesteld voor de judge en kan bijvoorbeeld bestanden met de testcode bevatten.
  - **Een optionele `solution-map`:** bestanden in deze map zullen getoond worden op de oefening-informatiepagina als voorbeeldoplossing. Meerdere voorbeeldoplossingen zijn mogelijk, maar bestanden met een naam beginnend met `solution` zullen vooraan staan.

Een voorbeeld van de mappenstructuur voor Python oefeningen met de OutputJudge. Meer informatie:

<https://docs.dodona.be/nl/references/exercise-directory-structure/>

```
+- python
| +- getallen_optellen
| | +- description
| | | +- description.nl.md
| | | +- media
| | | | +- some_image.png
| | | +- boilerplate
| | | | +- boilerplate
| | +- evaluation
| | | +- 0.in
| | | +- 0.out
| | | +- 1.in
| | | +- 1.out
| | +- solution
| | | +- solution.py
| | +- config.json
+- dirconfig.json
+- README.md
| :
| :
```

We kunnen oefeningen groeperen per programmeertaal  
Elke oefening zit in een map en heeft een korte naam

De opgave van de oefening in het Nederlands

Een afbeelding die enkel gebruikt wordt in de beschrijving van deze oefening

Tekst/Code wordt automatisch ingeladen in het indieningsvak van de gebruikers.

Invoer eerste tab in Dodona  
Verwachte uitvoer en instellingen voor tweede tab  
Invoer tweede tab in Dodona  
Verwachte uitvoer en instellingen voor tweede tab

Bestanden in deze map worden enkel voor leerkrachten getoond op de oefening-informatiepagina  
Modeloplossing van leerkracht  
Configuratie voor deze oefening  
Gedeeld configuratiebestand voor alle oefeningen in kindmappen  
Beschrijft de repository

In Dodona kan je ook leesactiviteiten aanmaken. De structuur is heel gelijkaardig als die van oefeningen. Meer informatie: <https://docs.dodona.be/nl/references/exercise-directory-structure/>

## 11.2 Oefeningconfiguratie (config.json of dirconfig.json)

Dodona laat toe om de configuratie van een **oefening** of een **leesactiviteit** in te stellen door middel van configuratiebestanden. Deze bestanden moeten het JSON-formaat hebben en `config.json` genoemd worden in oefeningenmappen en `dirconfig.json` in andere mappen. Om de finale configuratiewaarden te bekomen voor een leeractiviteit, voegt Dodona het standaardconfiguratiebestand samen met de `dirconfigs` in de bovenliggende folders van de oefening en met het oefeningconfiguratiebestand. Dit proces laat je toe om waarden in een bovenliggende folder te overschrijven.

Dit zullen we behandelen samen met een oefening. Eén keer het goed staat voor één oefening kan je het `config.json` bestand kopiëren en enkel de naam aanpassen.

Meer informatie: <https://docs.dodona.be/nl/references/exercise-config/>

Aanbevolen **dirconfig.json** (vul de meest gebruikte programmeertaal in bij “programming language” die in die (onderliggende) map(pen) voorkomt).

```
{
  "access": "public",
  "programming_language": "python",
  "contact": "firstname_lastname@ugent.be"
}
```

Aanbevolen **config.json** (programmeertaal aanpassen indien nodig)

```
{
  "type": "exercise",
  "programming_language": "python",
  "description": {
    "names": {
      "nl": "Voorbeeld oefening",
      "en": "Example exercise"
    }
  },
  "labels": ["label1", "label2"],
  "access": "public",
  "contact": "firstname_lastname@ugent.be"
}
```

# 12 Judges

## 12.1 Python

Documentatie: <https://docs.dodona.be/nl/references/python-judge/>

### 12.1.1 Kenmerken

- Zeer veel mogelijkheden
- Verwachte invoer en uitvoer moet je op voorhand in repository zetten (neemt veel tijd in beslag)
- Twee manieren van vergelijken
  - Deze **OutputJudge** is enkel geschikt voor oefeningen die input vragen via **input()** en de resultaten uitprinten via **print()**.
  - De **DoctestJudge** is een judge die **functies** in de ingediende broncode evalueert functies door er een serie unit tests op uit te voeren die beschreven worden aan de hand van een uitgebreide versie van het format van de Python **doctest** module. De DoctestJudge is enkel geschikt om Python functies te testen.

Beide manieren gebruiken .in- en .out-bestanden die zich in de **evaluation** map bevinden, maar op een andere manier. Er wordt altijd gestart met het cijfer 0 als bestandsnaam. Per extra tab die wil verkrijgen, wordt dat getal met 1 vermeerderd.

### 12.1.2 OuputJudge

De invoer wordt in het .in-bestand opgenomen. De uitvoer in 0.out. De instellingen staan ook in het .out-bestand onder een rij van minstens 3 koppeltekens.

#### 12.1.2.1 Voorbeeld: echo

Bekijk deze oefening aandachtig, neem de structuur over en verander de inhoud van de bestanden waar nodig. Maak bijvoorbeeld een oefening waarin twee getallen opgeteld worden en de resulterende som geprint moet worden.

<https://github.com/dodona-edu/example-exercises/tree/master/python/echo>

### 12.1.3 DoctestJudge

Het oproepen van de functie en het resultaat die de functie print of teruggeeft wordt er onder tegen de linker kantlijn genoteerd. Het oproepen van de functie wordt voorafgegaan door drie groter-dan-tekens (>>>).

Testen worden geschreven in het Python doctest formaat. Bijvoorbeeld

```
>>> getallen_optellen(3, 2)
5
```

Enkel de instellingen staan in het .out-bestand onder een rij van minstens 3 koppeltekens.

#### 12.1.3.1 Voorbeeld: echo-function-print

Bekijk deze oefening aandachtig, neem de structuur over en verander de inhoud van de bestanden waar nodig. Maak bijvoorbeeld een oefening waarin twee getallen opgeteld worden gebruik makende van een functie "getallen\_optellen(getal1, getal2)" en het de resulterende som geretourneerd moet worden.

<https://github.com/dodona-edu/example-exercises/tree/master/python/echo-function-print>

## 12.2 Turtle

Documentatie: <https://github.com/dodona-edu/judge-turtle>

### 12.2.1 Kenmerken

- Zeer gebruiksvriendelijk (enkel modeloplossingcode nodig)
- Vergelijken gebeurt op basis van automatisch gegenereerde een png-afbeelding

- Feedback in de vorm van absoluut en percentueel verschil in zichtbare pixels
- Nog in ontwikkeling

## 12.3 HTML/CSS

Documentatie: <https://github.com/dodona-edu/judge-html>

### 12.3.1 Kenmerken

- Geldige code?
- HTML render van code\*
- **Twee manieren om te vergelijken**
  - Vergelijken met **structuur** modeloplossing in HTML
    - Weinig feedback
    - Zeer gebruiksvriendelijk
  - Vergelijken met een **checklist** gemaakt in Python
    - Heel veel feedback mogelijk
    - Vraagt wat tijd
- Feedback
  - Syntax errors
  - Unieke ID's
  - Tag op juiste plaats?
  - Verplichte zaken aanwezig?

### 12.3.2 Nieuwe oefeningen

Voor nieuwe oefening heb je ofwel de modeloplossing in HTML nodig of een controlescript in Python.

### 12.3.3 Voorbeelden

<https://github.com/dodona-edu/example-exercises/tree/master/html>

## 12.4 SQL

Documentatie: <https://github.com/dodona-edu/judge-sql>

### 12.4.1 Kenmerken

- **SQLite**
- **Resulterende tabel vergelijken\***
- **Lezen van en schrijven naar databank**
- Datastructuur aanpassen
- Feedback
  - Syntax fouten
  - Te veel of te weinig kolommen/rijen
  - Volgorde rijen
  - Data types

### 12.4.2 Features

- Volgorde van de kolommen speelt geen rol
- Dezelfde query uitvoeren op meerdere databases
- Meerdere query's uitvoeren op één database

### 12.4.3 Nieuwe SQL-oefeningen

- **Zeer gebruiksvriendelijk**

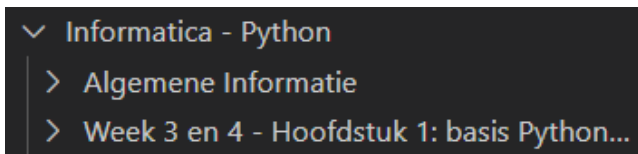
1. Database (niet te groot/klein)
2. Modeloplossingsquery nodig

#### 12.4.4 Voorbeelden

<https://github.com/dodona-edu/example-exercises/tree/master/sql>

## 13 Dodona in VS Code

Als de koppeling tussen Dodona en VSCode gemaakt is (zie 3.5.3) kan je alle Dodona-cursussen raadplegen met (links in activity bar halverwege). Klap de gewenste cursus uit. Nieuwe reeksen worden onderaan toegevoegd. D

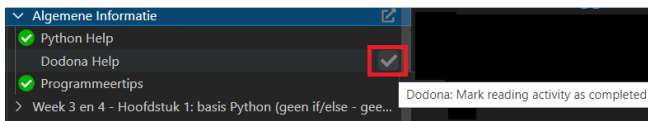


Elke reeks start bij voorkeur met een **leesactiviteit** die belangrijke informatie bevat (wat zijn basisoefeningen, extra oefeningen, wat zijn de leerdoelen).

### 13.1 Leesactiviteiten in VS Code

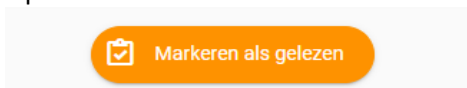
Leesactiviteiten kan je herkennen aan het boek symbool (📖). Lees elke activiteit, en duid aan dat je hiermee klaar bent:

In VS Code:



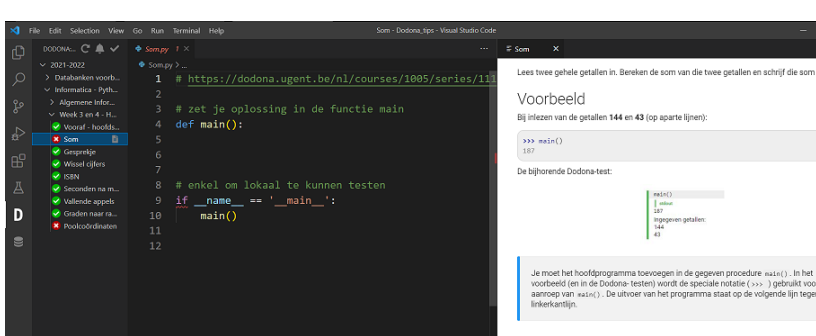
- ➔ Klik op een leesactiviteit
- ➔ Lees de tekst
- ➔ **Markeer als gelezen** door op het vinkje te klikken naast de naam van de leesactiviteit.

Op Dodona:



Dit kan ook in Dodona (onderaan de leesactiviteit).

### 13.2 Dodona oefening aanmaken in VS Code



- ➔ Twee vensters openen (Python bestand met link naar opgave op Dodona en de opgave zelf)
- ➔ *In de instellingen (Ctrl+,) kan je instellen dat de opgave niet automatisch opent: zoek naar `dodona.exercise.description.auto`.*

**Merk op:**

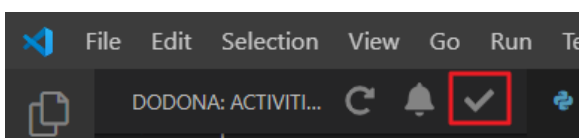
- Je lost elke oefening eerst op in in VSCode
- Gebruik telkens dezelfde folder voor de workspace!
- Bovenaan staat een link in commentaar, **verwijder die niet** (VS Code gebruikt deze eerste regel om voor de juiste oefening in te dienen). Als je de CTRL-toets ingedrukt houdt terwijl je op de link klikt opent de Dodona oefening in een nieuw browservenster.

### 13.3 Dodona oefening indienen via VS Code

De meeste oefeningen kunnen worden ingediend in Dodona, en zijn dan ook zichtbaar voor de docent.

Bij indienen in Dodona starten de automatische testen en krijg je feedback van de computer.

Als je opnieuw wil indienen, dan voer je gewoon opnieuw hetzelfde commando uit. Je kunt zo vaak indienen als je wenst, maar best eerst voldoende testen in VS Code zelf.



of

**Gebruik** ✓

of

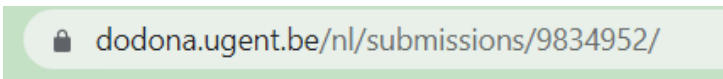
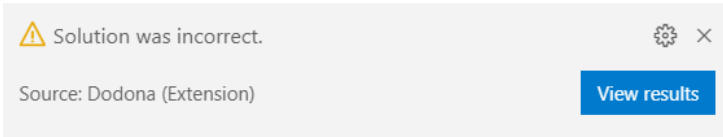
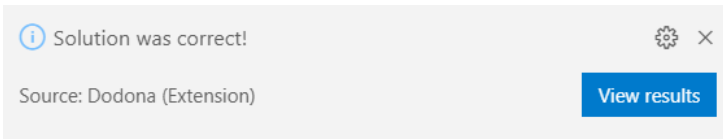
➔ Ctrl+Shift+P (Command Palette)

```
>dodona submit
Dodona: Submit solution Alt + D recently used
```

**Let op:** De cursor moet in het .py bestand staan die ingediend moet worden.

- ➔ Typ **dodona submit**
- ➔ **Enter**

Of short-cut **Alt+D**



Na enkele seconden zie je rechtsonderaan of de ingediende oplossing correct is of niet.

Als je op **View Results** klikt, zie je het verschil tussen jouw uitvoer en de verwachte uitvoer in een nieuw browservenster.

In de adresbalk van je browser staat nu een link van de vorm:






<https://dodona.ugent.be/nl/submissions/.../>.

Geef die link door aan de lesgever als je een vraag hebt.

## 13.4 Dodona Tips

- **Maak eerst een werkende versie van de opgave in VS Code!** De automatische controle met **Dodona is slechts een hulpmiddel**. Je moet zelf ook in staat zijn om je code lokaal te testen op basis van de opgave. Het is de bedoeling dat je zelf je code kan uittesten in VS Code met behulp van de terminal.
- **Indien er fouten zijn, pas je dit best aan in VS Code en dien je de opgave opnieuw in. Rechtstreeks in de browser programmeren geeft veel minder ondersteuning.**
- Er zijn ook opmerkingen over de layout van je code - ook die zijn belangrijk. Meestal kan je ze ook wegwerken.
- Als je merkt dat de automatische controle niet volledig correct zou zijn, meld je dit aan de begeleider.

## 13.5 Overige functionaliteit in VScode

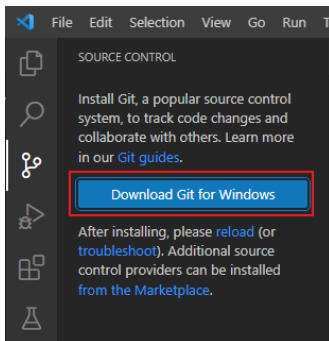
	Vernieuw de lijst met activiteiten.
	Geeft aan of de student ongelezen feedback heeft. Klikken op dit icoon verwijst door naar <a href="https://dodona.ugent.be/notifications">https://dodona.ugent.be/notifications</a> .
	Code indienen in Dodona. Alternatief voor Alt+D en Ctrl+Shift+P, Dodona: Submit Solution.
	Bekijk de beschrijving van de oefeningenreeks of cursus in een nieuw browservenster. (Zweef met je muisaanwijzer over een oefeningreeks of cursus in de linkerkolom van de Dodona extensie.)
	Bekijk de opgave van de oefening in VS Code. (Zweef met je muisaanwijzer over een oefening in de linkerkolom van de Dodona extensie.)


## 14 GitHub in VS Code (optioneel)

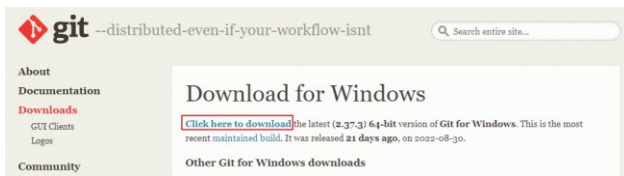
Dit is een manier om git reektsreeks in VS Code te gebruiken. Het voordeel is dat je dan maar één programma nodig hebt, maar het is helaas minder gebruiksvriendelijk.

### 14.1 Configuratie

Hier maken we de koppeling tussen VS Code en GitHub.



-  Source control
- **Download Git for Windows**
- Open

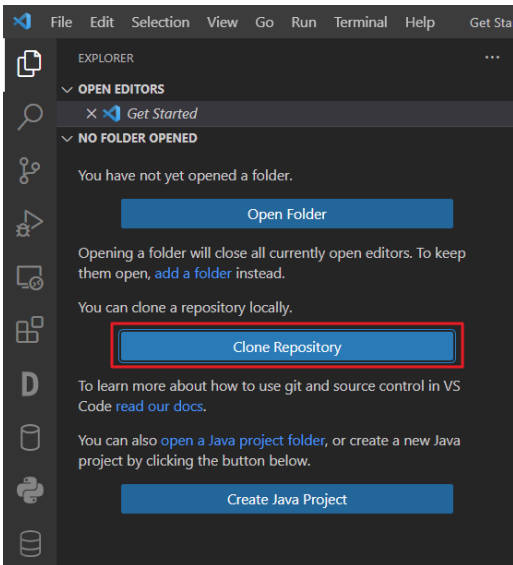


- **Click here to Download** the latest 64-bit version of Git for Windows
- Ergens opslaan
- Bestand dat werd gedownload openen/uitvoeren

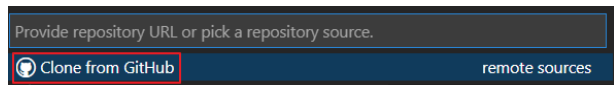
- Next
- Select Destination Location > Next
- Select components > Next
- Select Start Menu Folder > Next
- Choosing the default editor used by Git:  
**Use Visual Studio Code as Git's default editor**  
> Next
- Adjusting the name of the initial branch in new repositories > Next
- Adjusting your PATH environment > Next
- Choosing the SSH executable > Next
- Choosing HTTPS transport backend > Next
- Configuring the line ending conversions > Next
- Configuring the terminal emulator to use with Git Bash > Next
- Choosing the default behavior of git pull > Next
- Choose a credential helper > Next
- Configuring extra options > Next
- Install
- Finish



- **reload** je VS Code (Hetzelfde effect kan je bereiken door VS Code te sluiten en opnieuw op te starten.

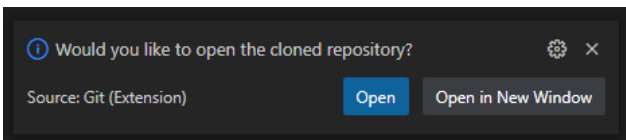


- **File** (linksboven)
- New Window
-  File Explorer
- Clone Repository (zie je deze optie niet, kies dan voor Help (bovenaan), Get Started)
- Clone from GitHub



- Pop-up: The extension 'GitHub' wants to sign in using GitHub. > **Allow**
- Inloggen op GitHub
- Pop-up: VS Code Openen? > **VS Code openen**
- Allow an extension to open this URI? > **Open**

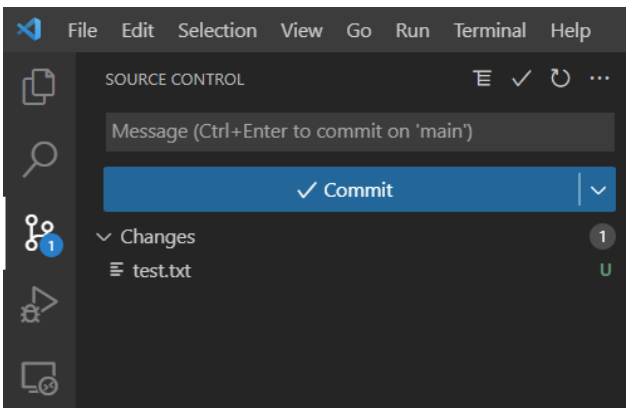
**Nu krijg je een overzicht van al je repository's.**




- Selecteer je Dodona-oefeningen repository
- Definieer waar de lokale kopie van de bestanden op Github opgeslagen moeten worden. Meestal is dit C:\Users\jouw\_gebruikersnaam\Documents\GitHub. Maak indien nodig zelf een map GitHub aan.
- Eventueel nog "Sign in with your browser"
- Would you like to open the cloned repository? > **Open**
- Do you trust the authors of the files in this folder? **Yes, I trust the authors**

## 14.2 Bestanden opladen naar GitHub

Nadat je enkele wijzigingen gemaakt hebt, kan je die versturen naar GitHub. In principe kan je welke bestanden verstuurd moeten worden, maar in de praktijk zullen dit bijna altijd alle bestanden zijn.



- Voeg een bestand toe of wijzig de inhoud van een bestand.
-  Source control
- Voeg een eigen bericht toe die kort zegt wat er veranderd is in **Message**. Dit is puur voor jezelf om achteraf makkelijk dingen terug te vinden.
- Commit
- There are no staged changes to commit. Would you like to stage all your changes and commit them directly? Yes of **Always**

De eerste keer moet je naam en e-mail nog toevoegen.

- Windows-knop
- Zoek naar **Git CMD**
- Geef deze twee regels in met je eigen gegevens

```
git config --global user.name  
"Voornaam Achternaam"
```

```
git config --global user.email  
voornaam.achternaam@example.com
```

- Sync Changes
- This action will pull and push commits from and to 'origin/main'. > OK of **OK, Don't Show Again**

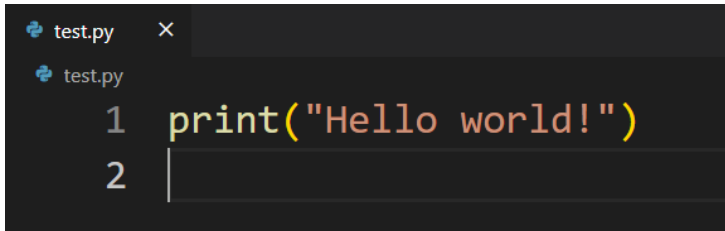
Ga nu naar je repositories op GitHub met je browser. Nu zou je je wijzigingen online moeten zien.

# 15 Python code uitvoeren, inlezen van tekst en debuggen (Optioneel)

## 15.1 Python code uitvoeren in Terminal

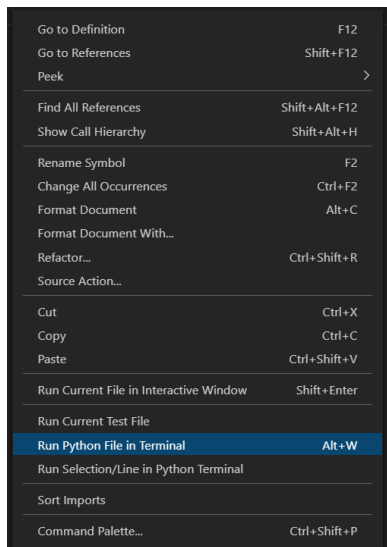
Start VSCode op vanuit de map waarin je wilt werken. Je plaats alle oefeningen best in dezelfde map.

De Python-code van een (.py) bestand kan je uitvoeren in de **Terminal**. De [Python](#) extensie (developer: Microsoft) moet hiervoor geïnstalleerd zijn.

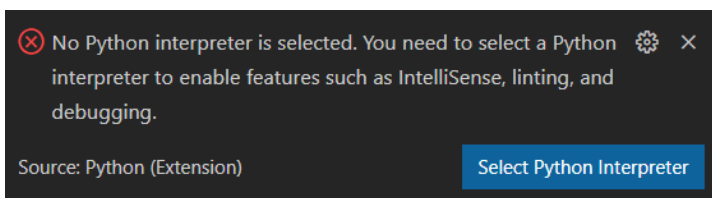


```
test.py x
test.py
1 print("Hello world!")
2
```

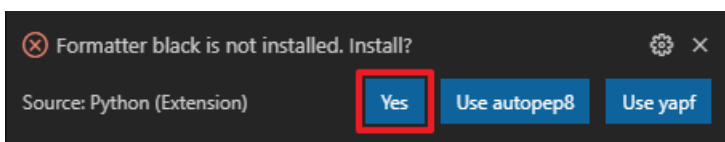
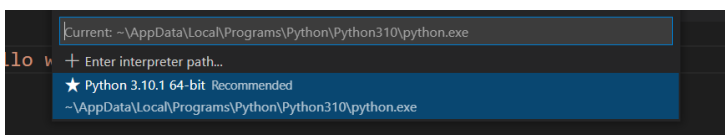
- Open een Python (.py) bestand (of maak er één indien nodig met `+`, kies een bestandsnaam en voeg zelf de `.py` extensie toe, Enter)
- Typ de Python code in dit .py-bestand



- Rechtsklikken op de Python-code
- **Run Python File in Terminal**
- of
- **Alt+w sneltoets** (nadat je cursor in de Python code staat). (Deze sneltoets werd automatisch ingesteld, zie [Instellingen en sneltoetsen manueel aanpassen](#) als dit niet correct werkt).

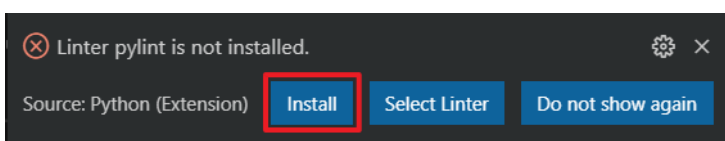


- Selecteer indien nodig de Python interpreter die je zonet geïnstalleerd hebt.
- Python 3.10.1 64-bit (Recommended)



Als je een melding over **black** en/of **pylint** krijgt:

- **Install / Yes** (en eventueel toestemming geven om te installeren)
- Wachten
- Bovenstaande stap opnieuw uitvoeren



```
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
Microsoft Windows [Version 10.0.19042.630]
(c) 2020 Microsoft Corporation. Alle rechten voorbehouden.
>"C:/Program Files/Python38/python.exe"
Hello World
```

Er wordt onderaan een tweede venster (**Terminal**) geopend met de uitvoer van het programma.

## 15.2 Informatie inlezen met Python via Terminal

```
python_test.py > ...
1 b = input("Geef b:")
```

→ Om informatie aan de gebruiker te vragen typ je volgende code in een **Python** bestand.

```
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
Microsoft Windows [Version 10.0.19042.630]
(c) 2020 Microsoft Corporation. Alle rechten voorbehouden.
>"C:/Program Files/Python38/python.exe"
Geef b:128
```

→ Bij het uitvoeren van deze code via **Terminal** verschijnt de tekst in het Terminal venster.  
→ Geef een waarde in  
→ Bevestig met **Enter**

Het is de bedoeling dat je zelf je code uittest in VS Code met behulp van de Terminal, vooraleer je het indient in Dodona. Dodona is slechts een hulpmiddel.


## 15.3 Python code debuggen (optioneel)

Aangezien zelfs de meest doorgewinterde programmeur onmogelijk broncode kan schrijven die meteen kan uitgevoerd worden, zonder fouten te produceren, bieden programmeeromgevingen zoals VS Code heel wat ondersteuning voor het debuggen van broncode.

Debuggen is een manier om fouten te detecteren en op te lossen door te kijken wat er allemaal achter de schermen gebeurt. [Hier](#) lees je vanwaar deze term zijn oorsprong vindt.

Om te debuggen volstaat het om eerst één of meerdere breekpunten (**breakpoints**) toe te voegen aan je broncode. Dit doe je door te klikken in de linkermarge van de Editor.


```
File Edit Selection View Go Run Terminal Help
demo.py - De
RUN
demo.py > ...
1 for i in range(10):
2     print(i)
3
```

→ **Klik links van een regelnummer, zodat er een rode bol verschijnt.** Vanaf dit punt kan je in detail zien wat er gebeurt achter de schermen.  
→  **> Run and Debug**

```
Een debug-configuratie selecteren
Debug-configuratie
Python-bestand Python-bestand debuggen
Module Python module/package debuggen
Extern aankoppelen Een externe Python-applicatie debuggen
Attach using Process ID Attach to a local process
Django Web-applicatie
Flask Web-applicatie
Pyramid Web-applicatie
```

→ Python-bestand  
→ Naast de code wordt de huidige waarde weergegeven en in de debugger kan je ook de huidige waarden en gegevens-types van de variabelen zien in de **Variables** tab.

```
test.py - Demo esteroso VS Code - Visual Studio Code
VARIABLES
Locals
special variables
i: 2
Globals
special variables
i: 2
1 for i in range(10): i = 2
2     print(i) i = 2
3
```







→ Met de blauwe pijl  kan je dan stap voor stap door je code gaan en

Merk op dat ondertussen ook in het editor-venster de waarden van de variabelen weergegeven worden. Je kan de waarde (en het gegevenstype) van een variabele ook bekijken door met de muis over de naam van de variabele in de broncode te bewegen.

zien wat er normaal achter de schermen gebeurt.

Een overzicht van de knoppen die je bij debuggen kan gebruiken in onderstaande tabel.




	<b>Continue (F5)</b> - Gebruik deze knop om naar het volgende breekpunt te springen. Indien er verder geen breekpunten meer tegengekomen worden, zal de code dus tot het einde uitgevoerd worden (of totdat er zich een fout tijdens de uitvoering voordoet).
	<b>Step Over (F10)</b> - Spring over een aanroep van een functie of een methode heen; het zal een methode of functie uitvoeren zonder binnen de geassocieerde code ervan te springen.
	<b>Step Into (F11)</b> - Spring naar het volgende statement dat zal uitgevoerd worden in je broncode; indien het volgende statement een methode of een functie is, dan zal deze sneltoets ervoor zorgen dat je binnen de geassocieerde code van deze methode of functie zal springen.
	<b>Step Out (Shift+F11)</b> - Springt naar de code die de huidige methode of functie heeft aangeroepen; het zal het huidige codeblok dus verlaten en terugkeren naar de aanroepende code.
	<b>Restart (Ctrl+Shift+F5)</b> - Gebruik deze knop om opnieuw te beginnen met te debuggen (van vooraf aan).
	<b>Stop (Shift+F5)</b> - Gebruik deze knop om te stoppen met debuggen.

# 16 HTML bestand openen in Chrome

Hiervoor moet de [Open in Browser](#) (developer: TechER) geïnstalleerd zijn.

Start VS Code op vanuit de map waarin je wilt werken. Je maakt dus best de nodige map(pen) aan in Windows Verkenner (of Mac Finder).

```
index.html x python_test.py
index.html > html > body > h1
1 <html>
2   <body>
3     <h1>
4       Hello World
5     </h1>
6   </body>
7 </html>
```


- ➔ Open het HTML-bestand (index.html) bestand (of maak er één indien nodig met , kies als bestandsnaam **index** en voeg zelf de **.html** extensie toe, Enter)
- ➔ Typ je de tekst die hiernaast getoond wordt over.

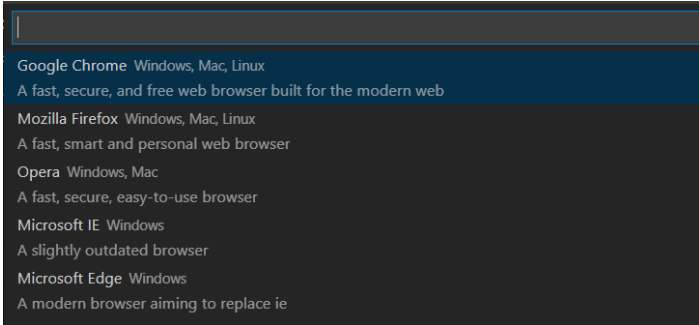
```
<> index.html x
<> index.html > head
1 <head>
2   <tit
3 </head>
4
5 <body>
6   <h1>
7
8 </h1>
9 </body>
```

Go to Definition	F12
Peek Definition	Alt+F12
Find All References	Shift+Alt+F12
Peek References	Shift+F12
Change All Occurrences	Ctrl+F2
Format Document	Shift+Alt+F
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Open In Default Browser	Alt+B
<b>Open In Other Browsers</b>	<b>Shift+Alt+B</b>
Command Palette...	Ctrl+Shift+P

Om het resultaat van deze code te zien:

- ➔ **Alt + B**
- of
- ➔ **Rechtsklikken** in index.html bestand, **Open in Default Browser**

Je hoeft niet iedere keer een nieuwe tab te openen. Door op **F5** te klikken of op dit  icoon te klikken wordt de webpagina geüpdatet.



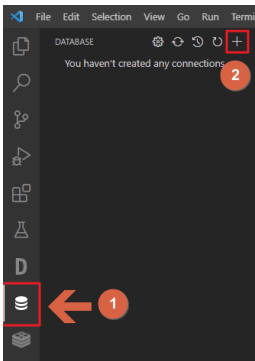
**Opgelet:** Indien Google Chrome niet je standaardbrowser is:

- ➔ **Shift+Alt+B**
- of
- ➔ **Rechtsklikken** in index.html bestand, **Open in Other Browser**

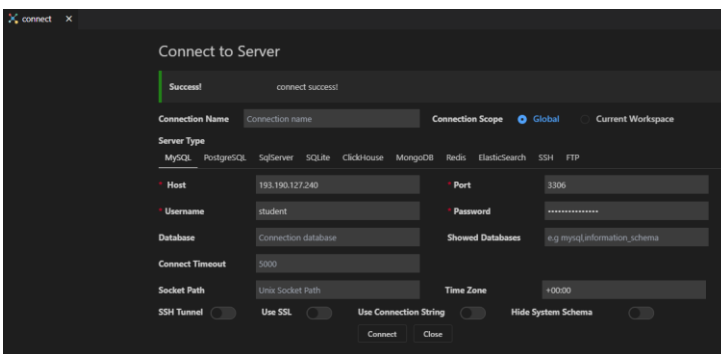
## 17 Databanken in VS Code (Optioneel)

De MySQL Databank waarmee je kan werken tijdens het labo is beschikbaar gesteld via een server op volgend adres: 193.190.127.240. Verbinden met databank in VS Code

Hiervoor moet de [MySQL](#) (developer: cweijan) geïnstalleerd zijn.



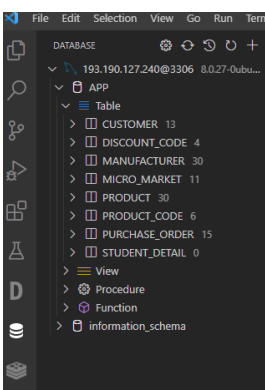
- Start VS Code op vanuit de map waarin je wil werken.
- **Database icoon** in activity bar
- **+** -teken (Add Connection)



Vul achtereenvolgens volgende informatie in:

Host	<b>193.190.127.240</b>
Port	<b>3306</b>
Username	<b>???</b>
Password	<b>???</b>

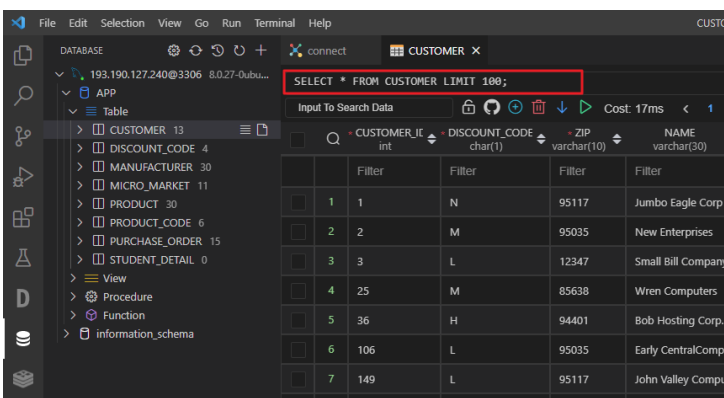
- **Connect**



Links verschijnt nu de geconnecteerde databank: **193.190.127.240@3306**

- Klik de geconnecteerde databank open

### 17.1 Eerste query uitvoeren




- Selecteer een table (bv. CUSTOMER)

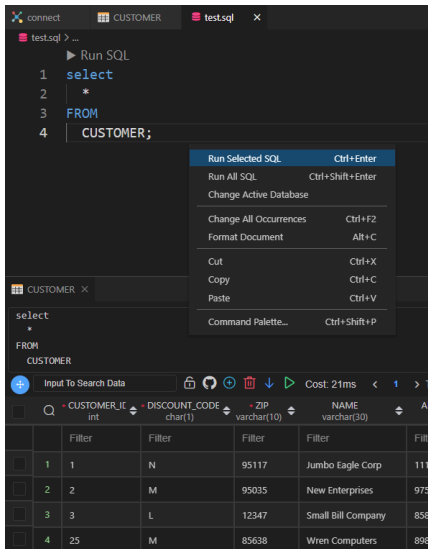
Er verschijnt een tabblad. Het bovenste tekstvak bevat de gevraagde query, daaronder staan de resultaten.

### 17.2 SQL script maken en uitvoeren

Je kan alle query's ook opslaan als .sql files in de map de geopend werd in VSCode.

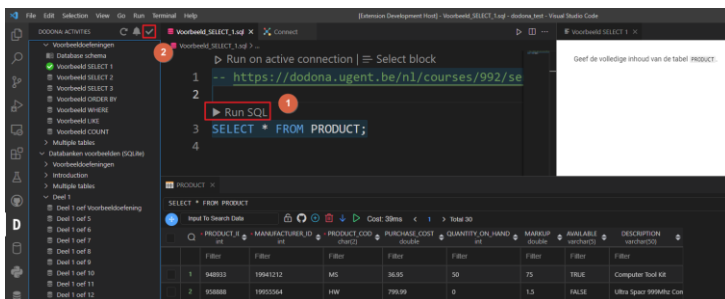
- ➔  (New File icoon naast mapnaam)
- ➔ Kies een bestandsnaam met de **.sql** extensie
- ➔ Voeg zelf de **.sql** extensie toe
- ➔ **Enter**

Het bestand wordt nu aangemaakt in de geselecteerde map.



- ➔ Rechtsklikken in **.sql** bestand op query
- ➔ **Run Selected SQL** (Ctrl+Enter) of **Run SQL** (bovenaan)
- ➔ **Let op:** Voorzie na elk statement een **puntkomma**.

## 17.3 Databank oefeningen in Dodona



- ➔ **Run SQL** (testen op online databank)
- ➔ Resulterende tabel wordt getoond.
- ➔ **Vinkje** om in te dienen in Dodona
- ➔ Jouw resulterende tabel wordt vergeleken met de modeloplossing.



## 18 VS Code sneltoets (Optioneel)

Sneltoets (*shortcuts*) laten je toe om een snel bepaalde actie uit te voeren die je anders met de muis zou uitvoeren.

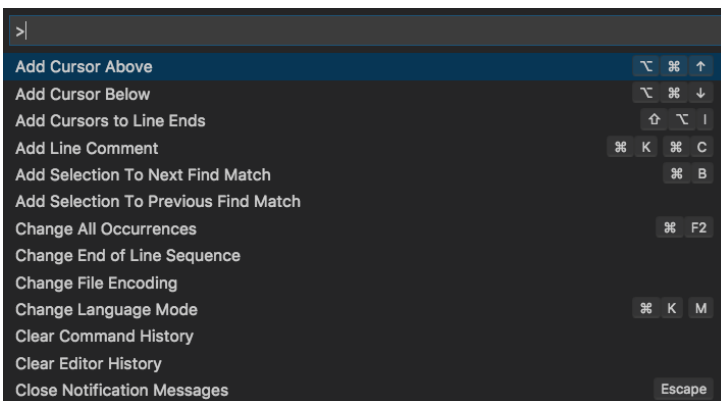
### 18.1 Command Palette openen

Het Command Palette is een veelgebruikte manier om te interfacen met VS Code.

Er zijn veel methodes om dit venster bovenaan de editor te openen, maar de meest gebruikte is de sneltoets:

Voor Windows  +  +  en voor Mac  +  + .

Bijna alles wat je via de UI kan doen in VS Code, kan je ook uitvoeren met een commando in het Command Palette. Dit is dus een sneltoets om zeker te onthouden.



**CTRL + SHIFT + P**

(Mac: **Command + Shift + P**)

of

→ **View**

→ **Command Palette...**

of

→  (linksonder)


→ **Command Palette...**

**Tip:** Aan de rechterkant in het Command Palette staat ook telkens de sneltoets om het commando uit te voeren (indien die bestaat).

## 18.2 Sneltoetsen

### 18.2.1 Belangrijkste sneltoetsen

1. **Command Palette** (veelgebruikte manier om te interfacen met VS Code)

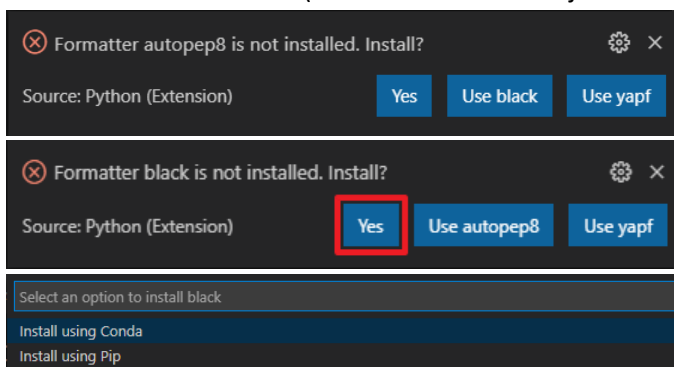
Ctrl+Shift+P (Mac: Command+Shift+P) of  > **Command Palette...**

2. **Code uitlijnen en herschikken** (`editor.action.formatDocument`):

Shift+Alt+F sneltoets of Rechtsklikken, Format document (aanbevolen sneltoets Alt+C)

Als je deze melding krijgt:

→ **Use black** of **Yes** (Het om het even wat je kiest: Conda of Pip)



3. **Code uitvoeren in Interactive Window** (`jupyter.runFileInteractive`):

Shift+Enter sneltoets (sneltoets zelf toevoegen indien nodig) of Rechtsklikken op de code, Run Current File in Interactive Window

4. **Code uitvoeren in Terminal (python.exeInTerminal):**

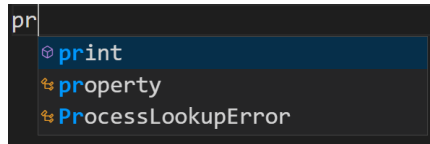
Alt+W sneltoets (sneltoets zelf toevoegen indien nodig) of Rechtsklikken op de code, Run Current File in Interactive Window

5. **Regel code in commentaar zetten (editor.action.commentLine):**

Ctrl+: sneltoets of Edit, Toggle Line Comment (aanbevolen sneltoets Alt+X)


6. **Autocomplete oproepen**

Ctrl+Tab (meestal gaat dit automatisch, maar niet altijd)



## 18.2.2 Overige sneltoetsen

1. **Debug code (workbench.action.debug.start):**

F5 sneltoets of  in Activity Bar, Run and Debug > Python-bestand

2. **Autocomplete suggestie accepteren**

Tab

3. **Bestand/map openen in Windows Verkenner (revealFileInOS)**

Rechtsklikken op bestand/map in VS Code, **Reveal in File Explorer** (of Shift+Alt+R)

4. **Documentatie/help opvragen:**

Zweef met je muisaanwijzer boven een keyword.

5. **Snel zoeken (workbench.action.findInFiles en actions.find):**

Gebruik Ctrl+Shift+F om te zoeken in je volledige project. Ctrl+F zoekt in het huidige bestand (f van find). Deze sneltoets werkt in bijna alle programma's (bv. Word, Adobe Reader, Chrome ...).

6. **Snel vervangen (workbench.action.replaceInFiles en editor.action.startFindReplaceAction):**

Gebruik Ctrl+Shift+H om te vervangen in je volledige project. Ctrl+H vervangt in het huidige bestand.

7. **Zoeken in files:**

In de linkerkolom met files kan je gewoon beginnen typen. De files die beantwoorden aan je zoekopdracht worden gemarkeerd.

8. **Variabele hernoemen (editor.action.rename):**

F2 sneltoets of Rechtsklik, Rename Symbol

9. **Lijn(en) code met één regel naar boven of onder verplaatsen (editor.action.moveLinesUpAction en editor.action.moveLinesDownAction):**

Alt+UpArrow of Alt+DownArrow (combineer met Shift om die regel ook te kopiëren)

10. **Snel een bestand openen (workbench.action.quickOpen):**

CTRL+P (Enter om het bestand te openen of RightArrow om het op de achtergrond te openen)

11. **Snel een bestand opslaan (workbench.action.files.save):**

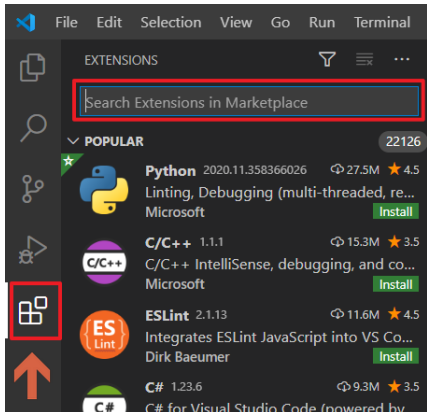
CTRL+S


## 18.3 Alle sneltoetsen

Een volledig overzicht op <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf>.

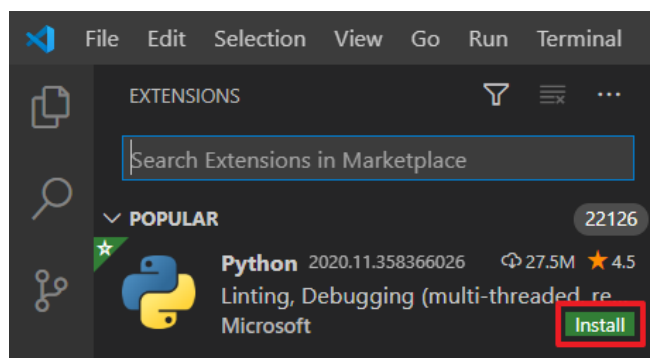
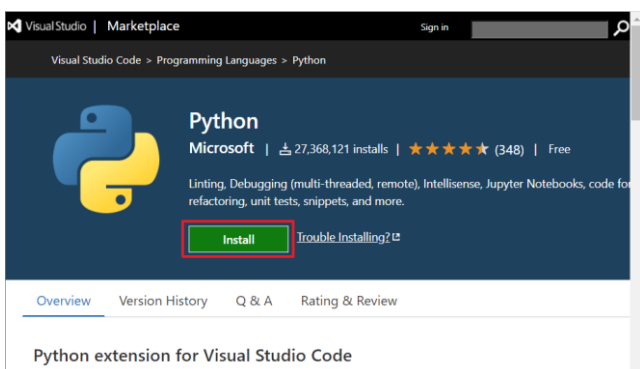
# 19 Extensies (optioneel)

## 19.1 Extensie manueel installeren



-  **Extensions** in linkerkolom
- of
- **Ctrl+Shift+X**





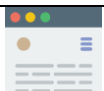
Je kan een extensie op twee manieren toevoegen:






**Manier 1 (via browser):** Klik op **Install** (groene knop) in je browser nadat je op onderstaande links geklikt hebt. De installatie pagina van de extensie opent nu in VS Code. Klik op de blauwe **Install** knop in VS Code om te extensie te installeren.

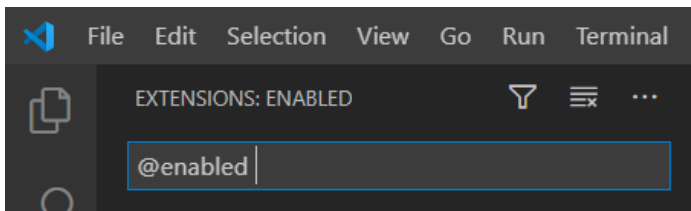
**Manier 2 (rechtsreeks in VS Code):** Typ de naam van de derde kolom in onderstaande tabel over in de zoekbalk linksbovenaan. Klik vervolgens op **Install**.



Volgende extensies worden in het labo gebruikt:

	<a href="#"><u>Python</u></a> (developer: Microsoft)  Bevat automatisch deze extensie:  Pylance	zoek: <i>ms-python.python</i>
	<a href="#"><u>Black Formatter</u></a> (developer: Microsoft)	Zoek: <i>ms-python.black-formatter</i>
	<a href="#"><u>Jupyter</u></a> (developer: Microsoft)	zoek: <i>ms-toolsai.jupyter</i>
	<a href="#"><u>Open in Browser</u></a> (developer: TechER)	zoek: <i>techer.open-in-browser</i>

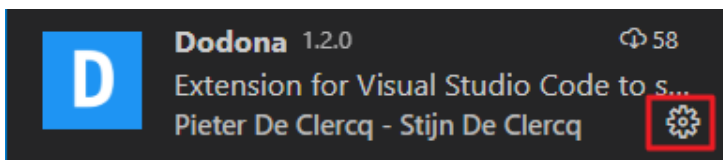
	<a href="#">MySQL</a> (developer: cweijan)	zoek: <i>cweijan.vscode-mysql-client2</i>
	<a href="#">Excel Viewer</a> (developer: GrapeCity)	zoek: <i>grapecity.gc-excelviewer</i>
	<a href="#">Dodona</a> (developer: Pieter De Clercq – Stijn De Clercq)	zoek: <i>thepieterdc.dodona-plugin-vscode</i>



## 19.2 Geïnstalleerde extensies weergeven



-  **Extensions** linksbovenaan in linkerkolom (of Ctrl+Shift+X)
- Typ @enabled
- of
-  > Enabled

## 19.3 Specifieke instellingen voor een extensies weergeven

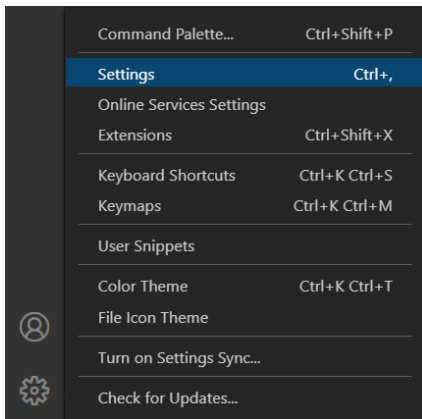


-  **Extensions** linksbovenaan in linkerkolom (of Ctrl+Shift+X)
- 
- Extension settings

## 20 Instellingen en sneltoetsen manueel aanpassen (Optioneel)

De VS Code programmeeromgeving biedt heel wat mogelijkheden waar je als beginnende programmeur niet direct gebruik zult van maken en die je gaandeweg zult leren ontdekken. Bovendien kan je de omgevingen voor een groot stuk naar je hand zetten door de voorkeursinstellingen aan te passen. Elke programmeur ontwikkelt immers ook zijn eigen manier van werken. Hieronder zetten we je alvast op weg met enkele instellingen die we jullie kunnen aanbevelen.

### 20.1 Instellingen openen en aanpassen



- File
- Preferences
- Settings

of

-  (linksonderaan) > Settings

of

- Ctrl+,

#### 20.1.1 Aanbevolen instellingen

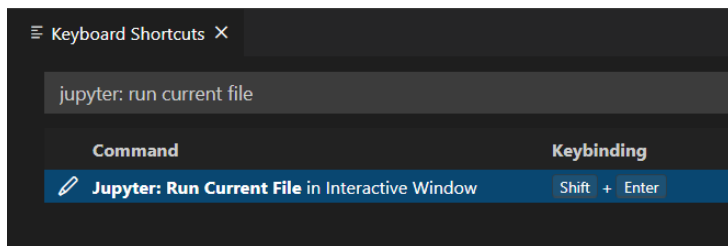
*Als je de Settings Sync extensie gebruikt hebt om de aanbevolen instellingen te downloaden in een eerdere stap, worden onderstaande instellingen automatisch goed ingesteld.*

Volgens bovenstaande stappen kan je in VS Code onderstaande instellingen aanpassen. Het wordt sterk aanbevolen onderstaande stappen te doorlopen. Typ in het zoekvak telkens de tekst van de eerste kolom van onderstaande tabel:

Naam	Betekenis	Oude waarde	Aanbevolen waarde
<code>window.zoomLevel</code>	Knoppen worden iets groter weergegeven	0	1
<code>python.analysis.completeFunctionParens</code>	Functies krijgen automatisch haakjes bij het aanvaarden van de suggestie.	false	true
<code>editor.fontSize</code>	Lettergrootte van de editor.	14	20
<code>files.autoSave</code>	Automatisch opslaan als de editor niet meer gefocust is. Verlies geen kostbaar werk door automatisch op te slaan.	off	onFocusChange
<code>editor.mouseWheelZoom</code>	Tekst in editor vergroten met Ctrl + muis scroll.	false	true


<code>debug.inlineValues</code>	Waardes van variabelen worden weergegeven tijdens debuggen.	false	true
<code>editor.minimap.enabled</code>	Mini overzicht van de code staat in de weg.	true	false
<code>editor.defaultFormatter</code>	Black zal de code beter uitlijnen.	none	black
<code>editor.formatOnSave</code>	Code uitlijnen telkens de code opgeslagen wordt.	false	true
<code>python.analysis.typeCheckingMode</code>	<b>Automatisch controleren of types overeenkomen op basis van type hinting.</b>	off	basic
<code>editor.guides.bracketPairs</code>	Volledige inhoud van haakjes onderlijnen.	false	active
<code>editor.linkedEditing</code>	Openende en sluitende HTML-tag tegelijk wijzigen	false	true

## 20.2 Sneltoets aanpassen/toevoegen



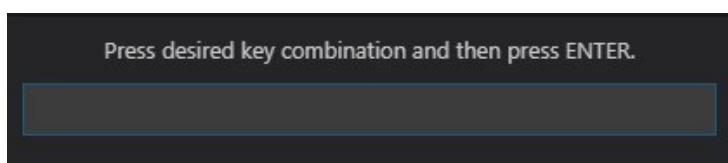
- File
- Preferences
- Keyboard Shortcuts

of

-  (linksonderaan) > Keyboard Shortcuts

- Zoek naar een commando (bv.):

**Jupyter: Run Current File** in Interactive Window



- **Dubbelklik op het eerste commando**
- Geef de nieuwe sneltoets in in nieuw venster (bv.):

**Shift+Enter**

- Bevestig met **Enter** (negeer de opmerking)

### 20.2.1 Aanbevolen sneltoetsen

*Als je de Settings Sync extensie gebruikt hebt om de aanbevolen instellingen te downloaden in een eerdere stap, worden onderstaande instellingen automatisch goed ingesteld.*

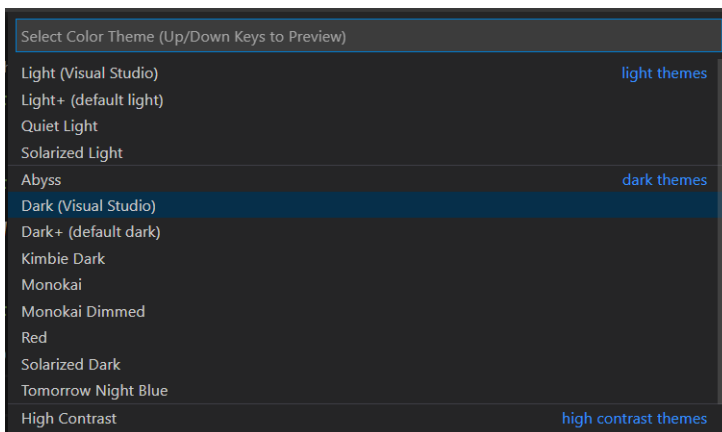
Volgens bovenstaande stappen kan je in VS Code voor veelgebruikte functies je eigen (eenvoudigere) sneltoets kan aanmaken. Typ in het zoekvak telkens de tekst van de eerste kolom van onderstaande tabel:

Naam	Betekenis	Alternatief voor shortcut	Oude shortcut	Aanbevolen shortcut

editor.action.formatDocument	Code uitlijnen	Rechtsklik in code > Format document	Ctrl+Alt+F	Alt+C
editor.action.commentLine	Regel code in commentaar zetten	Edit, Toggle Line Comment	Ctrl+:	Alt+X
python.execInTerminal	Python code uitvoeren in terminal	Rechtsklik in code > Run Python file in <b><i>Terminal</i></b>	geen	Alt+W
jupyter.runFileInteractive	Huidig .py bestand uitvoeren in interactief venster	Rechtsklik in code > Run Current File in <b><i>Interactive Window</i></b>	geen	Shift+Enter


Bovenstaande functionaliteit wordt heel veel gebruikt tijdens het programmeren in VS Code. Het loont dus zeker de moeite om bovenstaande stappen te doorlopen (tijdwinst).

## 20.3 Kleurthema aanpassen



- ➔ File
- ➔ Preferences
- ➔ Color Theme

of

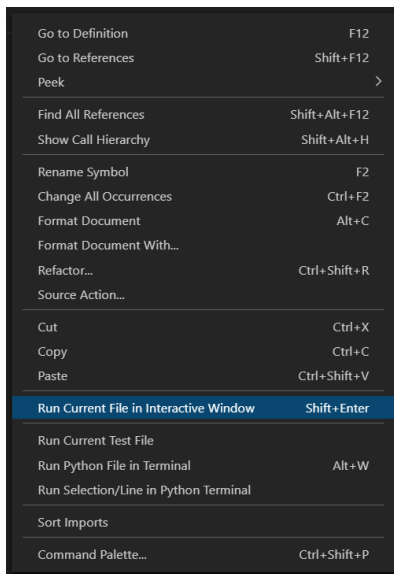
- ➔  (linksonderaan) > Color Theme
- ➔ Gebruik de pijltjestoetsen om een preview te krijgen
- ➔ Bevestig je keuze met **Enter** of selecteer je keuze met de muis.

Tip: Er is enorm veel keuze door gratis Color Theme Extensions te downloaden van de VS Code Marketplace.

## 21 Python in Interactive Window (Optioneel)

In een Python *Interactive Window* kan je code lijn per lijn ingeven en kijken naar de aanpassingen in de variabelen.

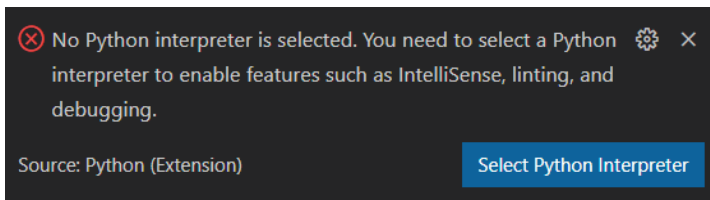
### 21.1 Python code uitvoeren in Interactive Window



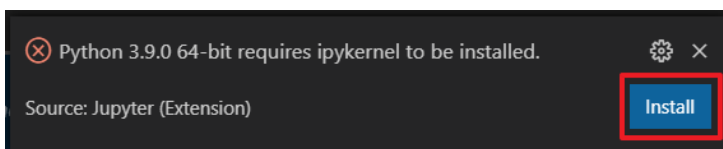
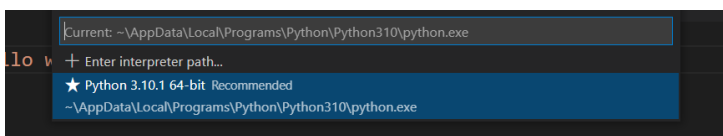
- ➔ Rechtsklikken op de code
- ➔ **Run Current File in Interactive Window**

of

- ➔ **Shift+Enter** sneltoets (nadat je cursor in de Python code staat). (Deze sneltoets werd automatisch ingesteld, zie [Instellingen en sneltoetsen manueel aanpassen](#) als dit niet correct werkt.)

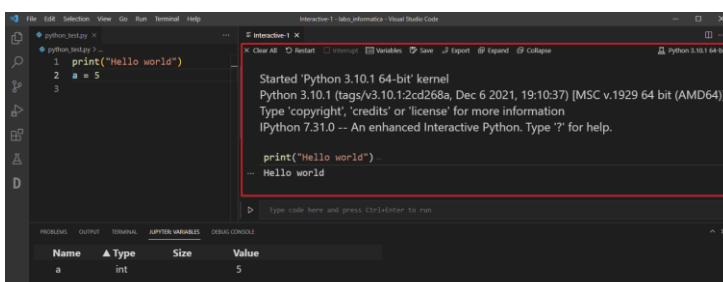
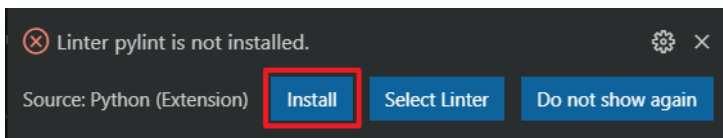
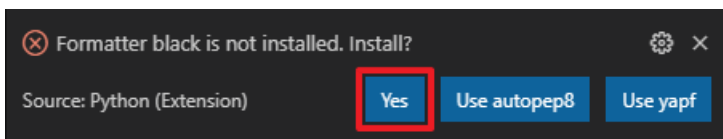


- ➔ Selecteer indien nodig de Python interpreter die je zonet geïnstalleerd hebt.
- ➔ Python 3.10.1 64-bit (Recommended)



Als je een melding over **ipykernel**, **black** en/of **pylint** krijgt:

- ➔ **Install / Yes** (en eventueel toestemming geven om te installeren)
- ➔ Wachten
- ➔ Bovenstaande stap opnieuw uitvoeren



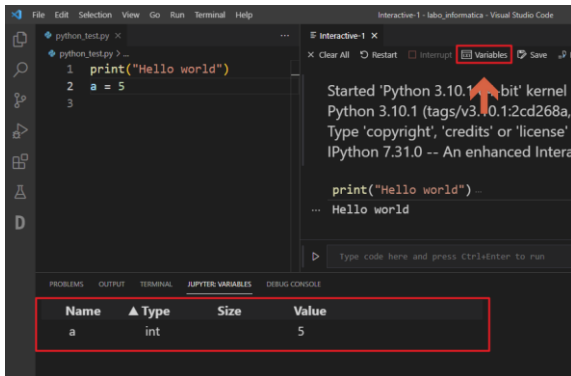
Er wordt (rechts) een tweede venster (**Python Interactive**) geopend en na een tijdje wordt dit ingevuld met informatie. Daaronder komt de uitvoer van het programma.

(De eerste keer moet je een tijdje wachten op inhoud van het venster).



```
[2] Type code here and press shift-enter to run
```

In een Python Interactive Window kan je code lijn per lijn ingeven en kijken naar de aanpassingen in de variabelen.




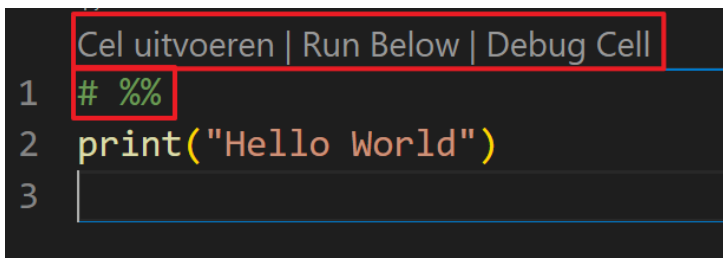
Typ volgende code in het **Python Interactive** venster (zie hiernaast):

```
a = 5
```

Voer deze code uit met **Shift+Enter**

➔ **Tip:** In het Python Interactive Window kan je ook functies oproepen.

➔  (Show variables in Jupyter notebook)



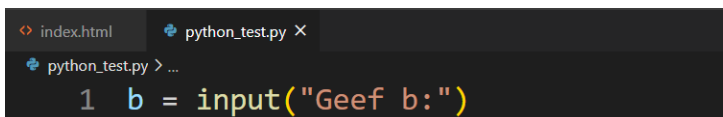
➔ Maak een cell code aan in een .py bestand door `#%%` te typen

➔ **Shift+Enter**

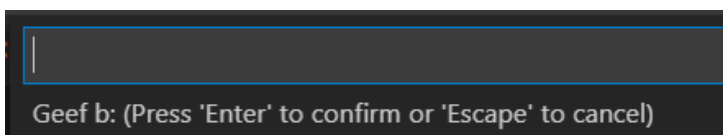
De uitvoer van de cell wordt getoond in het **interactive window**

Op deze manier kan je makkelijk een deel van je code uitvoeren en/of debuggen.

## 21.2 Informatie inlezen met Python via Interactive Window



Om informatie aan de gebruiker te vragen typ je volgende code in een **Python** bestand.



Bij uitvoeren van deze code via **Interactive Window** verschijnt een apart venster (bovenaan) waar je de waarde voor `b` kan intypen.

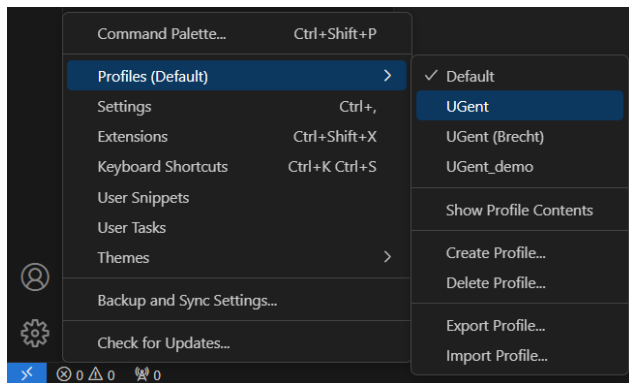
## 22 Foutoplossing

### 22.1 Al mijn instellingen zijn plots verdwenen?!



Soms kan het gebeuren dat VS Code terugvalt op het Default profiel. Dat kan gebeuren bij het openen van een andere map of na het updaten.

Als het UGent profiel actief is, zal je de eerste twee letters **UG** zien staan bij het tandwiel icoon.



→  (linksonder)

→ Profiles (Default)

→ **UGent** profiel selecteren

Linksonderaan VS Code zou er nu “**UG**” moeten staan bij het tandwiel icoon, om aan te duiden dat je in het UGent-profiel aan het werken bent.

## 23 Bronnen

- <https://docs.dodona.be/nl/>
- <https://github.com/dodona-edu>