

Python для тренинга EAP

Изучение Python перед прохождением тренинга **Extron Authorized Programmer** чрезвычайно важно, если вы хотите получить максимум от тренинга. Как мы уже говорили, во время тренинга мы не будем изучать основы Python. Поэтому мы хотим, чтобы перед посещением курса EAP, вы приобрели навыки программирования на Python, и это подразумевает не только умение читать и понимать код, но и написать его с нуля.

```
print('Hello World')
```

Хотя код сверху и является базовой программой на Python 3, но этого на самом деле недостаточно для наших целей!

Чтобы помочь вам в освоении основ Python мы выбрали несколько упражнений, которые могут быть выполнены на любом ПК без какого-либо дополнительного оборудования. В зависимости от вашей уверенности в своих знаниях, вы можете либо выполнить эти задания с нуля, либо воспользоваться прилагаемым GUI шаблоном, чтобы не тратить время на изучение графических библиотек.

Что мне нужно, чтобы начать программировать на Python?

Вам необходимо установить Python 3.x (последнюю стабильную версию) на ваш ПК. Дистрибутив под нужную ОС (Win/Mac/Linux) можно найти по ссылке:

<https://www.python.org/> (Также один из лучших ресурсов по Python)

В стандартном дистрибутиве есть встроенная среда разработки (называется IDLE), однако помимо нее есть еще множество доступных и более функциональных вариантов, которые возможно понравятся вам больше. Мы рекомендуем попробовать Wing IDE (101 версия) или PyCharm – обе полностью бесплатные. Есть множество других вариантов, например Eclipse (с плагином Pydev) и даже Visual Studio. Некоторые из них бесплатны, за другие необходимо заплатить (естественно получив при этом дополнительный функционал). Функционала бесплатных версий более чем хватит для наших целей.

<http://www.wingware.com/>

<https://www.jetbrains.com/pycharm/>

<http://pydev.org/>

<https://www.visualstudio.com/en-us/features/python-vs.aspx>

Как только вы установили Python и выбрали в среде разработки **режим для Python 3** – вы готовы начать.

По завершению упражнений, пожалуйста вышлите их нам, мы их проверим и вернемся к вам с результатами и рекомендациями. .

До тренинга осталось не так много времени как кажется, так что не откладывайте эти упражнения на потом. Лучший вариант – зарезервировать несколько часов в неделю на выполнение упражнений. В списке достаточно упражнений, чтобы делать их по одному в неделю. Так вы

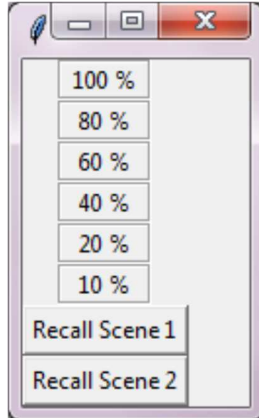
получите практические навыки в Python и не успеете их забыть до начала курса, что и является основной целью этой подготовки.

Удачи, и не стесняйтесь спрашивать, если вам нужна дополнительная информация, помощь и поддержка.

Упражнение 1 – Пресеты для света

GUI:

Используя Tk создайте 6 текстовых окон для отображения каналов управления светом, которые будут показывать уровни в процентах, т.е.: 100%. Также создайте 2 кнопки с названиями “Preset Recall 1” и “Preset Recall 2”



Функционал:

По нажатию “Preset Recall 1” уровни (сохраненные в двумерном списке) выводятся в текстовые поля. Другой набор уровней выводится в те же поля по нажатию кнопки “Preset Recall 2”.

Hint:

Воспользуйтесь шаблонным проектом (project 1 GUI.py) для знакомства с Tk.

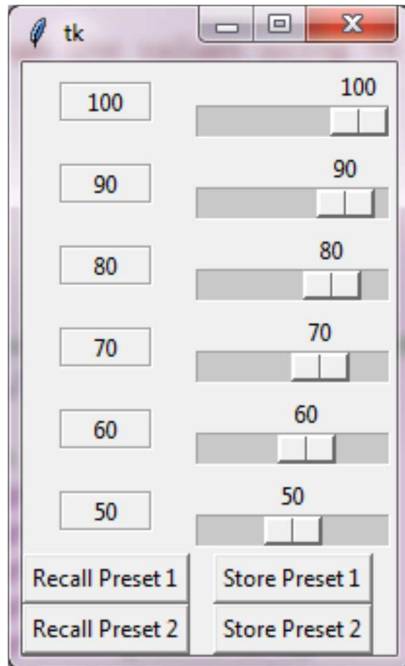
Бонусная задача:

Используйте циклы чтобы сделать код более компактным.

Упражнение 2 – Добавить управление для пользователя

GUI:

Используйте GUI из предыдущего упражнения и добавьте в него возможность для пользователя изменять уровни света. Это может быть сделано с помощью кнопок UP & DOWN, либо с помощью виджета TK Scale.



Можно сделать с кнопками UP & DN вместо слайдера...

Функционал:

Кнопки “Preset Recall 1” и “Preset Recall 2” вызывают значения, сохраненные в двухмерном списке. Изменения, вносимые пользователем, должны также отображаться в текстовых полях. После реализации управления светом добавьте еще функционал сохранения измененных пресетов.

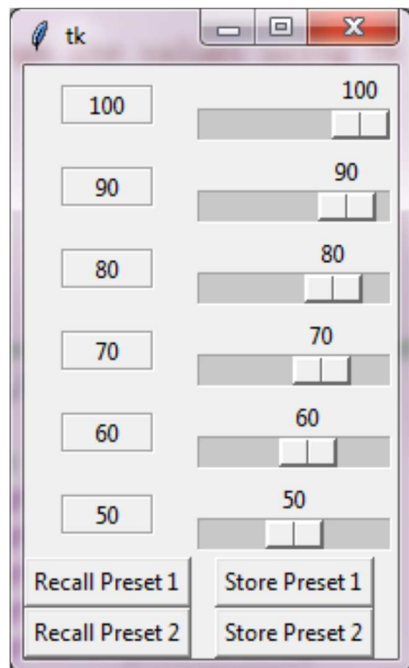
Hint:

Используйте онлайн помощь для Tk, чтобы разобраться с работой виджета.

Упражнение 3 – Запись пресетов на диск

GUI:

Используйте GUI от предыдущего упражнения.



Можно сделать с кнопками UP & DN вместо слайдера...

Функционал:

Когда пользователь сохраняет пресет, запишите значения уровней в файл на диск, чтобы при любых изменениях в системе можно было восстановить из файла ранее сохраненные пресеты.

Hint:

Вы можете либо написать обработку файлов вручную, либо использовать json для сохранения переменных в файл. Для управления файлами потребуется импортировать библиотеку os.

```
import json
import os
```

Упражнение 4 – PIN код

GUI:

Используйте Tk чтобы создать цифровую клавиатуру на подобие этой:



Функционал:

Используя дефолтный PIN '1230' проверьте корректность введенного PIN. Выведите сообщение для пользователя в соответствии с результатом проверки (Верно/неверно). Во время ввода PIN кода отображайте в текстовом поле звездочки вместо вводимых цифр.

Hint:

Вы можете воспользоваться готовым GUI файлом для этой задачи (Exercise4 GUI.py).

Бонусное задание:

Используйте циклы, чтобы сделать код компактнее.

Упражнение 5 – Добавьте смену PIN кода и сохраните его в файл

GUI:

Используйте GUI из предыдущего упражнения.



Функционал:

После ввода правильного PIN кода дайте пользователю возможность ввести новый код, после этого сохраните его в файл.

Hint:

Вы можете либо написать обработку файлов вручную, либо использовать json для сохранения переменных в файл. Для управления файлами потребуется импортировать библиотеку os.

Бонусное задание:

Добавьте возможность сохранять PIN коды для разных пользователей. Добавьте запрос ID перед вводом PIN кода.

Упражнение 6 – Сделайте хранение PIN кода более безопасным.

GUI:

Используйте GUI и код от предыдущего упражнения и добавьте в него шифрование PIN кода используя библиотеку hashlib.



Hint:

```
import hashlib
```

Бонусная задача:

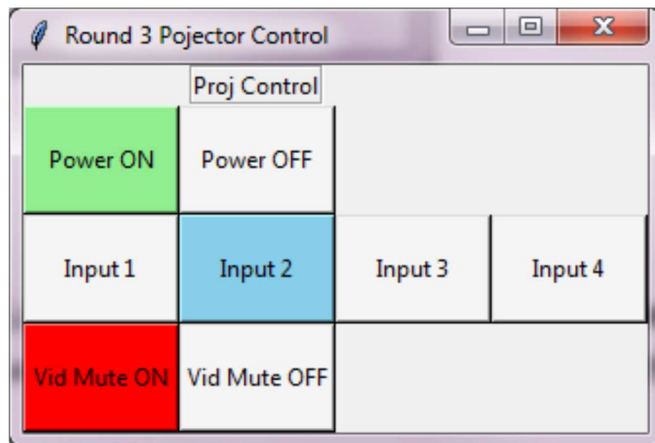
Добавьте “соли” к программе PIN кода, чтобы улучшить безопасность системы.

[https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BB%D1%8C_\(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F\)](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BB%D1%8C_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F))

Упражнение 7 – Управление проектором, используя класс в модуле.

GUI:

Используйте Tk чтобы создать панель управления проектором.



Функционал:

Используйте прилагаемый модуль (proj.py) чтобы создать программу для управления проектором.

Модуль содержит класс, который эмулирует управление проектором и будет возвращать обратную связь для команд:

```
projStatus = ProjectorName.control('PWR_OFF')
print('projStatus')
>> Power OFF
```

Hint:

Методы в proj.py:

function('PWR_ON') включает проектор и возвращает статус 'Power ON'

function('PWR_OFF') выключает проектор и возвращает статус 'Power OFF'

function('INP_1-4') изменяет вход в диапазоне 1-4 и возвращает статус в виде 'Input 1'..

function('PMT_ON') блокирует видео и возвращает статус 'Picture mute on'

function('PMT_OFF') разблокирует видео и возвращает статус 'Picture mute off'

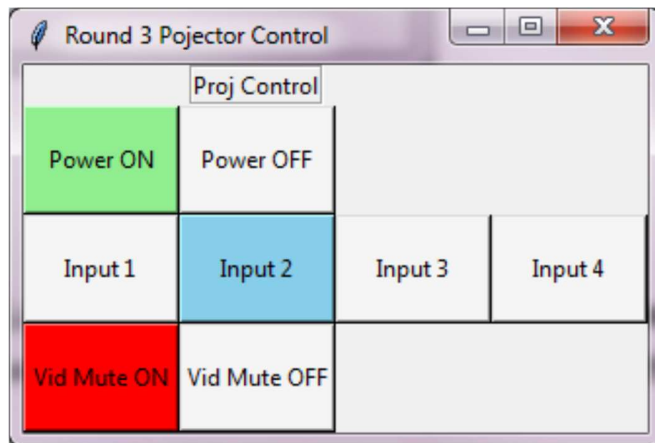
Бонус:

Используйте регулярные выражения для парсинга обратной связи от проектора.

Упражнение 8 – Добавьте второй проектор, используя тот же модуль и класс

GUI:

Используйте GUI от предыдущего упражнения.



Функционал:

Используйте те же кнопки для управления обоими проекторами. Добавьте кнопки для выбора проектора.

Бонусное задание:

Используйте словарь (dictionary) для сохранения настроек проекторов и отслеживания их состояний.

Заключение:

Это последнее упражнение перед тренингом. Я надеюсь, что вы продолжите изучать Python, и в скором времени у вас появится возможность применить свои знания на практике.

Увидимся на тренинге и спасибо вам за усилия и время, потраченное на подготовку к нему.