So you want to make a DPOP? Fair enough, but who has the time? Following this documentation you can make a DPOP and save a bit of time on the way!

**How it Works**

Makes a DPOP for you. A bit more useful: it looks at the DPOP shell and tries to fill it in with the major requirements for a given day. It assigns mandatory shifts like HWU, Dinosaurs, and daily shows. For each shift to be assigned it randomly selects a worker from the shell and checks to see if they are available for the shift, then sees if they are trained on it, and then sees if there is a reason not to assign it to them (avoids double HWU, Dinos to HWU, ect.) Assuming those requirements are met it then assigns the shit and goes on to the next time needed to fill, otherwise it randomly selects a different worker.

AutoPops uses a brute force approach to DPOPing, going shift by shift and randomly selecting people to assign things to. As a result it can take a few times to get a valid DPOP. Currently the program starts by tring to assign all the major requirements for a day in this order: Shows => Open shifts => Required Shifts => Closing Shifts. After going through all those it will check to see if all the required shift are assigned (things like Dinos, AMNH, ect). If everything if there the program goes on to assign LKS shifts.

If the EPT DPOP is missing something the program starts over from the beginning. If after 500 tries no solution has been found that meets all the requirements AutoPops changes its approach a bit: it stops trying to assign shows and focuses on the other shifts.

After the EPT shifts are assigned AutoPops assigns LKS shifts, in much the same way.

**Requirements**

AutoPops runs in Python 3.7 and in addition to the standard library will require the package: openpyxl.

AutoPops also requires a well maintained database that is up to date.

**Limitations**

Currently AutoPops is able to create DPOPs for the following: Evening Events, extended hours, early mornings.

**Running the Program**

To run the program and create a DPOP go to the scripts folder and right click on ‘buildDpop’. A window will open and after the program loads a prompt will ask what dpop you are making. Type in the full file name (E.G. 03.14.19 Thurs) and hit enter. The program will run and when it is finished the window will close. This process can take a while so I recommend running it before you leave for the day or during a meeting. After the program is finished, it saves the DPOP to the Finished folder. I highly recommend looking through it to see what changes are needed and ensure everything was assigned (basically like you do in the morning as point).

**Choosing What Shifts to Schedule**

**Defining Shifts**

Example of an existing shift:

amnh = Shift('Mythic Creatures', “”, 4, ['HWU', 'Dinosaurs', 'Cafe'], '00339966', 'Temporary Programs', 9)

Let’s break that apart a bit. In the generic a shift is defined like this:

amnh=Shift(name, clearance, long, restrictions, color, database, rowKnow)

amnh is what the program will use to reference the script when making a DPOP.

Name is what the shift is called on the DPOPs. The current ANMH exhibit is “Mythic Creatures”.

Clearanceis the fire clearance for the shift. If there is no non-obvious clearance leave this as “”.

Long is how long the shift lasts in rows, since the AMNH shifts are 1 hour long they last 4 rows in Excel.

Restrictions is where you define what the shift can’t be scheduled around, and it looks insdie the brackets. In this case we want to avoid HWU/Mythic swaps, Dinos/Mythic swaps, ect.

Color is the color for the background of the shift. It’s a simple hex code, so you can find the color code with a quick google search.

Database is what database to check for assigning things. In this case AMNH exhibits are in the “Temporary Programs” database.

rowKnow is what column Python will check to see if a person is trained, this needs to be a number. In the database column “i” is the column that holds that data we want, so it is column 9.

**Creating a new shift**

Let’s say we have a new traveling exhibit coming, so we need a new shift assignment. Open up buildDpop.py with an editor (IDLE is a fine choice). This is where we define shifts and choose what to include in the dpop.

Once the editor opens up scroll down to Section 2: Defining Shifts, then go to the appropriately labeled sub section (LKS, EPT, Ect)

Since traveling exhibits are staffed by EPT shifts I’ll add the new shift to the EPT section.

It’s a traveling exhibit so I call it “travel” in the program. It is a new shift so I tell the computer it is a Shift(capitalization matters to Python) and inside the parenthesis I add all the data I need.

travel = Shift()

The name is Crocodiles, so on the DPOP I’ll call it Crocs. travel = Shift(‘Crocs’)

The person in crocs are responsible for clearing Ocean and Gallery One so clearance is “G1 & O”. travel = Shift(‘Crocs’, ‘G1 & O’)

It lasts an hour so the length is 4 rows on the DPOP. travel = Shift(‘Crocs’, ‘G1 & O’, 4)

There are no scheduling restrictions so we have empty brackets []. travel = Shift(‘Crocs’, ‘G1 & O’, 4, [])

The background color for this shift on the DPOP is a light blue hex code '0099CCFF'. travel = Shift(‘Crocs’, ‘G1 & O’, 4, [], '0099CCFF' )

The database that stores our traveling exhibit training data is “Temporary Programs”. travel = Shift(‘Crocs’, ‘G1 & O’, 4, [], '0099CCFF', ‘Temporary Programs’ )

And the column to check in that database is ‘H’ so numerically that is 8. travel = Shift(‘Crocs’, ‘G1 & O’, 4, [], '0099CCFF', ‘Temporary Programs’ , 8)

End result is: travel = Shift(‘Crocs’, ‘G1 & O’, 4, [], '0099CCFF', ‘Temporary Programs’ , 8). This will define a shift but not actually add it to the DPOP when running the program. Also this is only good for a shit that will run all day the same way.

You should define shifts in the buildDpop.py file, this is also where you choose what shifts to include in the dpop when you make it.

**Choosing What to DPOP**

Now that you can create new shifts you will want to make sure the new shift is included when the program runs. To add the new shift to the dpops scroll to section 3: Making the DPOP, subsection build the main shifts. Then add buildShift(travel) to the list, maintaining the style and formatting of the other shifts.

Now when you run the script the new shift will be included in the program!

To remove a shift from the list either delete it from the build section, or put a # in front of that line (maybe that exhibit is closed for an event so you don’t want it, but you also don’t want to delete it yet)

**Creating a New Show Shift**

Let’s take a look at what a show shift looks like.

chemLivePm = ShowShift('Chem Live', '0000CCFF', '1:30p', '3p', 'Show Training', 8)

Again we will break this down a bit: name = ShowShift( name, color, shiftStart, shiftEnd, database, rowKnow). Most of this is the same as a standard shift. The differences are: there are no restrictions (you can easily do a HWU=>chem swap), there is no fire clearance section, and there is a start and end time for the shift (this is for the whole shift, not just the show).

**Assigning a show shift**

Assigning a show is very similar to assigning a regular shift. In section 3: Make the DPOP of buildDpop.py the subsection for shows is one of the first. There just add the show to build like this: buildShow(chemLivePm). One thing to note about adding shows: because they are more flexible and can be DPOPed at different times as needed, and because the pool of potential performers is smaller than the pool for something like Dinos, AutoPops assigns shows first and then all day shifts. Also because some days are very tight staffed AutoPops only tries to assign shows and exhibit shifts for the first 500 iterations, and if a valid DPOP cannot be made then it stops trying to add shows and only adds exhibits.