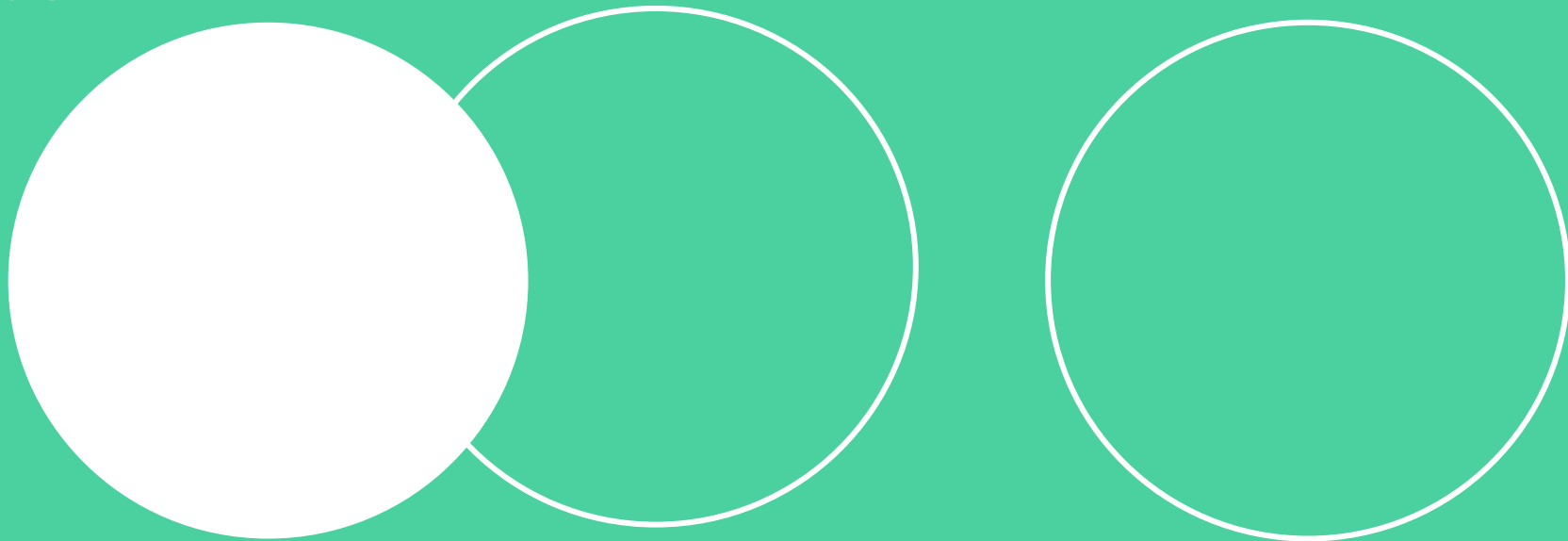


Функции. Разработка приложения ToDo



Занятие 3



Дмитрий Демидов
Разработчик Яндекс

План:

1. Доработка программы
2. Цикл for
3. Функции
4. Использование сторонних библиотек

Доработка программы

1



План:

1. Функциональность программы
2. Меняем хранилище для данных

Функциональность программы

Что уже умеет наша программа?

- Выводить справку по доступным командам.
- Добавлять задачи в список.
- Выводить на экран все добавленные задачи.
- Работать бесконечно.
- До тех пор, пока пользователь не введет специальную команду.

Что еще нужно сделать?

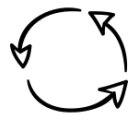
- Нет привязки задач к датам.
- Обещанная секретная фишка.

Итоги:

1. Наша программа почти готова, но
2. Задачи выводятся не красиво
3. Все еще нет секретной фичи

Цикл for

2



План:

1. Использование цикла `for` для перебора элементов контейнера
2. Использование цикла `for` в программе для вывода на экран списка задач

Цикл for

```
for element in <список>:  
    <блок кода>  
    break
```

- для каждого элемента списка <список>
- присвой element элемент списка (имя переменной может быть любым),
- выполни <блок кода>,
- если встретишь break, прервись (необязательное условие).

Цикл for (пример)

```
for i in [4, 5, 6]:  
    if i % 2 == 0: # остаток от деления  
        print(i)
```

Номер итерации	Значение i	Вывод на экран
1	4	4
2	5	
3	6	6

Итоги:

1. Цикл `for` служит для перебора элементов контейнера (списка)

Функции

3



План:

1. Добавление базового варианта команды random
2. Создание собственных функций
3. Вынос добавления задачи в отдельную функцию

Функции (синтаксис)

```
def my_func(i):
```

```
    if i > 2:
```

```
        i = 2
```

```
    return i
```

Подробнее про функции

```
def my_func(i): # название функции
    if i > 2:    # тело функции
        i = 2
    return i     # возвращаемое значение
```

```
x = my_func(3)
```

В переменную `x` будет помещено возвращаемое значение функции `my_func`, полученное в результате выполнения тела функции, с условием, что `i = 3`

Что будет в `x`?

Итоги:

1. Функции - это именованный блок кода, который вызывается в нужных местах программы по имени.
2. Создание собственных функций
3. Вынос добавления задачи в отдельную функцию

Использование сторонних библиотек

4



План:

1. Ключевое слово `import`
2. Виды сторонних функций (встроенные, стандартная библиотека, `third-party`)
3. Доработка программы с использованием библиотеки `random`

Виды сторонних функций

Встроенные	Доступны без дополнительного кода	Уже использовали: print, input
Стандартная библиотека	Необходимо импортировать (import)	Будем использовать сегодня
Другие библиотеки (third-party)	Необходимо предварительно установить	Познакомимся на следующем занятии

Итоги:

1. Ключевое слово `import` служит для доступа к готовым функциям из библиотек, написанных другими разработчиками
2. Есть 3 вида функций, которые вы можете использовать в своем приложении: встроенные, стандартные и `third-party`

Заключение

5



Итог:

1. Цикл for
2. Функции - создание собственных и использование сторонних

**До встречи на следующем
занятии**



Дмитрий Демидов