

DOCUMENTATION FOR USING PYTHON SCRIPTS FOR MAKING PPT FROM EXCEL SHEET

Contents

OVERVIEW.....	1
CREATING TEMPLATE PPT.....	2
CREATING LAYOUTS.....	3
NAMING PLACEHOLDERS.....	4
Create LayoutTest.xlsx.....	5
CREATING DATA EXCEL.....	5
CREATING .JSON DIRECTORY.....	6
File name:.....	6
Purpose:.....	6
Format:.....	6
Finding Slide IDs.....	7
UNDERSTANDING LAYOUT TEST.....	7
File name:.....	7
Purpose:.....	7
Logic:.....	7
UNDERSTANDING MAIN SCRIPT.....	8
File name:.....	8
Purpose.....	8
Logic.....	8
CONCLUDING REMARKS.....	10

OVERVIEW

The purpose of the entire process is to make a customized ppt from a lot of data, which is present in an excel, **within seconds**. Earlier it was done using photoshop. Drawbacks of photoshop is that its time consuming and requires a good amount of manual intervention. This doesn't allow last moment changes and also lot of time has to be invested.

The advantages of this process are that once the templates and excel are made, the script compiles in seconds. Surely this will allow last moment changes, only if the user has properly understood the script. Also by this only those student's details will be added who are present on day. This was a restriction using photoshop.

This document is being made in very detail. If the user is pretty well-acquainted with python, ppt, placeholders etc. then he/she need not go through the entire document line by line. Still an overview is must. But if the case is otherwise, then reading the document and using python documentation for things not mentioned here, would be needed.

It is strongly advised that you should understand the entire process thoroughly, as some changes will surely be required. For faster and proper understanding this document is made. Once you understand this, whatever changes you wish, the logic will more or less be same. (Also here commands are not explained or explicitly mentioned. It is left to user to find out what command perform what actions, and yes it will be important to have this understanding. If time permits, one may simultaneously type the main script again, not copy, so that he/she understand each line and the logic behind it)

The complete procedure can be summarized into 8 steps as shown below

1. Create a template ppt with layouts and naming the placeholders
2. Create an excel, with the placeholder names written in order as they are to be filled.
3. Create the data excel in the specified format.
4. Create .json file directory for ease.
5. Run layout test script.
6. Save as the output file as .xlsx
7. Run the main script
8. Delete first three slides (which were added in step 2) and do personalized changes wherever needed.

Each step is described in detail below. For scripts main logics is explained here. You can find detailed explanation of each step in the script itself, which is mentioned by comments.

Few technical information that was used/available while the scripts were prepared:

- IDLE 3.6 version was used to prepare and edit the python scripts.
- MS office 2016.

DISCLAIMER: If you are new to this kind of stuff, do not judge the entire method/process as something out of the box/ very difficult. If you are experienced in this domain, then you know this is not. It's just using simple commands and features of python to suit our task. Hence don't be demotivated by the length of this document. It's pretty easy, but may help.

CREATING TEMPLATE PPT

This is a power point presentation, which will act as a template, from which the main script will extract layouts to prepare the final presentation. **[REFER 'template.pptx', simultaneously while reading this section]**

Open power point





Go to view and open slide master



Delete the default layouts under the master slide. And add three layouts. As shown in the figure.

These three layouts will be our three layouts.

1. First layout for only names [**only names is referred to 'onlyNames' throughout**]
2. Second for two names per slide [**this is referred as 'double' throughout**]
3. Third for single name per **slide** [**this is referred as 'single' throughout**]

By default, the indexes of these layouts are referred by integers, **in the order they are arranged under master slide**, i.e. for referring

- **onlyNames use index 0**
- **double use index 1**
- **single use index 2**

(This is same as indexes used in arrays. This will be used in main script while calling the layouts, by these indexes, to add a slide)

CREATING LAYOUTS

Important thing here to understand is the concept of placeholders. You may google for more details, but for basic understanding they are similar to text boxes. All you data will be added in placeholders only.

Once the three layouts have been added, then follow the following steps

1. Delete the existing placeholders, if needed, and add the placeholders (option available in tool bar only, as 'insert placeholder').
2. Use text placeholders for text and picture placeholders for photo of student. **(Photo will not be inserted in any other type of placeholder)**
3. **You can and should, given all alignments and font specification to placeholders itself, so they are incorporated for every slide.** Also see that the max amount of text can be entered or not. Otherwise change the size. (This can be done once you have done a test run on the script)
4. Insert placeholders and arrange them as required.
 - a. For **onlyNames**: Here, **7 text placeholders** were used. Align them to center.

- i. Choose font and font size.
 - ii. Align the text to center
- b. For **double**: the layout was divided into two parts. (refer file)
 - i. Add placeholders for text and picture.
 - ii. For Data, for e.g. if it was to be displayed "State board %: 99%" in one line, so this was divided into two parts, P1 - "State board %:" and P2 - "99%". So P1 was given one placeholder and P2 another. Also, text in P1 was aligned to right and in P2 to left. This was alignment was done, otherwise it will become very random.
 - iii. The above step was done for both left and right section.
 - iv. Choose appropriate font and font size for each placeholder individually.
- c. For **single**: this was done same as one side of 'double'.

The number of placeholders for double and single were assumed for following data

1. SB%
2. SB PCM%
3. Mains rank
4. Mains category rank
5. Advanced Rank
6. Advanced category rank
7. BITS

Remember the layout is made with provision for all these data, even though all students will not have all the fields valid for them. For this a logic is used in main script which will only add those fields in the slide which have some value for them. (this is also the reason why we are not directly writing the above categories into the master layout and adding only data using script)

So, for a student (in double and single) there are $7 \times 2 = 14$ placeholders for data + one for name and one for picture. So, a total of **16 placeholders for single** and **32 for double**.

Once these steps are done, the layouts are ready for next step.

NAMING PLACEHOLDERS

Once the layouts have been created the next major and important step is to name the placeholders. This is required as python, will use the placeholder when we provide the id, which is unique to each placeholder.

These ids cannot be seen directly. They extracted from a xml file. Hence to identify the placeholder there, they have to named uniquely. [Don't worry, we will not be extracting the ids manually. For this another python script, [LayoutTest.py](#), will be used, which will directly get these ids and store them into a excel file in the order specified by you.] [Refer Understanding Layout test section for details]

Follow the steps below for naming

1. Close master slide. And delete all existing slides, if any.
2. In the home tab, you'll find an option 'New slide'. Use the layouts from the drop down to add 1 slide of each type. This will create 3 slides.
3. Then in the home tab go to editing section and click on select dropdown. Then select selection Pane. This will show you the default names of each

placeholders. Select the placeholder, and the corresponding name will get highlighted.

4. Now for each slide/layout, name the according to your convenience, so that you can know, after getting ids for each, in which order you have to fill them.
 - a. For **onlyNames**: they were simply names N1 to N7, from top to bottom. i.e. N1 was top most placeholder and needs to be filled first.
 - b. For **double**: (See the file for better understanding, click on each name and understand the rationale behind the naming). L and R were used to specify left and right. Picture and name placeholders were named accordingly. D was used to specify data. E.g. L2D mean 2nd row of data on left side. Also each row was divided in two part A and B. This means LD1A will be filled first, then LD1B then LD2A and so on. Then right side will be filled.
 - c. For **single**: simple D1A and D1B pattern is followed.
5. After this save the ppt with appropriate name.
Next step is to create an excel for layout test.

Create LayoutTest.xlsx

- a. It is an excel, which basically contains 3 sheets, with names of placeholders **written in order they are desired to be filled**, from top to bottom. This order is to be used in the main script. **THIS HAS TO BE DONE MANUALLY**. (excel features like “and operation” can be helpful)
- b. The order in which they are to be filled are quite clear, but if any confusion is there refer the LayoutTest.xlsx.
 - i. onlyName : top to bottom
 - ii. double: Left - top to bottom (for data A Then B) then Right - top to bottom.
 - iii. Single: top to bottom
(picture is filled first then name)
- c. Name First sheet as '**onlyName**'
- d. Second sheet as '**double**'
- e. Third sheet as '**single**'

CREATING DATA EXCEL

This is the crucial part of entire process. The data of students has to be organized in the excel according to the specified format, because the main script is made according to it. **Surely this might not suffice and the user has to incorporate changes according to the need.** Hence understanding this section and the part of the script for filling this data is necessary so that changes according to the need can be made. **[REFER 'data.xlsx']**

- See the order in the excel.
- Second column is important. It will specify which type of layout has to be used for the particular student.
 - o 3: onlyName layout
 - o 2: double layout
 - o 1: single layout**This has to do nothing with layout index discussed above in creating layouts.**
- Third column marks 'p' (small) if the student is present. **If the entry is anything other than 'p' (a or blank etc.) the value will not be entered. This data has to be carefully entered from on-day attendance.**

- Then columns are to be inserted according to the order you desire to display. Here the order was SB, SB pcm, Mains, Mains cat, Adv, Adv cat, Bits. (Major changes might come here only)
- Last column contains the path of photo of the respective student. **[They should be in same folder]** [for those students of whom we do not have photo, use a smiley or blank image and put its path for all such cases.]
- **For data that is not present and/or you don't wish to display, put a '0' at that cell.** The script is designed such that, if it gets a 0 in a cell, it would not add that data to slide.
- **Once this is ready, sort the excel using layout type in descending order (3 2 1).**
[This is extremely important, because according to present script it will check this first for 3 then 2 then 1. If it is not sorted then if a 2 or 1 comes then all 3s below will never be added]

Once these steps are done, your data excel is ready.

CREATING .JSON DIRECTORY

File name: "configuration.json"

Purpose: This basically contains value that will depend on the design the user will choose for ppt. All you need to do is that define these values here. And even if u need to change some value (say for example no of place holders, which are used multiple time) you only have to do it here, and it will be updated, while the scripts will be used. **This saves time finding and editing these values in script.** (If for some reason, which cannot be seen now, if this doesn't work, then you can directly enter the values at respective places. Just be careful)

Format: [REFER TEXT FILE: configuration.json] Here explanations are added, wherever necessary.

```
{
    TemplatePPT" : "t2.pptx",           #name of the template ppt
    "LayoutTestExcel" : "LayoutTest.xlsx", #name of the LayoutTest excel
    "slideID_onlyName" : 257,           #(See below for finding the
slide ids)
    "slideID_double" : 260,
    "slideID_single" : 259,
    "No_of_placeholders_onlyName" : 7,   # no. of placeholders present
    "No_of_placeholders_double" : 32,    in the respective
    "No_of_placeholders_single" : 16     template.
    "LayoutTestOutput" : "LayoutTestOutput.xlsx" #name of the layout test
output excel
}
```

- Use notepad to type
- String to be kept in ""
- Integer written directly
- Variable and value separated by ' : '
- After end of each line ' , ' has to be inserted
- **Save the file with .json extension**

Finding Slide IDs

- These are used in Layout Test.
- Once the master layouts are created, add one slide of each layout. After this name the placeholders, with suitable nomenclature which helps to understand the order. **[More details mentioned in Creating Template]**
- Then select the slide in slide sorter view or just move to it in normal view then run this macro:

```
Sub ShowSlideID()  
MsgBox ActiveWindow.Selection.SlideRange(1).SlideID  
End Sub
```
- For this, after selecting the slide go to 'view' section in toolbar. Select 'Macros'. Enter a name and click on create. Then just paste the middle line of above code, in the new window that pops up.
- Run the macro to find the id. Once you have created one macro, for other slides simply select the slide and click on macro option and then select your previously made macro.
- **Enter these values in the configuration.json file.**

UNDERSTANDING LAYOUT TEST

File name: "LayoutTest.py"

Purpose: This will give us the placeholders id, which will be required by the Main script to populate the text boxes by extracting them from master slide layouts. **The output of this script will be in the form of an excel(LayoutTestOutput.xls) with names of placeholders arranged in the order as they are to be filled and with their respective ids written in front of them.**

Logic:

1. Json directory is used for ease.
2. **LayoutTestExcel** has to be created. **[Refer the excel: LayoutTest.xlsx]**
3. Opening a new workbook which will be later saved as **LayoutTestOutput.xls**
 - a. **THIS WILL BE DONE AUTOMATICALLY**
 - b. Three sheets will be added, to the new excel, also named 'onlyName' 'double' 'single', and according to order of placeholders as defined by user in **LayoutTestExcel** along with their respective placeholder ids.
 - c. Understand the logic by going through the script. It's simple. All that is done after extracting the placeholder name and id from slide, it is compared to the name written in **LayoutTestExcel**. Then these two values are written in the new excel. They will also be printed on screen, just to cross check the excel.
 - d. One important point to note is that the 3 slides contacting the three layouts are called by their ids. **[Refer Finding out ids of slides under Creating json directory]**

Once this is done open the LayoutTestOutput.xls and save it as LayoutTestOutput.xlsx

This is necessary because of the openpyxl module doesn't work with xls format. You may find proper library to avoid this step. Otherwise its working fine.

UNDERSTANDING MAIN SCRIPT

File name: "script.py"

Purpose: This is the main part of the entire process. Understand each and every line of this code. So that you can do the necessary changes according to your need.

Logic: I would try to explain this part as elaborately as possible. Hope you don't get bored reading it. I would also like to remind you that the commenting in the script in itself is sufficient. This doc is being created for extra information and explanation if one requires. (another reason is for the sake of completeness of this document :p)

Also I would suggest that you first try to understand it on your own and then read the following elaboration. Reason that you will enjoy more and more importantly you won't get bored reading this. Also you'll be able to read it more quickly if you have even once tried understanding the script on your own.

So here's the explanation (same order as in script.py):

- All the necessary modules were imported. Directory was configured and variables were assigned the value. Excel and templates were opened and assigned to variables.

Starting with "FOR ONLY NAMES" which is highlighted in the script.

- Print statements are used, so that if any error occurs, by seeing the terminal window you can see how much your code was executed.
- A local array k[] is created. What it does is that it stores the value of ids that are used to address the placeholders on this type of layout in the exact order they are to be filled.
- This is done using the command

```
k.append(onlyName.cell(row = z, column = 2).value)
```

where onlyName is the excel that you get as the output of layout test. (Hope this is clear what exactly is being done. If you still have any doubts try executing the layout test and see the output.)

Just mentioning again, why is this is being done or why layout test is helpful. (if you are clear with the layout test and its need, you can jump to next point.)

To make a ppt you need data and you need a text box where it can inserted. Now the data is being extracted from the excel that you have prepared. But to know where it has to be inserted, you need to tell the computer the address of that text box, where you wish to enter the data.

This address is unique for every placeholder, that you have made while creating master templates. Now how do you get that unique address/id? For this one direct approach is that you open the xml type file of your layout and then out of some 100-200 lines of code manually search the id. (you can do this by saving your .pptx file as .zip, then opening it WinRAR, then going to slide layouts. There you will see your templates xml files. Open the files with any browser. There you will find the names of the placeholder that you gave in power point and in front of them their unique id) Hope this clarifies why it is so time consuming to take out the ids manually. Hence the layout test will do this for you.

Now another question is why is it important to make an excel with the names in order that we want to be filled. (why to create Layout test.xlsx?) Can't we directly get the ids and store it in array? The answer is no, because the order of IDs that you will get through the layoutTest.py **may not** be the same as they appear. Also they have to be stored in an array because they do not follow any order which can be used to generalize them.

- Once the placeholder's ids are stored in the array, it is convenient to point them generally while filling the data. Its use is more valuable when we use for 2 student or 1 student per slide and when we don't want to display any particular column from the data excel.
- Now the data of only names is going to be added. For this, infact for the entire script, its important that the format of data excel is same. So first it is checked from the excel if this is the layout for the student.
- Then it is checked if the student is present or not.
- Then the counter i counts how many students are added. The logic is that a new slide has to be added for every 8th student. (here we have decided to display only 7 names per slide. you may choose the number according to the screen etc)

Now moving to "two students per slide"

- Here again the placeholder ids were stored in local array. Two arrays were used, for left side and right side. See and understand the logic.
- As mentioned above layout value and present or not is checked for each student.
- If the student is present, then its data is extracted from the excel and stored in variables having the corresponding names.
- Now the data is added, before that it is ensured that a new slide is added for every odd student, using logic same as above.
- Now here is the crux, why we are using array instead of directly specifying the values of placeholder's id. Say if you do not have the data of state board marks for one student, and hence you don't wish to display that field, as showing NA looks very unprofessional. So what we are doing is that first we are checking if the value is not equal to 0. If it is this means that we don't wish to display and hence we can move to next column. But our placeholder can't be left blank. So if we used directly the ids rather than array values to address the placeholder, then gap would appear on the ppt. but using an array allows you to keep the same placeholder for which ever next value of data you have.

Moving to "Single student per slide"

- It is using the same logic as for two students. Only difference that you have to add slide for every student.
- Finally, the presentation is saved as output.pptx

CONCLUDING REMARKS

This document as mentioned above was just to explain the entire process. It is pretty sure that one cannot directly use the entire process as it is. Few modifications will be required. For this it is important that one understands the scripts and process thoroughly. Again mentioning the method is not at all difficult or some "really smart stuff". It's one of the simple use of python and features of power point. So in case you are feeling its too complicated or something extra ordinary is being done, then you are wrong. It may appear little long, maybe because of this document, but believe it is not. All you need to do is given time and try to do it on your own.

You may use some better logic or steps to make it more user friendly. One improvement which can surely be done is finding right syntax or modules in python that don't require user to save the .xls file to .xlsx.

Also please correct any grammatical or technical errors.

ALL THE BEST.