

Syntra - Python Developer - eerste jaar

Examen 1 - Module Logisch Denken

Dit examen bestaat uit 2 vragen. De eerste vraag is een logische schakeling. Het resultaat schrijf je weg in een tekst bestand. (zie opgave)

De tweede vraag is een programmeer-opdracht. De opdracht wordt opgebouwd uit verschillende vragen. Je verdient voor elke vraag punten. Ook al geraak je niet aan de finale oplossing, door telkens een deel van de opdracht te maken, verdien je de nodige punten om te slagen.

Het is heel belangrijk dat je je code test. In elk van de bestanden zijn er reeds tests gemaakt. Je kan elk deel testen door de testen te runnen via “python naam.py”, waarbij naam.py het bestand is dat je wil testen. Of je kan uiteraard het bestand runnen in PyCharm. Heb je geen fouten in je code, dan zal dit commando niets melden. Heb je wel een fout, dan zal je zien wat er scheelt.

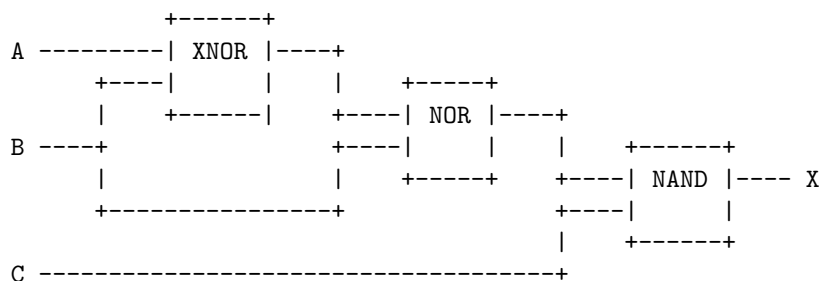
Commit na elke opdracht je code. Dat laat toe een evaluatie te maken en punten te geven. Geef als commentaar bij je commit “opdracht 2.x”. Vergeet niet te pushen.

Voor je het examen beëindigt, geef je door welke je laatste commit is. Dan kan met een pull gekeken worden dat alles effectief is opgeladen in GitHub.

Op het examen mag je handboeken en Internet gebruiken. Enkel hulp van en voor elkaar is niet toegestaan. Ook het gebruik van een telefoon is niet toegestaan. Microsoft Teams mag ook niet gebruikt worden tijdens het examen.

Vraag 1: logische schakeling (3 x 5 punten)

Gegeven: volgende logische schakeling:



Wat is de waarde van X, wanneer 0 = false, 1 = true, voor de volgende 3 gevallen:

- A = 0, B = 1, C = 1
- A = 1, B = 0, C = 1
- A = 1, B = 1, C = 0

De oplossingen van deze 3 vragen, mag je invullen in vraag1.txt.

Vraag 2: programmeer oefening

OPGELET: in deze oefening mag geen enkele Python library gebruikt worden. Er mogen enkel lokale bestanden uit deze folder gebruikt worden. Met name DateTime en Math zijn absoluut verboden. Voor de opdrachten zijn floor() en ceil() niet nodig.

Het doel van deze oefening is een kalender te maken voor het jaar 2022.

De uiteindelijke kalender zal geprint worden op 12 bladzijden, waarbij elke bladzijde een maand is. Hou geen rekening met layout of lettertypes. Het enige wat gemaakt moet worden is een tekst output die deze kalender dicht benadert. De “eindgebruiker” zal zelf de output “knippen” zodat de kalender lijkt op deze in calendar.jpg. De output moet op het scherm verschijnen wanneer “python calendar.py” gerund wordt.

Nota: de weken in de voorbeeld-foto zijn niet zoals gevraagd wordt in één van de laatste opdrachten. Het voorbeeld neemt de eerste volle week als week 1 van het jaar, terwijl gevraagd wordt in de opdracht om de week van 1 januari als eerste week van het jaar te aanzien. Wij wijken voor het examen af van hoe het week nummer eigenlijk bepaald moet worden, in functie van het vorige jaar. Dit is een vereenvoudiging enkel voor het examen.

Opgave 2.1 (5 punten) In het bestand leapyear.py, bevindt zich een functie “is_leapyear()” die als return waarde terug stuurt of een jaar een schrikkeljaar is of niet. Op dit moment is deze functie nog niet geïmplementeerd. Implementeer deze.

Als test kan je de test functie uitvoeren. Run “python leapyear.py” om te checken of je functie voldoet aan de eisen.

Opgave 2.2 (10 punten) In het bestand dom.py bevindt zich een functie “days_of_month” die als return waarde terug stuurt hoeveel dagen er in een maand zijn. Op dit moment is deze functie nog niet geïmplementeerd. Implementeer deze. Hoe rekening met een schrikkeljaar! Opnieuw is er een test voorzien om je implementatie te checken.

In Europa tellen we vanaf 1. Dus januari is de eerste maand en niet de nulde maand.

Opgave 2.3 (10 punten) Bekijk de functie in dow.py. Dit is de functie om de dag van de week te bepalen van onze Japanse vriend. Deze functie werd licht aangepast omdat in Europa maandag aanzien wordt als dag 1 van de week en zondag dag 7. Bij de oorspronkelijke functie was zondag dag 1 en zaterdag dag 6. Voorzie de code in regel 6 en 7 van de nodige commentaar (zoals hierboven geschreven) zodat iedereen de aanpassing snapt. Schrijf ook een commentaar header waarin je de nodige gegevens over auteur en dergelijke vermeldt. De commentaar moet bruikbaar zijn in de documentatie generator, dus maak deze

in “doc string”. Deze functie is de enige die je voor het examen van de volledige commentaar moet voorzien. Alle andere functies hoef je niet te documenteren met een doc string.

Opgave 2.4 (5 punten) Bekijk het bestand `date_string.py`. Dit bestand zal nodig zijn om de dagen van de week en de namen van de maanden in het Nederlands te tonen. Helaas, wanneer je “python `date_string.py`” intikt zijn er nog fouten. Corrigeer de fouten.

Hint: de testen zijn juist, de code niet.

Opgave 2.5 (10 punten) Schrijf een functie die van een bepaalde datum, berekent de hoeveelste dag van het jaar deze datum is. Deze functie is voorbereid in `day_number.py` maar niet geïmplementeerd.

```
1 januari == 1
1 februari == 32 (eerste dag van februari,
    we hebben reeds 31 dagen gehad in januari - hint)
31/12 == 365 of 366
```

Deze functie heb je niet strict nodig voor opdracht 2.6

Opgave 2.6 (20 punten) Nu we de hoofdfuncties hebben, kunnen we de kalender printen. Print de kalender voor deze maand (december 2022). Kijk naar het voorbeeld.

Deze kalender bevat:

Lijn 1: de naam van de maand plus het jaartal
Lijn 2: blanco
Lijn 3: de afkortingen van de dagen van de week,
te beginnen met "Ma" voor maandag.
Lijn 4: naargelang waar de maand start, de 1ste van de maand
bij de juiste dag van de week
Lijn 5-?: de rest van de kalender, tot het einde van deze maand

Als uitvoer print je gewoon je resultaat op de “command line”. M.a.w, gebruik gewoon de `print()` functie.

Opgave 2.7 (10 punten) Eén maand is leuk, maar een gans jaar is leuker. Schrijf de code om het ganse jaar te tonen volgens het principe van de vorige opgave.

Opgave 2.8 (10 punten) Nu we de basis kalender hebben, gaan we een stap verder. De kalender van Outlook bijvoorbeeld bevat ook de week informatie.

Implementeer het week nummer van een datum. Het week nummer begint bij 1 voor 1 januari en iedere maandag beginnen we een nieuwe week (in Europa). Het

kan dus best zijn dat 1 januari niet op een maandag valt en dat de tweede week niet zo evident is om te bepalen. Maar volgend Nasi-Schneidermann diagram zou een oplossing kunnen zijn. Implementeer de functie en schrijf de nodige testen.

```
+-----+
| bepaal de dag van de week van 1 januari van het jaar (variable dow) |
+-----+
| bepaal het dag number in het jaar van de datum (variable days) |
+-----+
+ bepaal het weeknummer volgens deze formule (dow - 2 + days) // 7 (variable week) |
+-----+
| tel 1 bij de variable week omdat we vanaf 1 tellen |
+-----+
```

De variable “week”, in het diagram hierboven, zou het week nummer bevatten. Test dit met het kalenderjaar 2022 via assert functies. Een aanzet is gegeven in week_number.py.

Indien je een meer efficiënte manier ziet, mag je deze uiteraard ook implementeren.

Opgave 2.9 (5 punten) Voeg het week nummer toe aan je kalender. Het week nummer komt tussen vierkante haakjes “[]” voor elke week. Zie het finale resultaat hieronder.

Finale resultaat

```

    Januari 2022

    Ma Di Wo Do Vr Za Zo
[1]      1  2
[2]  3  4  5  6  7  8  9
[3] 10 11 12 13 14 15 16
[4] 17 18 19 20 21 22 23
[5] 24 25 26 27 28 29 30
[6] 31

    Februari 2022

    Ma Di Wo Do Vr Za Zo
[6]      1  2  3  4  5  6
[7]  7  8  9 10 11 12 13
[8] 14 15 16 17 18 19 20
[9] 21 22 23 24 25 26 27
[10] 28
```

Maart 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[10]		1	2	3	4	5	6
[11]	7	8	9	10	11	12	13
[12]	14	15	16	17	18	19	20
[13]	21	22	23	24	25	26	27
[14]	28	29	30	31			

April 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[14]					1	2	3
[15]	4	5	6	7	8	9	10
[16]	11	12	13	14	15	16	17
[17]	18	19	20	21	22	23	24
[18]	25	26	27	28	29	30	

Mei 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[18]							1
[19]	2	3	4	5	6	7	8
[20]	9	10	11	12	13	14	15
[21]	16	17	18	19	20	21	22
[22]	23	24	25	26	27	28	29
[23]	30	31					

Juni 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[23]			1	2	3	4	5
[24]	6	7	8	9	10	11	12
[25]	13	14	15	16	17	18	19
[26]	20	21	22	23	24	25	26
[27]	27	28	29	30			

Juli 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[27]					1	2	3
[28]	4	5	6	7	8	9	10
[29]	11	12	13	14	15	16	17
[30]	18	19	20	21	22	23	24
[31]	25	26	27	28	29	30	31

Augustus 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[32]	1	2	3	4	5	6	7
[33]	8	9	10	11	12	13	14
[34]	15	16	17	18	19	20	21
[35]	22	23	24	25	26	27	28
[36]	29	30	31				

September 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[36]				1	2	3	4
[37]	5	6	7	8	9	10	11
[38]	12	13	14	15	16	17	18
[39]	19	20	21	22	23	24	25
[40]	26	27	28	29	30		

Oktober 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[40]						1	2
[41]	3	4	5	6	7	8	9
[42]	10	11	12	13	14	15	16
[43]	17	18	19	20	21	22	23
[44]	24	25	26	27	28	29	30
[45]	31						

November 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[45]		1	2	3	4	5	6
[46]	7	8	9	10	11	12	13
[47]	14	15	16	17	18	19	20
[48]	21	22	23	24	25	26	27
[49]	28	29	30				

December 2022

	Ma	Di	Wo	Do	Vr	Za	Zo
[49]				1	2	3	4
[50]	5	6	7	8	9	10	11
[51]	12	13	14	15	16	17	18
[52]	19	20	21	22	23	24	25
[53]	26	27	28	29	30	31	