

Scripttalen

- [2010 juni examen](#)
- [2011 juni examen](#)
- [2012 juni examen](#)
- [2013 juni examen](#)
- [2018 juni examen](#)
- [2020 augustus examen](#)
- [2020 juni examen](#)
- [2021 januari examen](#)

2010 juni examen

Je krijg een bestand in de vorm van datebook.txt

Dit bestand moet je weer splitsen en dan de huidige datum vergelijken met de datum van vandaag (localtime).

Indien de datums overeenkomen moet je naar die persoon een mailtje sturen, of een tekst bestand laten printen met daarin wat info.

Dit tekstbestand moet je dan ook wegschrijven.

Daarna moet je het programma voorzien van een menu, dit moet een entry bevatten om iemand toe te voegen aan de lijst en om het programma te stoppen.

Heel simepl dus.

1 oefening python:

oefening waar je een sorteeralgoritme moet schrijven

Een oefening waarbij je op sites moet gaan zoeken naar een bepaald woord dat gebruiker ingeeft eventueel via command line.

Op basis van het aantal keer dat dit woord komt in de site is de site hot or not.

2011 juni examen

Lector: Frans Sanen

Het examen bestaat uit 1 oefening Perl, 1 oefening Python en 1 regex (die je niet kan voorbereiden, maar ter plaatse, tijdens het mondelinge gedeelte, moet bedenken).

Permanente evaluatie: 8/20

Examen: Perl (6/20), Python (4/20), mondeling gedeelte (2/20)

Perl

Je krijgt een .ics bestand met daarin afspraken. Elke afspraak heeft een aantal eigenschappen zoals begin- en einduur, locatie en naam en emailadres van de betrokken persoon.

Opdracht:

- Het script moet een email kunnen sturen a.d.h.v. een geformatteerde tekst (printf) met daarin de variabelen uit de afspraak (.ics-bestand).
- De personen die in de week van het uitvoeren van het Perl script een afspraak hebben, moeten deze mail automatisch ontvangen.
- Het script moet ook een bestand kunnen genereren met daarin de klanten die al betaald hebben. Dit door in één of ander bestand de tekst PAID toe te voegen. *(Ik weer niet meer exact hoe dit deel in elkaar zat)*
- Extra: Hoe kan je er voor zorgen dat het script elke dag om 8u wordt uitgevoerd?

Python

Je moet voor 5 voetbalploegen bepalen welke de populairste is. Dat gebeurt aan de hand van het aantal hits ze elk hebben in een zoekmachine naar keuze. De populariteit kan dus in principe met de minuut wijzigen. Aan jou om de 3 populairste van de 5 op het scherm af te drukken.

Mondeling gedeelte

Je verdedigt je geprogrammeerde oefening van Perl en Python én je trekt 2 kaartjes, elk in verband met een regex:

- Leg de regex op kaart 1 uit

- Schrijf de gevraagde regex van kaart 2 uit (je krijgt hiervoor geen voorbereidingstijd vooraf, wel wat assistentie van de prof)

2012 juni examen

Lector: Frans Sanen

Hulpmiddelen: Oefeningen, Internet, je eigen laptop

Tijdens het jaar moet je oefeningen maken, dit staat op 8 van de 20 punten.

Oefening perl (6/20)

2 input files: 1 met rekeninguitreksels en 1 met verschillende categorieën aankopen. Je moest dan een financieel rapport genereren die per categorie de kosten weergeeft.

formaat categorie:

```
Categorie;Kenmerk1,Kenmerk2,...
```

voorbeeld:

```
VASTE KOSTEN;garage,tankbeurt,DATS,Lampiris,Scarlet
```

formaat rekeninguitreksel:

```
Rekeningnummer;Rubrieknaam;Naam;Munt;Afschriftnummer;Datum;Omschrijving;Valuta;Bedrag;Saldo;
```

voorbeeld:

```
BE26735012963329;;Fam. Geeft-Veeluit;EUR;2012012;01/05/2012;"BETALING AANKOPEN 01-05-2012 OM 12.22
UUR, COLRUYT 4145 MET KBC-BANKKAART 6703 73XX XXXX X303 3";03/05/2012;-16,93;608,69;
```

output:

Overzicht kosten mei 2012

VASTE KOSTEN 135,12 euro

WINKELLEN 65,50 euro

... ...

Oefening python (4/20)

je moet de Euromillions resultaten afhalen van de site en deze vergelijken met de nummers die je zelf ingeeft. Het programma moet teruggeven of je gewonnen hebt (meer dan 3 cijfers of cijfers en sterren) en welke cijfers je juist had.

2 regexen (2/20)

1 uitleggen, 1 schrijven zonder echte voorbereiding. (Je wordt wel op weg geholpen indien nodig :))

2013 juni examen

2013

lector: Frans Sanen

Permanente Evaluatie (8/20)

Willekeurige selectie uit de gemaakte oefeningen. Met andere woorden: geen risico lopen en alle oefeningen maken!

Oefening perl (6/20)

Een boodschap moet verborgen worden in een aantal mails. Deze kan via commandline worden meegegeven, indien niet het geval zal het script hier om moeten vragen. Vervolgens moeten er een aantal mails worden verstuurd (afhankelijk van de lengte van de boodschap). Deze mails worden opgevuld met willekeurige karakters (20 lijnen van 50 karakters). In elke mails wordt dan vervolgens een willekeurige plaats gekozen; op die plaats worden 5 opeenvolgende lettertekens gezet (bv. abcde) gevolgd door 10 karakters van de boodschap. Het aantal mails wordt zo gekozen dat elke mail een stuk van de boodschap bevat. Merk op dat de laatste mail dus minder dan 10 karakters kan bevatten, maar dit is niet erg.

Bonusopmerking: merk op dat op elke lijn een `\n` staat. Wanneer je de boodschap hier overschrijft zal er dus een newline verdwijnen. Dit kan op verschillende manieren worden opgelost, de eenvoudigste: er voor zorgen dat er altijd een index wordt gekozen zodat dit nooit gebeurt. Met andere woorden: willekeurige getal tussen 0 en $20 * 50$ (lijn) + willekeurige getal tussen 0 en 35 (-15 voor de mogelijke lengte van de boodschap + opeenvolgende letterrij).

Oefening python (4/20)

Uit een invoerbestand moet een lijst opgesteld worden van de woorden die het meest voorkomen in het bestand. Dit bestand, zowel als het aantal woorden dat moet worden getoond moet kunnen meegegeven worden via de commandline. Indien dit niet gebeurt (of deze parameters niet geldig zijn, vraagt het script de gebruiker om de juiste parameters in te geven. Denk aan inputcontrole (eventueel regex (whitelist)). Vervolgens bestand lezen en voor elk woord het aantal voorkomens registreren (pythons dictionary gebruiken). Sorteren (bv. `OrderedDictionary` Let wel: reversed) en ten slotte de top zoveel tonen (afhankelijk van het opgegeven aantal woorden).

(Deze opdracht is als oefening voorgekomen bij Perl tijdens het jaar)

2 regexen (2/20)

2 kaartjes trekken: 1 regex uitleggen, 1 regex opstellen Niveau: eenvoudig

2018 juni examen

lector: Frederic Vogels

Examen

Count courses

Je krijgt een JSON input in de vorm van

```
[  
  { "studentId": "r0000001", "courses": [ "MBI58x", "MBI12x", "MBI98x" ] },  
  { "studentId": "r0000002", "courses": [ "MBI58x", "MBI12x", "MBI98x" ] },  
]
```

Gevraagd is return een JSON in de vorm van { "r0000001": 3, "r0000002": 3 } Met 3 de lengte van courses.

Download

Je krijgt een lijst van ids, en wordt gevraagd de volgende output te returnen

```
if [ ! -d {id} ]; then  
  git clone https://github.com/{id}/project {id}  
else  
  (cd {id}; git pull)  
fi
```

voor {id} een id uit de lijst van id's.

Hash

Je krijgt een bestandsnaam als input, lees iedere lijn en vervang deze door de eerste 10 characters van zijn hash. Bv. als er staat a dan schrijf je 10 eerste hex-cijfers van SHA256(a).

Regex

Verander hoofdletters door _ gevolgd door kleine letter, Bv. isLarge --> is_large

Stats

Argumentparser oefening

Translate

Vertaal een Java klasse naar een Python OO klasse

2020 augustus examen

de opgave en oplossingen voor het 2020 augustus examen met dank aan ISW:

2020 juni examen

Het examen bestond uit 5 vragen, in totaal 23 punten. Het examen staat op 20 punten als je dus 10/23 hebt, ben je dus geslaagd. (20/23 = 20/20)

In `exam.zip` staan alle vragen zonder oplossing. En in `Oplossingen_Sigfried.zip` staan mijn oplossingen. Wellicht kan je sommige korter schrijven (mario-kart bv), zie het als een challenge voor jezelf!

Als je de bestanden download ga je ermee akkoord dat je hier geen misbruikt van zal maken, dit is enkel bedoeld om voor het volgende examen te oefenen!

Bestanden met dank aan ISW en Sigfried Seldeslachts: 20 Juni.zip

2021 januari examen

de opgave en oplossingen voor het 2021 januari examen met dank aan ISW: