Конспект

Словари и Функции

Словари

Словарь это набор ключей и значений, то есть мы можем к каждому элементу может обратиться по ключу.

Представим что у нас есть банк, в банке есть какие-то ячейки.
И на каждой ячейке написан уникальный ключ.
С помощью ключа человек может открыть ячейку.

Допустим приходит семен в банк и говорит - у меня есть такой то ключ, работник банка понимает что у него вот такой ключ, он подходит к ячейке, на которой этот ключ написан, Открывает, ячейку и выдает то что у семена там лежит.

Там нет никаких индексов, то есть мы не можем сказать какая у нас ячейка имеет индекс ноль или какая имеет индекс 3.

Однако мы можем обращаться с ними по ключам.

Как их объявить:

```
1 bank = {'anton': 10, 'dima': 20, 'petya': 4}
```

Через запятую будет прописывать пары - Ключ + Значение ключа. То есть у нас есть ключ Антон и ему соответствует значение 10.

Как же теперь обратиться к значению по ключу?

Это немного похоже на списки, только вместо индекса мы используем наш ключ.

```
1 bank = {'anton': 10, 'dima': 20, 'pe
2
3 print(bank['anton'])
```

Также мы можем обращаться по этому ключу и менять значения.
Потому что он является изменяемым объектом.

```
bank = {'anton': 10, 'dima': 20, 'petya': 4}
bank['anton'] = 159
print(bank)
```

Также важно уточнить

```
bank = {'anton': 10, 'dima': 20, 'petya': 4, 'anton': 15}
print(bank)
```

что каждый ключ в множестве должен быть уникален.

Если мы попробуем написать еще раз ключ антон и значение к нему 15ть, а затем вывести.

То мы заметим что Антон у нас всего лишь один, более того будет использован тот Антон который в самом конце.

Давайте попробуем решить практическую задачу.

Пускай у нас есть банк в который приходит икс запросов каждый запрос бывает двух типов:
Либо запрос на открытие новой ячейки с каким то ключом.
Либо мы хотим положить деньги в ячейку с каким то ключом.
И в конце нам нужно вывести что вообще получилось.

То есть какие есть ячейки сколько в них лежит денег.

Давайте посмотрим как объявить пустой словарь:

```
bank = dict()
```

То есть наш банк это дикт - как раз таки словарь. Если мы в круглых скобках ничего не передаем он будет как раз таки пустой.

Отлично, далее мы знаем что нам придет какое-то количество запросов которые придут к нашему банку:

```
bank = dict()
n = int(input())
```

Теперь с помощью цикла поочередно будем обрабатывать все запросы.

```
for i in range(n):
```

Для начала пускай запрос кодируется типом, если это:
create - значит мы хотим открыть ячейку,
если это add значит мы хотим добавить какое то количество денег на нашу ячейку.

Изначально сделаем ввод типа нашего запроса:

```
bank = dict()
n = int(input())

for i in range(n):
    req = input()
```

Теперь у нас есть два варианта:
если **req** = create значит что нам ведется еще ключ ячейки которую мы хотим открыть, пускай это будет K

```
bank = dict()
n = int(input())

for i in range(n):
    req = input(" Введите запрос (create или add): ")

    if req == "create":
        k = input("Введите ключ с которым хотите открыть
```

Чтобы открыть ячейку мы можем просто взять и сказать что

```
if req == "create":
    k = input("Введите ключ с которым хотит
    bank[k] = 0
```

Банк от К = 0, потому что при открытии нашей ячейки денег там нет.

Иначе если запрос add, то мы должны будет добавить какое-то количество денег.
Но чтобы понимать в какую ячейку и сколько денег добавить, мы должны это значение ввести.
Вводим ключ по которому будет обращаться к ячейке.
И вводим amount - количество денег которое хотим добавить:

```
elif req == "add":
    k = input("Введите ключ ячейки для пополнения: ")
    amount = int(input("Сумма для пополнения: "))
```

Теперь добавим проверку на то был ли такой ключ в нашем словаре.
То есть задача проверить присутствует ли какой то ключ в ключах нашего словаря.

```
    if k in bank.keys():
        bank[k] += amount
    else:
        print(" Ключ не найден ")
```

Функция keys возвращает набор ключей которые находятся в словаре, то есть мы проверяем присутствует ли К в наборе ключей keys банка.

Если это так то мы добавляем туда amount денег
Иначе выводим “Ключ не найден”

И выводим что у нас в итоге получилось.

```
bank = dict()
n = int(input("Сколько запросов пришло банку?: "))

for i in range(n):
    req = input("Введите запрос (create или add): ")

    if req == "create":
        k = input("Введите ключ с которым хотите открыть новую ячейку: ")
        bank[k] = 0
        print("Успешно!")

    elif req == "add":
        k = input("Введите ключ ячейки для пополнения: ")
        amount = int(input("Сумма для пополнения: "))

        if k in bank.keys():
            bank[k] += amount
            print("Успешно!")
        else:
            print("Ключ не найден")

    else:
        print("Извините, запрос неверный")
print(f'Итог на конец дня: {bank}')
```

Функции

Теперь разберемся с темой функций и зачем они вообще нам нужны.

Функции по сути это подпрограмма внутри нашей программы.
Функция помогает сделать программу более читаемой, разбить её на понятные блоки.
А так же у них огромный спектр возможностей.

Разберем на конкретном примере:

Нам дается число, мы должны определить четное оно или нет.

То есть на вход мы будем получать число, а возвращать строку.

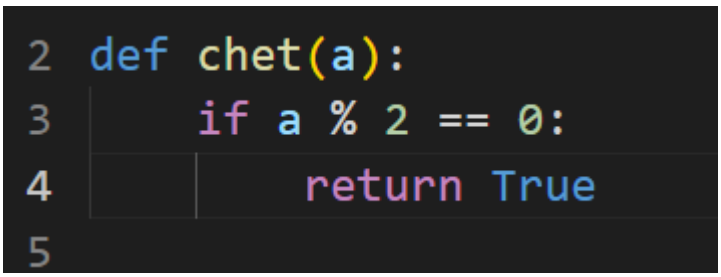
Функцию можно определить при помощи ключевого слова def, за которым должно следовать название функции и список её формальных параметров в круглых скобках.

В нашем случае просто передается число и мы должны определить является оно четным или нет.

Далее ставим двоеточие и все что происходит в функции идет с отступом.

Вот нам пришло число a на вход.

Добавим проверку на то четное оно или не четное:



?,

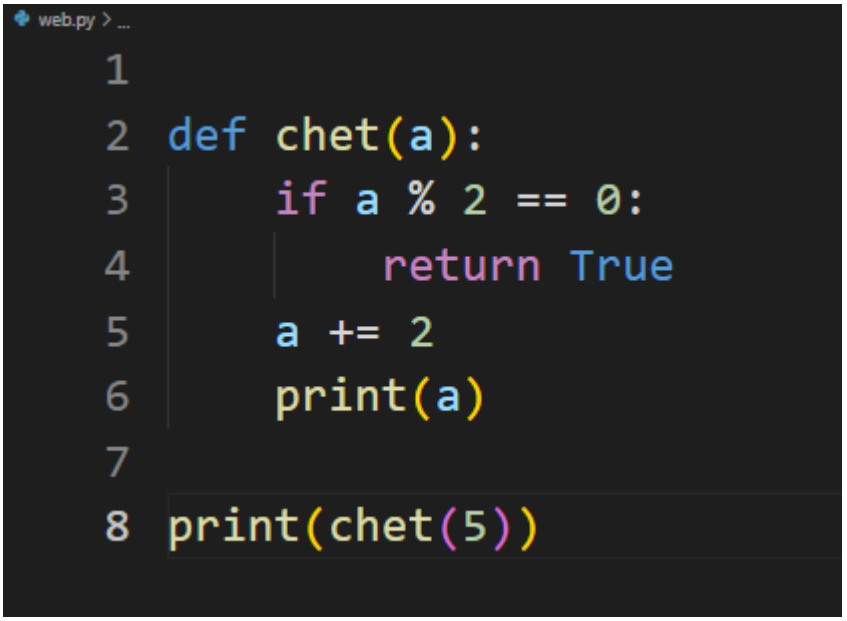
Что здесь произошло? Мы вернули значение TRUE из нашей функции, то есть результатом функции будет какое-то логическое значение. ФУНКЦИЯ не всегда принимает логическое значение, сейчас мы просто рассматриваем именно такую функцию.

```
def chet(a):
    if a % 2 == 0:
        return True
    a += 2
    print(a)
```

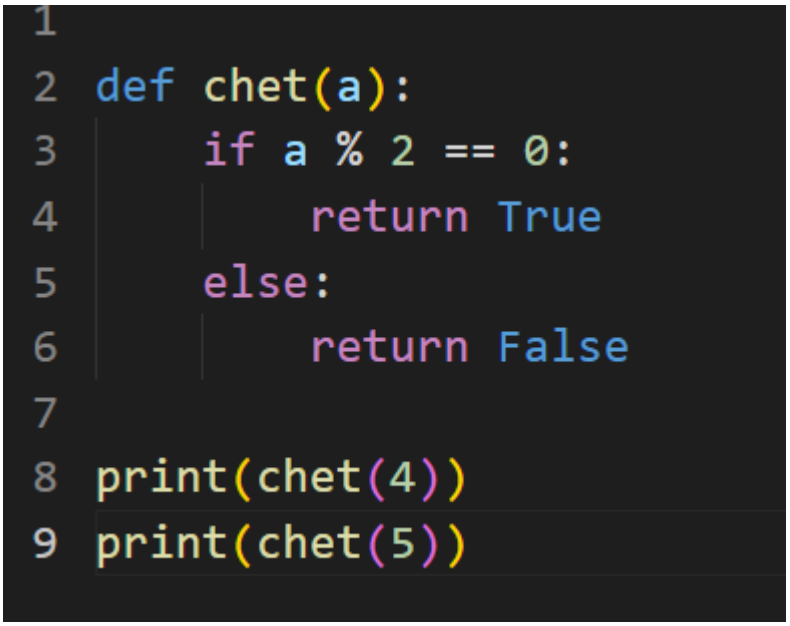
Важно уточнить что после того как мы возвращаем какое то значение из функции - функция завершается.

Как запустить такую функцию?

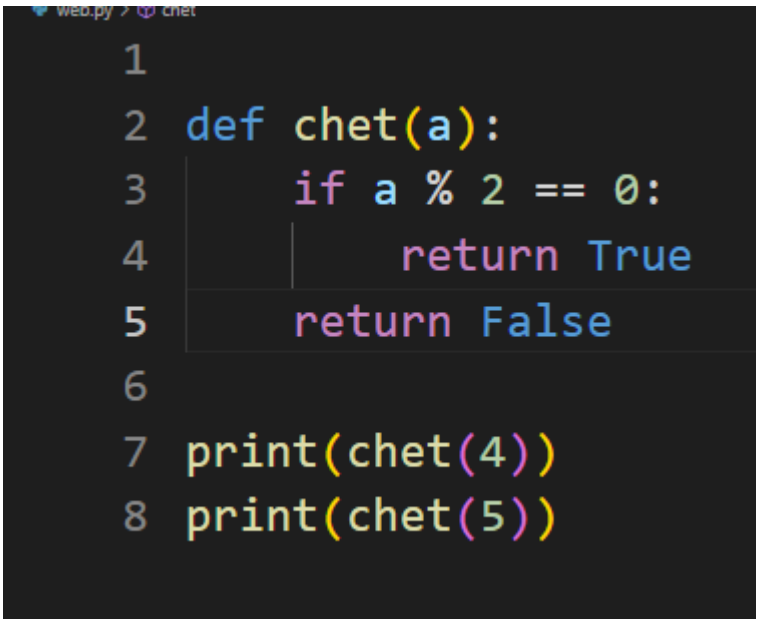
Важно уточнить что функция выполняется только после того как её вызвали.



Если ввести значение не четное то мы получим его +2 и NONE
Это значит что функция ничего не вернула, а мы пытаемся вернуть её результат, то есть ключевое слово return означает что мы что-то возвращаем из функции, то есть это некоторый результат её работы, а по скольку мы возвращаем ничего то и в результате NONE.



Также мы можем работать тут и без else:



Давайте посмотрим функцию которая не будет возвращать ничего, а просто будет выполнять какие-то действия.