TITLE

des

na

Script:

```
from typing import List, Tuple
from fpdf import FPDF
import itertools
import random
import os
import string
import numpy as np
import re
from math import nan
from datetime import datetime
import sys

print('Hello there! This program lets you decide which categorys with its attributes you want to include in your collection.\nAfter all informations are collected a number will be appended to each item of its category.\nThen each attribute will appear this amount of times in an upgoing order.\nHave fun!')
print('As soon as all pages are printed it will be your task to stage yourself in front of the camera with the attributes given to you.\n')
print('At the very end these images are fed back into the program and become a new complete PDF (it could even be printed live!).\nHave fun!\n')

#Collecting first data
project_title = input("Enter your Project title: ")
if project_title == None:
    project_title = 'New Project'
short_description = input("Enter a short description: ")
author_name = input("Add your name: ")

#Size of collection
page_count = int(input('How big do you want your collection to be?\n(min. 50; max. 10000) Enter size: '))
if page_count == None:
    page_count = 100
if page_count > 10000:
    print('Please keep your collection smaller than 10000!')
    exit()
if page_count < 50:
    print('Please make the collection bigger!')
    exit()

categories_per_page = int(input('How many attributes or categorys do you want in your collection to be?\n(min. 5; max. 20) Enter size: '))
if categories_per_page == None:
    categories_per_page = 20
#page_count = 100
#categories_per_page = 3

#All categorys
category_attributes_map = {}

print('What attributes would you like to include in this collection?')

for i in range(0, categories_per_page):
    category = input(f"Enter category {i+1}: ")
    print(f"Enter all attributes of category {category} at index\n(seperate with a comma)")
    items = input()
    items_l = items.split(',')

    category_attributes_map[category] = items_l

    if len(items_l) < 5:
        print("Warning, you have not enough items, be creative!")
        exit()
    elif len(items_l) > 12:
        print("Warning, you chose too many items, please respect the limit!")
        exit()

#Code from Guillaume Jacquenot and pjs (Stackoverflow: edited Jan 23, 2020 at 20:45)
def random_sum_to(n, num_terms = None):
    num_terms = (num_terms or random.randint(2, n)) - 1
    a = random.sample(range(1, n), num_terms) + [0, n]
```

```
    list.sort(a)
    return [a[i+1] - a[i] for i in range(len(a) - 1)]

#Function to merge each item with a computed number (possibility of appearance) and put them in order.
#k = specifies the list of items for each category.
def filler(attributes: list) -> List[Tuple[str, int]]:
    new_list = []
    temp_list = random_sum_to(page_count, len(attributes))
    #Add random numbers to the new list and sort the list
    temp_list.sort()

    for i in range(0, len(attributes)):
        new_list.append((attributes[i], temp_list[i]))

    return new_list

# print(str(attributes[0]) + ': ' +  str(clean_up_list(n1)))
# print('\n')
# print(str(attributes[1]) + ': ' + str(clean_up_list(n2)))
# print('\n')
# print(str(attributes[2]) + ': ' + str(clean_up_list(n3)))


'''Function to save a PDF'''
def save_pdf(filename):
    path = r'dp_pdfs' # Make sure to create the specific folder in advance.
    pdf.output(path + filename)


# Function to extract the probability of an item
def extract_number(item):
    # Create a list of all numerical characters
    n = [e for e in item if e.isnumeric()]
    # Join numbers (characters) to one number
    n = ''.join(n)
    # Convert string to int
    n = int(n)
    return n

'''Empty lists to be filles'''

pdf = FPDF('P', 'mm', (220,405)) #297 mm x 210 mm = A4
pdf.set_margins(left=5, top=5, right=5)
pdf.set_author(str(author_name))
pdf.set_title(str(project_title))
pdf.add_font('nulshock_bd', 'B', r'dp_fonts\nulshock_bd.ttf', uni=True)

'''COVER'''
#First parameters for the cover
pdf.add_page()
pdf.set_text_color(0,0,0)

#Title
txt = str(project_title)
pdf.set_font('nulshock_bd', 'B', 40)
pdf.set_y(70)
pdf.multi_cell(w=0, h=0, txt=txt, align='C')

#Header
txt = str(short_description)
pdf.set_font("helvetica", "I", 12)
pdf.set_y(190)
pdf.multi_cell(w=0, h=10, txt=txt, align='C')

#Author
txt = str(author_name)
pdf.set_font("helvetica", "", 12)
pdf.set_y(310)
pdf.multi_cell(w=0, h=10, txt=txt, align='C')
pdf.ln()
```

```
#Description
#pdf.add_page()
#pdf.set_margins(left=20, top=30, right=30)
#pdf.set_auto_page_break
#pdf.set_text_color(0,0,0)
#
#pdf.set_font("helvetica", "", 12)
#pdf.set_y(30)
#pdf.set_x(25)
#with
open(r"C:\Users\2578932\Documents\Module_Bauhaus_Uni\Fachmodule\5_Semester\bpwp\pbwp_001\Party
People\dp_decription.txt", "r", encoding='latin-1') as f:    #utf-8
#    description = f.read()
#pdf.multi_cell(w=0, h=8, txt=str(description), align='L')
#pdf.ln()

#Render code
filename = sys.argv[0]
with open(filename, "r", encoding='utf-8') as f:
    txt = f.read()
pdf.add_page()
pdf.set_margins(left=20, top=30, right=20)
pdf.set_font_size(10)
pdf.set_y(25)
pdf.set_x(25)
pdf.multi_cell(0, h=5, txt="Script:\n\n"+txt, align='L')
pdf.ln()



'''functions to change font for description and Trait'''
#defines position of first and second row to make two blocks
def m_cell_left(name: str, attribute: str):
    pdf.set_font('helvetica', style='', size=11)
    pdf.set_x(14)
    pdf.multi_cell(0,3,txt=str(name + ': ' + attribute + '\n'), align='L')
    pdf.ln()
    #pdf.set_font('helvetica', style='I')
    #pdf.set_x(49)
    #pdf.multi_cell(0,7,txt=attribute, align='L')

def m_cell_right(name: str, attribute: str):
    pdf.set_font('helvetica', style='', size=11)
    pdf.set_x(104)
    pdf.multi_cell(0,3,txt=str(name + ': ' + attribute + '\n'), align='L')
    pdf.ln()
    #pdf.set_font('helvetica', style='I')
    #pdf.set_x(140)
    #pdf.multi_cell(0,7,txt=attribute, align='L')

for category, attributes_w_count in category_attributes_map.items():
    attributes_w_count = filler(attributes_w_count)
    attributes_multiplied = []
    for attribute, count in attributes_w_count:
        for _ in range(count):
            attributes_multiplied.append((attribute, count))

    category_attributes_map[category] = attributes_multiplied

for i in range(page_count):
    number = i+1
    pdf.set_font('nulshock_bd', 'B', 12)
    pdf.add_page()
    pdf.set_y(15)
    pdf.set_x(14)
    rarityy = f"{number} / {page_count}"
    pdf.multi_cell(0,5,txt=("Rarity Rank: "+ str(rarityy)))
    pdf.ln(5)
```

```
    #INSERT IMAGE
    #path =
r"C:\Users\2578932\Documents\Module_Bauhaus_Uni\Fachmodule\5_Semester\bpwp\pbwp_001\PartyPeople
\dp_images\image_"
    #img_ext = i+1
    #ext = ".jpg"
    #img_name = path + str(img_ext) + ext
    #pdf.set_compression(True)
    #pdf.image(img_name, x = bn, y = 25, w = 190, h = 0)

    idx = 0
    pdf.set_y(30)
    for category, attributes_w_count in category_attributes_map.items():
        idx += 1
        attribute, count = attributes_w_count[i]
        #if idx > 10:
            #m_cell_right(category, f"{attribute} {count}%\n")
        #else:
        m_cell_left(category, f"{attribute} {count}%\n")

    #page numbers
    pdf.set_font('nulshock_bd', 'B')
    pdf.set_font_size(40)
    pdf.set_y(371)
    #if number < 10:
    #    x_pos_n = 190
    #elif number >= 10:
    #    x_pos_n = 181
    #elif number == 100:
    #    x_pos_n = 172
    #pdf.set_x(x_pos_n)
    pdf.set_x(180)
    pdf.multi_cell(0,0,txt=str(number))
#END
pdf.add_page()
end_txt = 'Interactive reallife NFT-generator by Jasper Venter \n\nSummer semester 202'
pdf.set_font("helvetica", "", 15)
pdf.set_y(200)
x_pos_n = 20
#set_p(end_txt)
pdf.multi_cell(0,5, txt=end_txt, align='L')

#get timestamp and save pdf
now = datetime.now()
current_time = now.strftime("%H%M")
save_pdf(f"/{project_title}-{current_time}.pdf")

print('Your attributes are all set, time to get in front of the camera!')
```

RARITY RANK: 1 / 50

one: sddscdscsdcs 5%

two: dsds 1%

three: dvsv 1%

four: dsvsd 1%

RARITY RANK: 2 / 50

one: sddscdscsdcs 5%

two: cdssdsd 2%

three: sdvds 2%

four: vdsdsv 1%

RARITY RANK: 3 / 50

one: sddscdscsdcs 5%

two: cdssdsd 2%

three: sdvds 2%

four: dsvsd 2%

RARITY RANK: 4 / 50

one: sddscdscsdcs 5%

two: sdcsd 2%

three: vsdvds 3%

four: dsvsd 2%

RARITY RANK: 5 / 50

one: sddscdscsdcs 5%

two: sdcsd 2%

three: vsdvds 3%

four: vds 3%

RARITY RANK: 6 / 50

one: cssdsdcsd 6%

two: csdsc 2%

three: vsdvds 3%

four: vds 3%

RARITY RANK: 7 / 50

one: cssdsdcsd 6%

two: csdsc 2%

three: vdsv 7%

four: vds 3%

RARITY RANK: 8 / 50

one: cssdsdcsd 6%

two: ssd 5%

three: vdsv 7%

four: vds 5%

RARITY RANK: 9 / 50

one: cssdsdcsd 6%

two: ssd 5%

three: vdsv 7%

four: vds 5%

RARITY RANK: 10 / 50

one: cssdsdcsd 6%

two: ssd 5%

three: vdsv 7%

four: vds 5%

RARITY RANK: 11 / 50

one: cssdsdcsd 6%

two: ssd 5%

three: vdsv 7%

four: vds 5%

11

RARITY RANK: 12 / 50

one: sdcsdsd 8%

two: ssd 5%

three: vdsv 7%

four: vds 5%

RARITY RANK: 13 / 50

one: sdcsdsd 8%

two: cds 6%

three: vdsv 7%

four: vs 5%

RARITY RANK: 14 / 50

one: sdcsdsd 8%

two: cds 6%

three: sdvsd 8%

four: vs 5%

RARITY RANK: 15 / 50

one: sdcsdsd 8%

two: cds 6%

three: sdvsd 8%

four: vs 5%

RARITY RANK: 16 / 50

one: sdcsdsd 8%

two: cds 6%

three: sdvsd 8%

four: vs 5%

RARITY RANK: 17 / 50

one: sdcsdsd 8%

two: cds 6%

three: sdvsd 8%

four: vs 5%

RARITY RANK: 18 / 50

one: sdcsdsd 8%

two: cds 6%

three: sdvsd 8%

four: dvsd 7%

RARITY RANK: 19 / 50

one: sdcsdsd 8%

two: ds 9%

three: sdvsd 8%

four: dvsd 7%

RARITY RANK: 20 / 50

one: csddc 9%

two: ds 9%

three: sdvsd 8%

four: dvsd 7%

RARITY RANK: 21 / 50

one: csddc 9%

two: ds 9%

three: sdvsd 8%

four: dvsd 7%

RARITY RANK: 22 / 50

one: csddc 9%

two: ds 9%

three: vsdvsd 10%

four: dvsd 7%

RARITY RANK: 23 / 50

one: csddc 9%

two: ds 9%

three: vsdvsd 10%

four: dvsd 7%

RARITY RANK: 24 / 50

one: csddc 9%

two: ds 9%

three: vsdvsd 10%

four: dvsd 7%

RARITY RANK: 25 / 50

one: csddc 9%

two: ds 9%

three: vsdvsd 10%

four: vsdv 13%

RARITY RANK: 26 / 50

one: csddc 9%

two: ds 9%

three: vsdvsd 10%

four: vsdv 13%

RARITY RANK: 27 / 50

one: csddc 9%

two: ds 9%

three: vsdvsd 10%

four: vsdv 13%

RARITY RANK: 28 / 50

one: csddc 9%

two: dssdc 23%

three: vsdvsd 10%

four: vsdv 13%

RARITY RANK: 29 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvsd 10%

four: vsdv 13%

RARITY RANK: 30 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvsd 10%

four: vsdv 13%

RARITY RANK: 31 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvsd 10%

four: vsdv 13%

RARITY RANK: 32 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvs 19%

four: vsdv 13%

RARITY RANK: 33 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvs 19%

four: vsdv 13%

RARITY RANK: 34 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvs 19%

four: vsdv 13%

RARITY RANK: 35 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvs 19%

four: vsdv 13%

RARITY RANK: 36 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvs 19%

four: vsdv 13%

RARITY RANK: 37 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvs 19%

four: vsdv 13%

RARITY RANK: 38 / 50

one: sdc 10%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 39 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 40 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 41 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 42 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 43 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 44 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 45 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 46 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 47 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 48 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 49 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

RARITY RANK: 50 / 50

one: sdcdscdc 12%

two: dssdc 23%

three: vsdvs 19%

four: ds 13%

Interactive reallife NFT-generator by Jasper Venter

Summer semester 202