

Rapporten

Inleiding

Om rapporten te maken voor studenten is een klein Python script gemaakt.

Zorg ervoor dat Python3 is geïnstalleerd, en dat via PIP de juiste library's zijn geïnstalleerd.

```
pip install pandas
```

Benodigde bestanden

studenten.csv

```
Nr;Klas;Canvas Id;Student nr;Naam  
1;3B;20576;2033543;Leonard Cranary  
2;3C;26328;2140079;Walid Sallek  
...
```

Dit bestand is een export van de Canvasmonitor waarbij niet relevante kolommen zijn weggehaald.

Dit bestand is leidend en de Id's worden gebruikt om een koppeling te maken met andere Excel lijsten.

De kolomnamen kunnen worden gebruikt om teksten in de template te vervangen. Zo kun je `_Klas_` in een template vervangen door de klas in dit overzicht.

template.docx

Dit is een Word-template ([template.docx](#)) waarbij de in-te-vullen velden worden gekenmerkt door een pre-fix en post fix, bijvoorbeeld:

```
_studentNaam_
```

Ander bestanden

Ander bestanden, presentie, blokken, kennis-check,.....

Afhankelijk van de gegevens die je nodig hebt kan je meerdere bestanden inlezen. Elk bestand dient een ID-kolom te hebben. Dit id moet het studentnummer bevatten dat overeenkomt met het studentnummer in de *studenten.csv*.

De bestanden worden ingelezen in het codeblok vanaf regel 51.

Vanaf regel 90 worden de velden uit de template vervangen.

Voorbeeld:

```
replace_text_in_doc(doc, "_kennis_", str(kennis.get(student_number, '???')))
```

In een van de ingelezen bestanden staat een kolom *kennis*. Deze wordt gematched uit kennis.csv via het student_number veld (uit studenten.csv).

Aanpassen in de code

Plaats de docenten en de klassen in de dictionary (regel 12 v/d code).

Code

```
import os, sys
# import chardet
import pandas as pd
from shutil import copyfile
from docx import Document

# Load the Excel file
basis_csv = './studenten.csv'
word_template = './template.docx'
presentie_csv = './presentie.csv'
kennis_csv = './kennis.csv'
canvas_csv = './canvas.csv'
base_output_folder = './'
docent = { '3A': 'Mina Ulas - Bicer', '3B': 'Muhammed Çağlı / Anton Boutkam', '3C': 'Max Bisschop' }

# Define the function to replace text in the document
```

```

def replace_text_in_doc(doc, old_text, new_text):
    for paragraph in doc.paragraphs:
        if old_text in paragraph.text:
            paragraph.text = paragraph.text.replace(old_text, new_text)
    for table in doc.tables:
        for row in table.rows:
            for cell in row.cells:
                replace_text_in_doc(cell, old_text, new_text)


def readExcel(excel_path):
    # with open(excel_path, 'rb') as file:
    #     result = chardet.detect(file.read())
    # print(f'Encoding {excel_path}: '+result['encoding'])
    try:
        excel_data = pd.read_csv(excel_path, index_col=None, sep=';', encoding='utf-8-sig')
        # print("Column names:", [col for col in excel_data.columns])
    except FileNotFoundError:
        sys.exit(f"The file {excel_path} was not found.")
    except pd.errors.EmptyDataError:
        sys.exit(f"The file {excel_path} is empty.")
    except pd.errors.ParserError:
        sys.exit(f"The file {excel_path} is probably not a CSV file or it is badly formatted.")
    except Exception as e:
        sys.exit(f"An unexpected error occurred in readExcel {excel_path}: {e}")
    return (excel_data)


def getDataFromExcel(file_name, column_name):
    print(f"Reading {column_name} from {file_name}")
    if not os.path.exists(file_name):
        sys.exit(f"File {file_name} not found")

    data = readExcel(file_name)
    if column_name not in data.columns:
        sys.exit(f"Error: Column '{column_name}' does not exist in the file.")
    if 'ID' not in data.columns:
        sys.exit(f"Error: Column ID does not exist in the file.")

```

```

dict = {}

for index, row in data.iterrows():
    dict[row['ID']] = row[column_name]

return(dict)

presentie = getDataFromExcel(presentie_csv, 'presentie')
kennis = getDataFromExcel(kennis_csv, 'kennis')
# praktijk = getDataFromExcel('./canvas.csv', 'praktijk')
blok1 = getDataFromExcel(canvas_csv, 'blok1')
blok2 = getDataFromExcel(canvas_csv, 'blok2')
# studieduur = getDataFromExcel(canvas_csv, 'studieduur')

cmon = readExcel(basis_csv)

for index, row in cmon.iterrows():
    print(f"Processing {row['Naam']}")
    person_name = row['Naam'] # Adjust the column name as necessary
    group_name = row['Klas'] # Adjust the column name as necessary
    student_number = row['Student nr']

    # Create a new file name with prefix "voortgang"
    new_file_name = f"voortgang {student_number} {person_name} {group_name}.docx"

    # Create a path for the new group folder if it doesn't exist
    group_folder_path = os.path.join(base_output_folder, group_name)
    if not os.path.exists(group_folder_path):
        os.makedirs(group_folder_path)

    # Define the full path for the new file within the group folder
    new_file_path = os.path.join(group_folder_path, new_file_name)

    # Copy the template to the new file name in the group folder

    doc = Document(word_template)
    replace_text_in_doc(doc, "_studentnaam_", person_name)
    replace_text_in_doc(doc, "_docentnaam_", str(docent.get(group_name, '???')))
    replace_text_in_doc(doc, "_studentnummer_", str(student_number))
    replace_text_in_doc(doc, "_presentie_", str(presentie.get(student_number, '???')))
    replace_text_in_doc(doc, "_kennis_", str(kennis.get(student_number, '???')))

```

```
# replace_text_in_doc(doc, "_praktijk_", str(praktijk.get(student_number, '???'))
replace_text_in_doc(doc, "_blok1_", str(blok1.get(student_number, '???'))
replace_text_in_doc(doc, "_blok2_", str(blok2.get(student_number, '???'))
# replace_text_in_doc(doc, "_studieduur_", str(studieduur.get(student_number, '???'))
doc.save(new_file_path)

print(f"File created for: {person_name} in group: {group_name}")

# sys.exit("Test")

print("All files have been created.")
```

Voorbeeld extra databestanden

(bovenstaande code werkt met deze bestanden).[template.docx](#)

presentie.csv (uit Eduarte)

```
Deelnemer;% Aanwezig;Nr;ID;presentie
2186011 - Yusuf Aksari;96;1;2186011;96
2132925 - Ibrahim Azarfane;13;2;2132925;13
...
```

canvas.csv (voor voortgang blokken, uit de Canvas monitor)

```
Rank;ID;Klas;Student;Studentnaam;Tot;blok7;blok8;Voortgang;studieduur;1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24;;praktijk;26;27;28;29;30;31;32;33;34;35;36;37;38;39;40;Naam  
7350;2184477;3D;Student;Giovanni Deniz;36;Ja;Ja;3653;3-  
;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;;0;100;  
0;100;100;100;100;0;100;100;13;100;100;100;100;100;40;GiovanniD|/public/index|code|935315184868c08dc18  
c67ff3f4d3645  
  
.....
```

kennis.csv (voor kennis-check)

```
ID;Naam;kennis
2202160;Alie Abduka;73,0
2197436;André Romburg;76,5
...
```

Stappen

Let op: Bewaar alle CSV-bestanden in het formaat: *CSV - UTF-8*

studenten.csv

Download uit de Canvas-monitor alles studenten. Menu (beheer - studenten en dan export).

Verander de naam naar *studenten.csv*.

Dit is het basisbestand.

Kolomnamen hoeven niet te worden aangepast.

canvas.scv

Download de voortgang uit de Canvas-monitor.

Noem de kolom met het studentenummer 'ID' en voeg kolommen toe blok1, blok 2,Zet in deze kolommen "ja" of "nee" afhankelijk of het blok is gehaald. Pas de code aan afhankelijk welke blokken je wilt afdrukken.

Formule: `=IF(SUM(<verwijzing naar blokken>)=300;"Ja";"Nee")`

presentie.csv

Voeg de presentie bestanden samen tot één CSV en bereken het percentage aanwezig.

Percentage aanwezig is: `=round(aanwezig/(aanwezig+ongeoorloofd+ziek)*100;0)`

kennis.csv

Maak een csv-bestand met ID (studentenummer) en zet het (gemiddelde) cijfer voor de kennis-checks onder de kop *kennis*.

Praktijk test

In 2024/25 nog niet getest.

Test run

Om te testen haal het commentaar uit regel 104. De code stopt na één rapport. Controleer het rapport en indien correct draai alle rapporten door regel 104 weer in commentaar te zetten.

--

Revision #8

Created 24 October 2024 11:49:22 by Max

Updated 26 October 2024 14:52:32 by Max