

# Extra Uitleg

---

Tijdens de toets zal je gebruik maken van een aantal Python-modules die je misschien niet eerder bent tegengekomen. Ter ondersteuning staat in dit document aanvullende uitleg over de volgende modules:

- unittest
- unittesting in VSCode

## unittest

---

Met de **unittest** module kan je automatisch Python-code testen. Dit is de module waarmee in CodeGrade gecontroleerd wordt of de output van een ingeleverde opdracht overeenkomt met de output die wordt beschreven op de opgave. Hoewel de volledige werking van deze module gaat verder dan nodig voor het tentamen, zeker omdat unittests geen onderdeel zijn van de opgaven, is het om die reden wél belangrijk om even stil te staan bij wat de **unittest** module doet.

Met **unittest** kan je zogeheten testsuites schrijven, meestal één per functie die je wilt controleren. In een testsuite beschrijf je alle potentiële uitkomsten van een functie. Vervolgens voert de testsuite de functie uit, en controleert de testsuite of de daadwerkelijke output van de functie gelijk is aan de verwachte output.

Neem bijvoorbeeld het geval van de testsuite voor de functie **normalize\_source\_and\_destination**. Er is maar één gewenste uitkomst van deze functie: een dictionary waarin de keys **source** en **destination** zijn verwijderd, en de values van die keys opgesplitst in **src\_ip**, **src\_port**, **dest\_ip**, en **dest\_port**. De testsuite controleert éerst of het gelukt is om **source** te splitsen. Daarna controleert de testsuite of **destination** correct is gesplitst. Allebei de situaties worden los van elkaar getest.

Er zijn 12 **methoden** die **unittest** gebruikt om de outputs te vergelijken. In de onderstaande tabel staan de meest relevante beschreven.

Method	Controleert
<code>assertEqual ( a, b )</code>	Zijn de waarden van <b>a</b> en <b>b</b> hetzelfde?
<code>assertNotEqual ( a, b )</code>	Verschillen de waarden van <b>a</b> en <b>b</b> ?
<code>assertTrue ( x )</code>	Is de <b>bool(x)</b> <b>True</b> ?
<code>assertFalse ( x )</code>	Is de <b>bool(x)</b> <b>False</b> ?
<code>assertIsInstance ( a, b )</code>	Is <b>a</b> een instance van de class <b>b</b> ?

Elk van deze assertions, beweringen, heeft zijn eigen foutmelding. Lees daarom goed wat de foutmelding is, en waar deze vandaan komt. Dat geeft je een beter idee welk onderdeel van de functie een foutmelding oplevert: of het komt doordat je functie iets flagt als virus terwijl het geen virus is, of doordat je functie een virus onterecht niet opmerkt.

## unittesting in Visual Studio Code

---

Het is mogelijk om via Visual Studio Code de verschillende tests, die bij de opgave zijn toegevoegd, uit te voeren. Begin door via **File - Open Folder...** de folder te openen waar je het zip-bestand opgh.zip hebt uitgepakt. Open vervolgens het bestand **opgh.py** in de editor. Aan de linkerkant van het scherm komt dan een nieuw icoontje tevoorschijn, in de vorm van een Erlenmeyer-fles (een driehoekige fles). Als je hierop klikt krijg je 2 knoppen te zien. Klik op **Configure Python Tests** om in te stellen wat voor bestanden in deze folder test-bestanden zijn. Het zijn de bestanden die beginnen met **test\_**. Vervolgens opent Visual Studio Code automatisch de tests in VSCode, en kan je deze gebruiken om je programma te debuggen. Lees meer op de site van [Microsoft](#).