

Practicumtoets 1 – Thema 2.1 Python

- Tijdsduur: 75 minuten
- Aantal opgaven: 3
- Toegang tot internet is NIET toegestaan bij het maken van opgaven
- Plagiaat wordt altijd gemeld bij de examencommissie, conform Studentenstatuut
- Gebruik van boeken en gebruik van bestanden op je laptop is wel toegestaan
- Aan het begin downloaden van Blackboard, onder “Python”, item “Practicumtoets”: `eredivisie.py`, `twitter.json` en `simulate.py`
- Aan het einde je uitwerkingen op Blackboard uploaden in de vorm van een enkel zip-bestand met drie Python source files
- Je mag de gegeven bestanden niet aanpassen
- Je mag alleen de standaard Python-library's gebruiken, dus geen library's die je met PIP moet installeren zoals bijvoorbeeld `pandas`
- Je programma's moeten zonder aanpassingen kunnen draaien in een omgeving waar de drie gegeven bestanden in de huidige directory aanwezig zijn
- Na afloop dit papier weer inleveren bij je docent
- Let erop dat de tijd beperkt is, maak eerst de opgaven die je makkelijk vindt
- Je kan 90 punten krijgen. Je cijfer wordt bepaald door $cijfer = punten / 10 + 1$
- Succes!

Opgave 1 (30 punten)

- (10 punten) In het bestand `eredivisie.py` is een lijst opgenomen van de voetbaluitslagen van de Eredivisie in het seizoen 2017-2018 met als naam `eredivisie`. Importeer deze lijst en converteer deze naar een dictionary met als sleutel een tuple met de twee teams, en als waarde het verschil in doelpunten. Druk het resultaat hiervan af op het scherm.
- (15 punten) Bereken het aantal punten en het totale doelsaldo van elk team. De winnaar van een wedstrijd krijgt 3 punten, de verliezer 0. Bij een gelijkspel krijgen beide teams 1 punt. Importeer hiertoe de klasse `Team` uit het bestand `eredivisie.py`. Sla het resultaat op in een lijst van objecten. Druk het resultaat af op het scherm. Ter controle is gegeven dat FC Groningen 38 punten heeft en een doelsaldo van 0, en dat SC Heerenveen 46 punten heeft en een doelsaldo van -5.
- (5 punten) Bepaal de winnaar van de competitie en druk die af. Dit is het team met de meeste punten; als meerdere teams een gelijk aantal punten hebben wint van die teams het team met het beste doelsaldo. Gebruik hiervoor de functie `max` waarbij je voor de `key` een lambda-functie gebruikt.

Opgave 2 (30 punten)

- (5 punten) Lees het bestand `twitter.json` met de tweets van Donald Trump in 2017 in naar een Python-variabele. Druk de lengte van deze lijst en de eerste vijf elementen af.

- b) (10 punten) In de API van Twitter kan je vinden dat de tekst van een tweet in het veld `text` staat, het aantal retweets in `retweet_count` en het aantal favorites in `favorite_count`. Maak een lijst met tuples (`text`, `retweet_count`). Neem alleen tweets met minimaal 10.000 favorites mee. Gebruik hiervoor een list comprehension. Druk de lengte van deze lijst en de eerste vijf elementen af.
- c) (10 punten) Sorteert deze lijst op het aantal retweets, en druk de eerste vijf elementen af. Sorteert de tweet met de meeste retweets bovenaan.
- d) (5 punten) Druk de inhoud van de kortste tweet in de lijst af. Gebruik hiervoor de functie `min` waarbij je voor de `key` een lambda-functie gebruikt.

Opgave 3 (30 punten)

In deze opgave maak je een simpele simulatie voor een simpele variant van een online rollenspel. In dit spel vechten spelers gezamenlijk tegen grote vijanden. In het bestand `simulate.py` vind je een klasse `Player`, die een speler beschrijft, en een klasse `Boss`, die een vijand beschrijft. Bovendien vind je een functie `simulate` die gebruikt kan worden om de simulatie uit te voeren.

- a) (5 punten) Maak een vijand en vijf spelers aan. Gebruik de methode `target` van beide klassen om de vijand alle vijf de spelers aan te laten vallen, en de spelers ieder de vijand aan te laten vallen. Draai de simulatie en druk de naam van elk van de zes karakters en hun overgebleven aantal *hp* (hit points) af.
- b) (10 punten) Implementeer drie subklassen van de klasse `Player`; `Mage`, `Warrior` en `Priest`.
 - Een *mage* vecht precies zoals de gegeven klasse `Player`.
 - Een *warrior* begint met 5000 hp. De warrior heeft maar 1 seconde nodig om opnieuw te kunnen aanvallen, maar elke aanval richt maar 50 hp schade aan. Bovendien is het pantser van de warrior zo sterk, dat elke aanval maar de helft van de schade aanricht.
 - Een *priest* begint met 2500 hp. De priest valt de vijand niet aan, maar kan zichzelf en zijn bondgenoten genezen. Dit kan elke twee seconden, en de genezing herstelt 200 hp.
- c) (5 punten) Draai een simulatie met een vijand en vijf spelers; één warrior, twee mages en twee priests. De vijand zal eerst de warrior aanvallen, maar als de warrior dood is valt hij ook de andere spelers aan. De warrior en de mages vallen de vijand aan. De priests proberen eerst de warrior te genezen, maar als die sterft zullen ze ook de andere spelers of zichzelf genezen. Druk de naam van elk van de zes karakters en hun overgebleven aantal hp af.
- d) (10 punten) Implementeer een subklasse van `Boss`; `RaidBoss`. Deze krachtigere vijand valt niet alleen zijn doelwit rechtstreeks aan, maar zal ook elke keer dat hij aanvalt alle spelers, inclusief zijn doelwit, 50 hp aan schade toebrengen. Draai dezelfde simulatie als in de voorgaande opgave nogmaals met deze sterkere vijand en druk weer de resultaten af.