

- **Leg in je eigen woorden uit wat een unit test is.**

Unit testing in Python is een testmethode. Het wordt gebruikt om de correctheid van een onafhankelijke eenheid of tuple te testen. Problemen kunnen tijdens het ontwikkelingsproces worden getest en opgelost.

- **Waarom worden unit tests gebruikt in softwareontwikkeling?**

Omdat het een testmethode is. Wanneer we tijdens het ontwikkelingsproces een foutmelding krijgen terwijl we niet weten waar het probleem ligt, kunnen we deze gebruiken om het probleem stap voor stap op te lossen en op te lossen. Relatief gezien is het een onmisbaar hulpmiddel in het ontwikkelingsproces.

- **Wat voor testsoorten zijn er nog meer? Noem er minimaal 2 en leg de verschillen met een unit test uit.**

1. Integratietest

Het controleert de coördinatie van verschillende modules of functies in het systeem. Het doel is om te testen hoe individuele eenheden samenwerken wanneer ze worden geïntegreerd in een groter systeem.

Het verschil tussen unit test en unit test is dat unit test een eenvoudige test is voor een enkele module of unit, terwijl het een test is voor verschillende componenten die samenwerken. Relatief gezien is het complexer.

<https://circleci.com/blog/unit-testing-vs-integration-testing/#:~:text=While%20unit%20tests%20always%20take,works%20in%20an%20integrated%20way.>

2. Systeemtest

Unit testing en system testing zijn twee verschillende soorten software testtechnieken. Unit testing richt zich op het verzekeren dat elke unit van een softwareproduct werkt zoals bedoeld. System testing test daarentegen het gehele softwareproduct als geheel.

<https://www.tutorialspoint.com/difference-between-unit-testing-and-system-testing#:~:text=Conclusion,reliability%20of%20a%20software%20application.>

- **Wat zijn assert statements? Bekijk een aantal voorbeelden; deze helpen je bij het uitvoeren van opdracht 2.**

assert statements zijn handig om tijdens het ontwikkelen snel fouten te detecteren en kunnen als een vorm van zelfdocumenterende checks worden gezien. Ze helpen om te garanderen dat de code zoals verwacht werkt door bepaalde aannames te valideren.

- **Noem een paar voordelen en nadelen van unit testing. Beantwoord de vragen in je eigen woorden (dus geen ChatGPT uitdraai of Nederlandse vertalingen van websites). Zorg ervoor dat je de bronnen vermeldt die je hebt gebruikt (minimaal drie).**

Voordelen

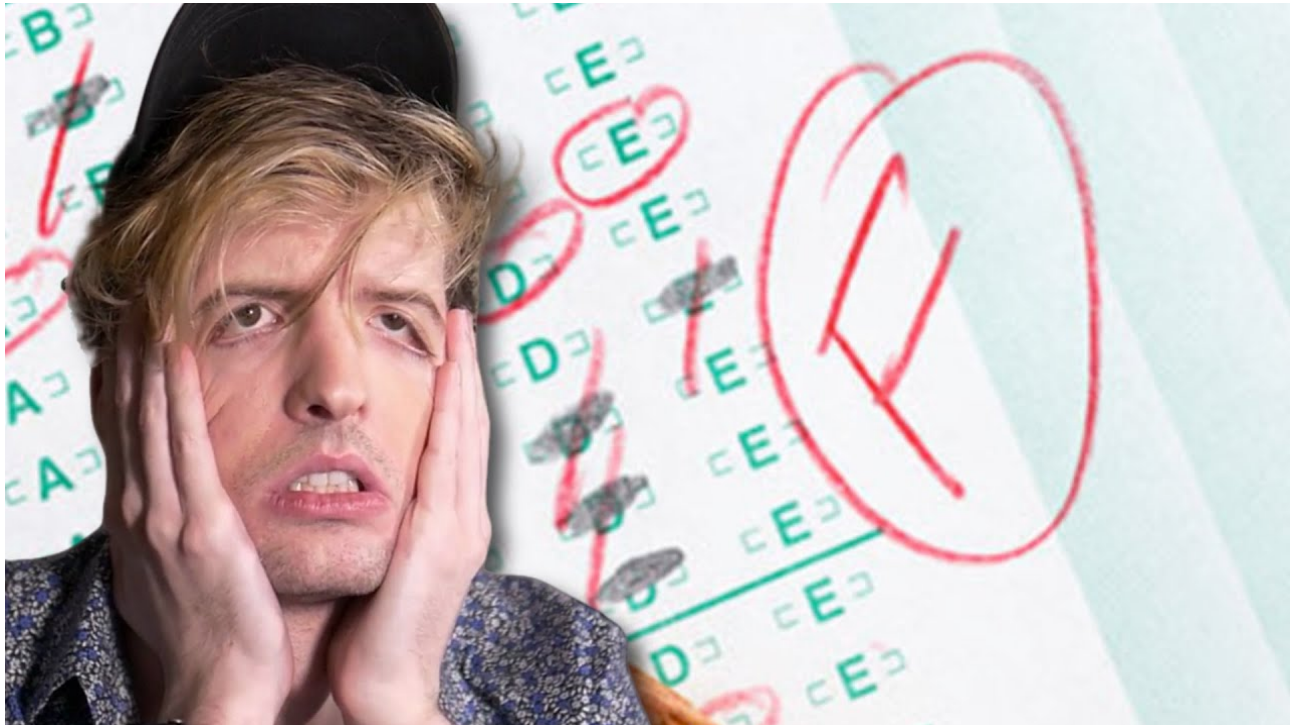
Unit tests helpen je om snel problemen in de code op te sporen door kleine functies of methodes afzonderlijk te testen. Omdat je de tests al vroeg in het ontwikkelproces schrijft, ontdek je fouten meteen, waardoor grotere en moeilijker te vinden bugs later voorkomen kunnen worden. Bovendien zorgt het testen van elk deel van de code ervoor dat je kleinere, beter gedefinieerde functies schrijft, wat de code makkelijker te onderhouden maakt en de kwaliteit van het project verbetert. Ten slotte geven unit tests vertrouwen wanneer je veranderingen doorvoert, omdat ze laten zien dat de basisfunctionaliteit nog steeds goed werkt.

Nadelen

Het schrijven van unit tests kost extra tijd, vooral als je met complexe systemen werkt. Daarnaast moeten deze tests worden onderhouden als de code verandert, waardoor het onderhoud van de tests net zo belangrijk en soms zelfs tijdrovend wordt als de ontwikkeling zelf. Ook de dekking van unit tests is beperkt; ze testen alleen afzonderlijke delen van de code en niet hoe die delen samen functioneren. Dit betekent dat je ook andere testsoorten nodig hebt, zoals integratietests en systeemtests, om te zorgen dat het hele systeem goed werkt. Tot slot kunnen slecht geschreven unit tests of een lage dekking een vals gevoel van

veiligheid geven, omdat de code er dan goed uitziet terwijl belangrijke functionaliteiten misschien niet zijn getest.

Video



Website

<https://docs.python.org/3/library/unittest.html>

Documentatie

https://en.wikipedia.org/wiki/Unit_testing