

# Telegram-bot 'DailyDay Bot'

Проект Здановича Никиты и Егорова Вячеслава

## Что за telegram-бот?

1. Это бот напоминка с помощью которой вы не забудете о каком-то событии
2. Так же вы можете пожертвовать на развитие данного бота



# В нашем программном коде мы использовали функции:

1. def start(начальная функция помогает пользователю запустить нашего бота)
2. def help(функция которая помогает пользователю лучше узнать все команды нашего бота)
3. def donate(функция дающая ссылку на сайт где можно пожертвовать создателю сайта)
4. def info(функция которая помогает пользователю ознакомиться с возможностями нашего бота )

```
@bot.message_handler(commands=['start'])
def start(message):
    timenow = int(''.join((' '.join(time.asctime().split()[3:-1])).split(':')[0]))
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("/help")
    markup.add(btn1)
    a = ''
    # message.chat.username
    if 23 < timenow <= 6:
        a = 'Доброй ночи'
    elif 6 < timenow <= 12:
        a = 'Доброе утро'
    elif 12 < timenow <= 18:
        a = 'Доброго дня'
    elif 18 < timenow <= 23:
        a = 'Доброго вечера'
    bot.send_message(message.chat.id, f"{a}, {message.from_user.first_name}! Я бот,"
                                     f" который создает напоминания для тебя!"
                                     f"\n"
                                     f"\n"
                                     f"Введите /help для того, чтобы увидеть возможности бота.")
```

```
@bot.message_handler(commands=['help'])
def help(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("/donate")
    btn2 = types.KeyboardButton("/info")
    btn3 = types.KeyboardButton("/today")
    btn4 = types.KeyboardButton("/note")
    markup.add(btn1, btn2, btn3, btn4)
    bot.send_message(message.chat.id,
                     text="Функции бота: \n"
                          "/donate - пожертвования разработчикам \n"
                          "/info - информация о боте, для чего он нужен \n"
                          "/today - выводит сегодняшнее число и время \n"
                          "/note - создает напоминание".format(
                          message.from_user), reply_markup=markup)
```

# В нашем программном коде мы использовали функции:

5. def reminder(сама напоминалка, вы вводите что конкретно вам нужно напомнить)
6. def clock(выводит часы, минуты, секунды)
7. def msg(функция где бот обрабатывает вашу заметку и напомнит об этой заметке в указанное время)

```
@bot.message_handler(commands=['note'])
@bot.message_handler(content_types=["text"])
def reminder(message):
    if message.text == "/note":
        bot.send_message(message.chat.id,
                           text="Напишите, через сколько времени вы хотите сделать напоминание. \n"
                                   "Например: чч:мм:сс\n"
                                   "\n"
                                   "Либо введите через какое количество секунд вы хотите получить сообщение".format(
                                       message.from_user))
        bot.register_next_step_handler(message, clock)

    if message.text == "/note_en":
        bot.send_message(message.chat.id,
                           text="Enter how soon you want to be reminded. \n"
                                   "For example: hours:minutes:seconds\n"
                                   "\n"
                                   "Or enter after what time you want to receive a message".format(
                                       message.from_user))
        bot.register_next_step_handler(message, clock_en)
```

```
def msg(message):
    note = message.text
    bot.send_message(message.chat.id, f'Спасибо! Мы получили вашу заметку! \n'
                                       f'Напоминание будет через {hour}ч {minutes}мин {sec}сек')

    hour_new = hour * 60 * 60
    minutes_new = minutes * 60
    sec_new = sec
    total_time = hour_new + minutes_new + sec_new
    time.sleep(total_time)
    bot.send_message(message.from_user.id, f'Здравствуйте, {message.from_user.first_name}! Напоминание для вас: \n'
                                       f'\n'
                                       f'{note}')

bot.polling(none_stop=True)
```

# Дополнительная функция

1. В наш бот мы добавили английский язык.  
Бот подскажет на английском языке какие клавиши если вдруг вы что-то не поняли нажимать

```
def clock_en(message):

    global hour1
    global minutes1
    global sec1

    hour1 = 0
    minutes1 = 0
    sec1 = 0
    alert = message.text.split(':')
    if len(alert) == 3:
        if alert[0].isdigit() == True and alert[1].isdigit() == True and alert[2].isdigit() == True:
            if 1 <= len(alert[0]) <= 2 and 1 <= len(alert[1]) <= 2 and 1 <= len(alert[2]) <= 2:
                if 0 <= int(alert[0]) <= 99 and 0 <= int(alert[1]) <= 59 and 0 <= int(alert[2]) <= 59:
                    hour1 += int(alert[0])
                    minutes1 += int(alert[1])
                    sec1 += int(alert[2])
                    bot.send_message(message.chat.id, f'Enter your note.')
                    bot.register_next_step_handler(message, msg_en)
                else:
                    bot.send_message(message.chat.id, f'Error in entering numbers!!! \n'
                                                    f'Note that the number of hours can NOT exceed 99,'
                                                    f'the number of minutes and seconds cannot exceed 59. \n'
                                                    f'\n'
                                                    f'Enter the time again')
                    bot.register_next_step_handler(message, clock_en)
            else:
                bot.send_message(message.chat.id, f'Error in entering numbers!!! \n'
                                                    f'Note that numbers can only be written with ONE digit'
                                                    f'or TWO!!! \n'
                                                    f'\n'
                                                    f'Enter the time again')
                bot.register_next_step_handler(message, clock_en)
        else:
            bot.send_message(message.chat.id, f'Error in entering!!! \n'
                                                f'Time CANNOT be spelled out!!!\n'
                                                f'\n'
                                                f'Enter the time again')
            bot.register_next_step_handler(message, clock_en)

    elif len(alert) == 1:
        if alert[0].isdigit() == True:
            sec1 += int(alert[0])
            bot.send_message(message.chat.id, f'Enter your note.')
            bot.register_next_step_handler(message, msg_en)
        else:
            bot.send_message(message.chat.id, f'Input error!!! \n'
                                                f'Time CANNOT be spelled out!!!\n'
                                                f'\n'
                                                f'Enter the time again')
            bot.register_next_step_handler(message, clock_en)
```

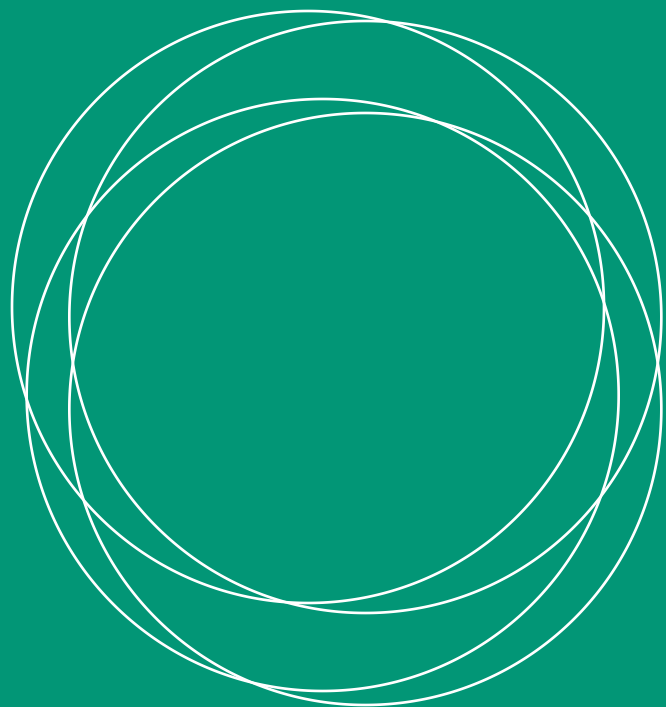
## Дизайн бота:

Изначально было несколько вариаций  
логотипа:



DAILY  
DAY

DAILY  
DAY  
AMONG US



**пасиба за внимание**