PYHTON LESSONS

Cmd commodes

1. Cd названия папка=> заходить на папку
2. Cd ../ => виходить из папку
3. Thouch => создаёт файл
4. Mkdir => создаёть папку
5. Print(variable or values) вы водит на терминал

Faster\_commonds

1. Alt+shift+enter => прировняеть код
2. Ctrl + /or? => делоед коментарий
3. Shift + 3 => делоед коментарий
4. “””…….””” => тип коментарирй
5. Fanction.\_\_\_doc\_\_\_ => показивает как работаеть фунция
6. Help => помогоеть
7. KeyWords => ключивый слава
8. \ => даказываеть что это str

Numbers

1. Float => настаяшая число
2. Int =>целая цичло
3. Type => тип переменний
4. \* + str => аичет несколко раз
5. Input() =>спрошивает у ползователя
6. /n => новая линия
7. /t => ставить таб
8. F”{}” => можно дабваить переменую
9. A = a + 5
10. A += 5
11. A -= 5
12. A\*= 5
13. a/=5
14. A\*\*=5
15. Isinstance(1v,2v) => паверяет тип
16. Isnumberic = показиваеть номер ли это
17. <,>,=>,=< ,== ,!=
18. Or => +
19. And => \*
20. Not => отрицаеть
21. Id(variable) => показываеть ид местоположения на диске
22. Is =>проверяеть по id
23. Is not => набароть
24. S == and
25. | == or
26. ~ == not
27. min(…)=> берёть самый болшой
28. max(…)=>берёть самый маленкий
29. Import math
30. Math.Abs(variable) => модул
31. Math.pow(variable , на ксолько) => возведёт по степени
32. Math.sqrt(variable) => показываеть корен
33. Math.pi => число пи
34. Math.floor =>округляеть в меньшию сторону
35. Math.cail =>сокругляеть в болшую сторону
36. Math.trune()=> округляеть в меньшию сторону
37. Round()=> округляеть по матиматике
38. Sep=”…” => место в принте
39. End=”…” =>посли аринта где писять
40. If условия :   
    print()  
    elif условия :1  
    print()  
    else:  
     print()

STRING

1. variable.upper() => все с болшой буквы
2. variable.lower()=>все с моленкой
3. variable.strip()=>удаяеть отступы с начала и конца
4. variable.replace(v1 , v2)=> v1==что менять ,v2 на что менять
5. variable.split(v1)=> позделяеть ,v1==смотря на что разделять
6. len(variable)=> показываеть длину
7. variable[v1:v2:v3]=> лист v1== с кокого брать , v2==до какого ,v3== каждый второй

list methods

1. variable.append() => дабовляеть с конца
2. variable.clear()=> очтшаеть листь
3. variable .copy()=> копируеть листь
4. variable .count(value)=> считаеть сколко точего мы указали
5. variable.extend(value)=>дабввляеть массив с конца
6. variable.index(value)=>показаеть индек значения
7. variable.insert(value, variable\_1)=>v=где дабавить и что дабавить
8. variable .Pop(value)=>берёть с конца и можно мнять с индексом
9. variable.Remove(value)=> удаляеть по имени
10. del variable => удаляеть
11. Reversed(variable)=>переворачивет
12. Sort(variable)=>сортирует сортируеть

Tuple

1) variable .count(value)=> считаеть сколко точего мы указали

2) variable.index(value)=>показаеть индек значения

3)tuple(variable)=>изменять на тапле

3) info about tuple=> тапле нельзя изменять просто так   
но можно в том слечий если вы тупле сделаете листом тогда можно исменить тупле

Git hub

1)git init => скачиваеть гит хаб

2)git status => проверять папку

3)git add name or . => приготовливаеть на отправку

4) git commit -m “name”=> делаеть коментарой или ветку

5) git push => отправляеть или обнавляеть

6) git remote add origin https://github.com/AkobirovS/test.git

7) git branch -M main

8) git push -u origin main

Dictionary

1. Clear = удаляеть
2. Copy = копируеть
3. Fromkeys =дабавляеть всем таплам значения и меняеть на dict
4. Get = берёть
5. Items = показываеть элементы
6. Keys =показываеть ключи
7. Pop = удаляеть с имини
8. Popitem = удаляеть последний
9. Setdefault = ?
10. Update = дабовляеть и изменяеть
11. Values = вфводить значения

**Def**

1. **Def () = создаёть нывй функцию**
2. **Def (arg) = мы можем довать таким оброзом значкения**
3. **Def ( arg , arg\_2 ) = ьы можем довать разный значения и в любом количестве**
4. **Def ( arg=none , arg\_2=10) = в можем довад дефалтный веши на свой значения**
5. **Def ( \*arg , kwargs\*\*) = в первом случий мы берём все элименты**
6. **= а в второй раз мы берём только dictiomary**
7. **Def ( ard: int , arg\_2: list ) = ма модем указывать тып значений или оргументов !!!**

**Class**

### Big information 1. Классы и Объекты

* **Класс** — это шаблон или чертёж для создания объектов. Он описывает свойства (атрибуты) и поведение (методы) объектов.
* **Объект** — это конкретный экземпляр класса, который обладает определёнными характеристиками и может выполнять действия.

### 2. **Инкапсуляция**

**Инкапсуляция** — это сокрытие внутренних деталей реализации класса и предоставление доступа к данным только через публичные методы. В Python инкапсуляция достигается с помощью приватных атрибутов и методов (начинающихся с \_ или \_\_).

### 3. **Наследование**

**Наследование** позволяет создавать новый класс на основе существующего. Новый класс (дочерний) наследует атрибуты и методы родительского класса и может добавлять новые или изменять существующие.

### 4. **Полиморфизм**

**Полиморфизм** означает способность использовать один и тот же интерфейс для разных типов объектов. Это позволяет вызывать одинаковые методы для разных классов.

### 5. **Абстракция**

**Абстракция** — это выделение общих характеристик объектов и создание интерфейсов, скрывающих сложность реализации. В Python абстракция достигается с помощью абстрактных классов и методов из модуля abc.

1. **MINI INFORMATION About <<Class>> => сласс мы можем исползовать как мокеть когда мы пытоемся исползовать это несколко раз  
   и унего есть поттерны == переменный но только внутри класса**
2. **\_\_init\_\_ => принимаеть задданый оргументы**
3. **\_\_str\_\_ => работаеть как инстукция или обесниния ползивателю   
   мы можем призвать его проста написать переменную класса   
   и прениметь толко тип стринг**
4. **\_\_int\_\_ => возврашаеть инить**
5. **\_\_new\_\_ =>**
6. **\_\_repr\_\_ => репер работаеть также как и str но он для девелоперов и   
   мы можем сказать чтото или какие то ошибки**
7. **\_\_del\_\_ => дел мы исползуем для удалений каких либо переменных или данних   
   мы можем исползовать там алгоритом**
8. **\_\_call\_\_ => работаеть когда мы ставим скобки в конце обекста и призивается он берёт в себя аргс и кваркс то есть листь и дикт**
9. **\_\_getitem\_\_ => берёт элементы и можно измкнять и берёть одно значения то есть kay   
   работаеть автомитически**
10. **\_\_setitem\_\_ => мы исползуем для обнавления и оно берёть два значения kay and value**
11. **\_\_delitem\_\_=> дел берёть ключ и удаляеть толко с ним и можно исползовать алгаритм**
12. **\_\_len\_\_ => почему мы не можем исползовать просто лен потомушто в классе не поддержываеть простый отребуды   
    лен проста даёть длинну**
13. **\_\_el\_\_(self, other) => мы исползуем это место == потомучто класс это не поддержываеть   
    self это который сам   
    other это второй значений который приходить из вне**
14. **\_\_lt\_\_(self, other): октругляеть в меншию сторуну и мы это указываем в отнашения селфа**

**Modul**

1. **From имя фела impot что нада импортировать => мы исползуем это чтобы мипортиривать что то из другого фаела**
2. **From имя фаеда impot и можно импортировать много вешей**
3. **Veriable As name => можно менять имя что имортироваль**
4. **Import platform => platform.system () => показываеть платформу**
5. **Import os =>**

**Json**

1. **Import json => импортируеть json**
2. **Json.dumps (veriable) => делаеть json**
3. **indent=”количество” => ставит отступы**
4. **separators = () => чем их пишем**
5. **sort\_keys=(bool) => сортируеть по алфабиту**
6. **Json.loads(veriable) => переоброзуеть в dict**
7. **For use === json pleace holder enter to posts => мы можем исползовать бесплотный Json от суда !!!**

**Pip**

1. **Pip -- version => проверяеть на версию pip па**
2. **Pip install pip => скачываеть пип**
3. **Pip list => провереряеть что скачаль**
4. **Pypi => это саеть с которого можно узнать много информыций про реквесты**
5. **Pip3 install –upgrade pip =>обнавления pip**
6. **Pip install anything => можно скачать разный вуши на свой проекст**
7. **Pip uninstall anything => можно удалять также лехко**

**Requests**

1. **Requests.get(addres) => берёть информацию от тудо кда мы указали   
   paremetr => отправляеть запрос**
2. **Requests.put(addres , {what we are changing}) => что мы хотим изминить в dicte or json**
3. **Requests.putch(addres ,{something we want to chande }=> что мы ходим изминить внутри dicta or json**
4. **Requests.deleta(addres) => удаляеть указынный dict or json**
5. **Requests.status\_code => показываеть статус 200 до 300 успешно**
6. **Requests.headers => инфо на про саеть и как исползоват это**
7. **Requests.text => берёть инфо про html**