



алгоритмика

# PYTHON

Занятие 19

# Сегодня на уроке

---

- Знакомство с Pygame.
- Бесконечный цикл.
- Спрайт.
- Создание класса.



# Вопросы

---

1. Что такое цикл?
2. Что такое условие?
3. Что такое условный цикл?
4. Из чего состоит цвет RGB палитры?
5. Откуда начинается отсчет координат в окне приложения?

# Цикл в Python

---

**Цикл `while`** – повторяет команды тела цикла, пока верно условие.

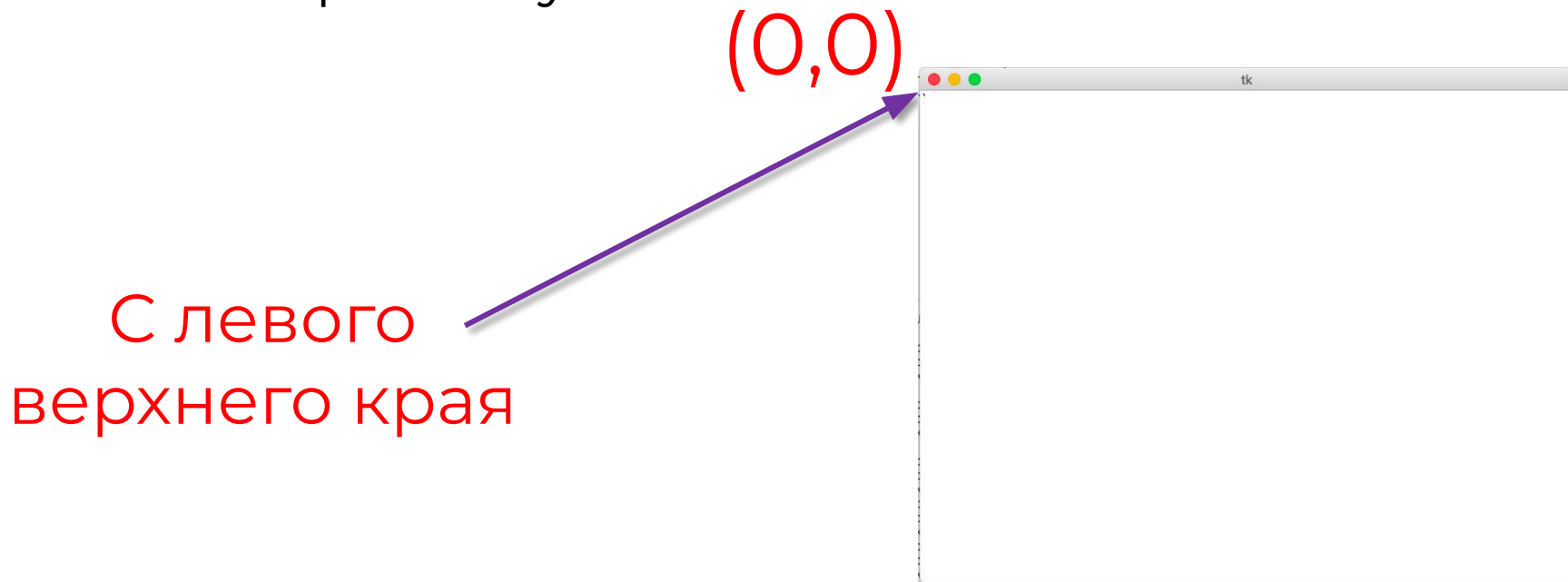
`while` условие: } заголовок

← отступ → } тело цикла

# Начало координат

---

Начало координат в приложении начинается с левого верхнего угла.



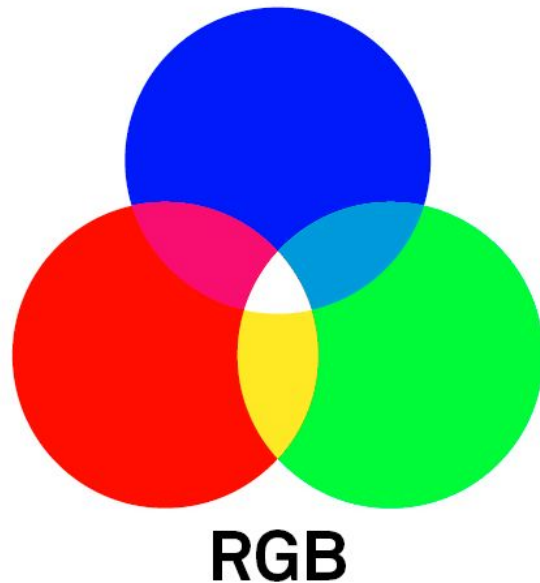
# RGB палитра

---

RGB палитра (**R**ed, **G**reen, **B**lue) – палитра, в которой цвета создаются с помощью 3-х составляющих – красного, зелёного и синего.

Вклад каждого цвета выражается числовым значением от 0 до 255.

Палитра содержит **16 777 216** комбинаций цветов.



# Модуль pygame

---

Pygame – модуль для языка Python, созданный для разработки 2D игр.



# Модуль pygame. Установка

---

Так как модуль не входит в основной пакет Python, его необходимо устанавливать дополнительно. [Инструкция по установке.](#)

1. Запустите Visual Studio Code.
2. Нажмите Ctrl+Shift+P (Cmd+Shift+P) для запуска строки поиска команд.
3. Введите “Python create”, в списке доступных команд останется только команда “Python: create terminal”. Нажмите на неё.
4. Подождите, пока откроется новое окно терминала и в нем выполнится команда “activate” (эта команда запускается автоматически). После выполнения команды “activate” должна появиться строка для ввода команды.
5. Для обновления “pip” до последней версии введите в терминале “pip install --upgrade pip”. Дождитесь окончания обновления пакета.
6. Введите “pip install имя\_пакета” (например, **“pip install pygame”**). Дождитесь окончания установки пакета.





# Модуль pygame

---

Подключение модуля:

```
import pygame
```

При таком импорте модуля – объектом в программе будет **pygame**.

```
pygame.init()
```

Импорт всего инструментария Pygame.



# Задание

---

Создайте новый проект с именем `first_game`. Добавьте в программу подключение модуля и его инструментария. Запустите проект и проверьте работу модуля.

```
import pygame  
pygame.init()
```



# Вопрос

---

Что произошло при запуске проекта? Чтобы мы хотели увидеть после запуска?



# Вопрос-ответ

---

Что произошло при запуске проекта? Чтобы мы хотели увидеть после запуска?

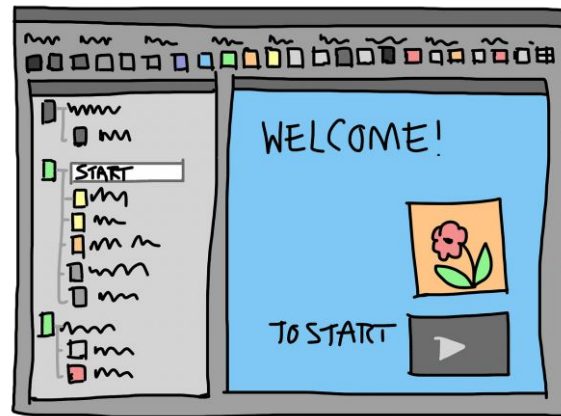
Появился значок запуска  
приложения и пропало  
окно игры.



# Вопрос

---

Как создавалось окно приложения в модуле tkinter?



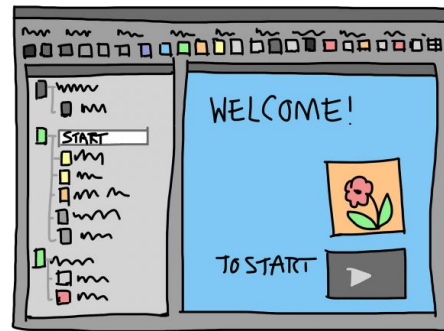
# Вопрос-ответ

---

Как создавалось окно приложения в модуле tkinter?

`root=Tk()` – создание объекта окна.

`root.geometry('640x480')` – задание размера окна.



# Создание игрового окна

---

Метод `display` позволяет создать игровое окно заданного размера.

```
display.set_mode((ширина, высота))
```



# Задание

---

Добавьте в программу создание объекта окна **sc** (screen). Для задания ширины и высоты добавьте переменные: **W** (ширина) и **H** (высота).  
Размер окна **400\*400**.

`pygame.display.set_mode((ширина, высота))`





# Задание. Решение

---

Добавьте в программу создание объекта окна **sc** (screen). Для задания ширины и высоты добавьте переменные: **W** (ширина) и **H** (высота).

Размер окна **400\*400**.

```
import pygame
```

```
pygame.init()
```

```
W = 400
```

```
H = 400
```

```
sc = pygame.display.set_mode((W, H))
```

```
import pygame
pygame.init()
W = 400
H = 400
sc = pygame.display.set_mode((W, H))
```



# Вопрос

---

С какой проблемой мы снова столкнулись после создания окна?

```
import pygame
pygame.init()
W = 400
H = 400
sc = pygame.display.set_mode((W, H))
```



# Вопрос-ответ

---

С какой проблемой мы снова столкнулись после создания окна?

Окно создалось и закрылось.

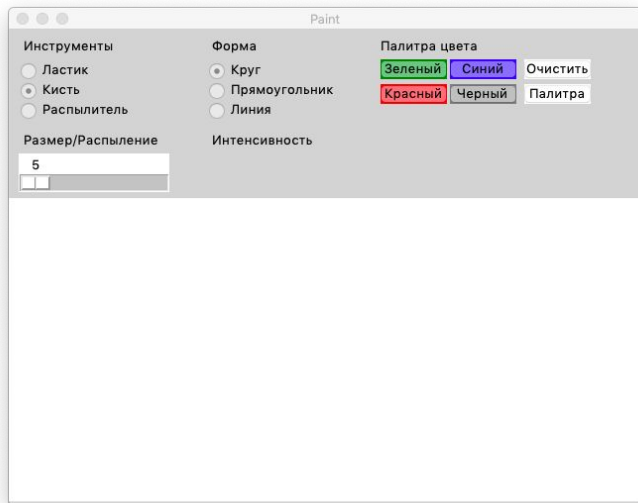
```
import pygame
pygame.init()
W = 400
H = 400
sc = pygame.display.set_mode((W, H))
```



# Вопрос

---

Чем отличается окно приложения от игрового окна?

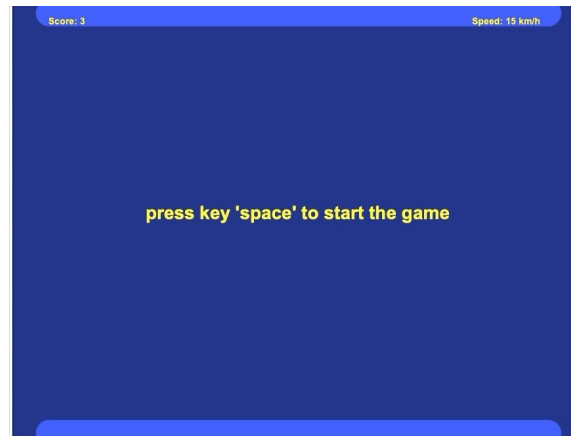
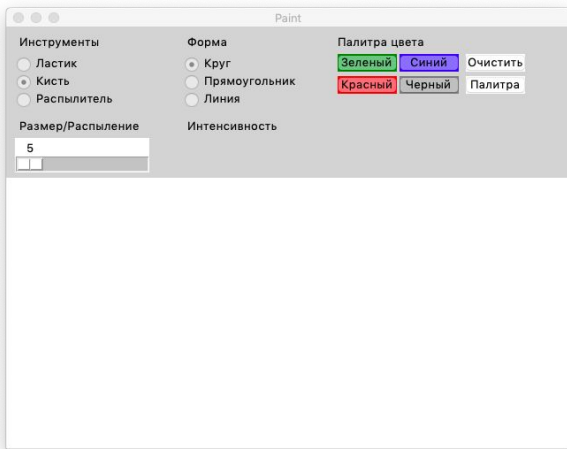


# Вопрос-ответ

---

Чем отличается окно приложения от игрового окна?

В игровом окне постоянно происходит действие, изменение содержимого окна.



# Вопрос

---

С помощью какой конструкции мы сможем постоянно заново создавать игровое окно много раз?



# Вопрос-ответ

---

С помощью какой конструкции мы сможем постоянно заново создавать игровое окно много раз?

С помощью Цикла.



# Вопрос

---

Известно ли, как долго будет длиться игра?





# Вопрос-ответ

---

Известно ли, как долго будет длиться игра?

Нет.

Пока кто-нибудь не выиграет  
или не проиграет, или не  
закроет игру.



# Вопрос

---

Как вы думаете, что такое бесконечный цикл?



# Вопрос-ответ

---

Как вы думаете, что такое бесконечный цикл?

Это цикл, который никогда  
не закончится.



# Вопрос

---

Что влияет на продолжение работы цикла или на его остановку?



# Вопрос-ответ

---

Что влияет на продолжение работы цикла или на его остановку?

Условие. Цикл заканчивает свою работу, когда условие становится ложным.



# Вопрос

---

Как сделать из условного цикла бесконечный?



# Вопрос-ответ

---

Как сделать из условного цикла бесконечный?

Условие всегда должно быть  
ИСТИННЫМ.



# Вопрос

---

Что будет всегда истинным?





# Вопрос-ответ

---

Что будет всегда истинным?

**Истина**  
**True**



# Бесконечный цикл

---

**Бесконечный цикл** – это цикл, условие выхода из которого никогда не выполняется.

game=True – переменная условия цикла.

```
while game:  
    тело_цикла
```

# Вопрос

---

Как отследить событие в приложении? В каком месте оно было совершено?



# Вопрос-ответ

---

Как отследить событие в приложении? В каком месте оно было совершено?

С помощью объекта  
event.



# Вопрос

---

С помощью какого метода возможно получить значение объекта?



# Вопрос-ответ

---

С помощью какого метода возможно получить значение объекта?

Метод `get()`.



# Обработка события

---

Метод `get()` вернет значение события, которое произошло в игровом окне.

```
pygame.event.get()
```



# Обработка события

---

События происходят в окне постоянно, поэтому их необходимо проверять в цикле.

```
while game:  
    for i in pygame.event.get():
```

При прохождении каждого цикла переменная **i** будет указывать на определённое значение события.

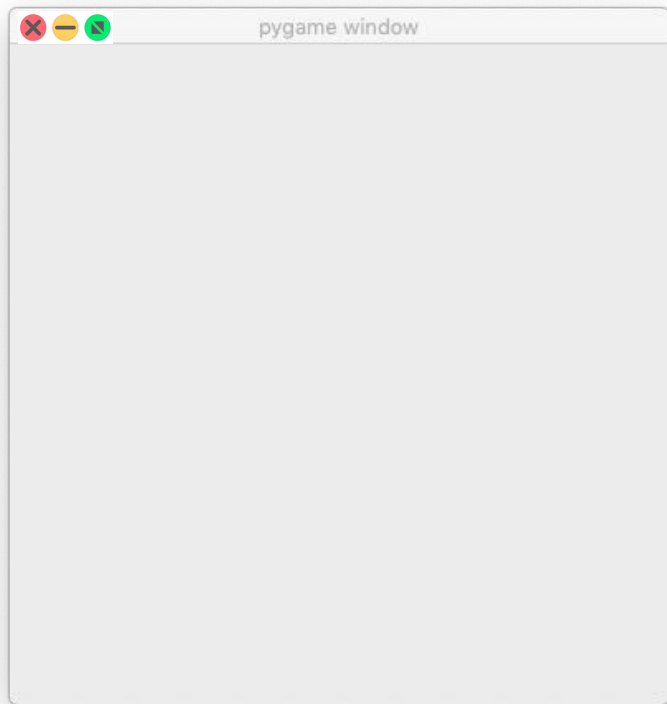




# Вопрос

---

Что должен сделать пользователь, чтобы закрыть игровое окно?

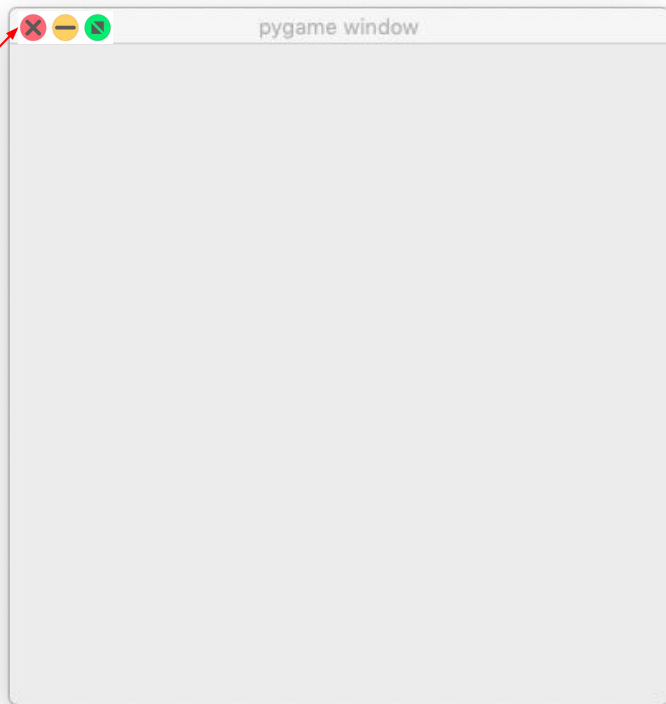


# Вопрос-ответ

---

Что должен сделать пользователь, чтобы закрыть игровое окно?

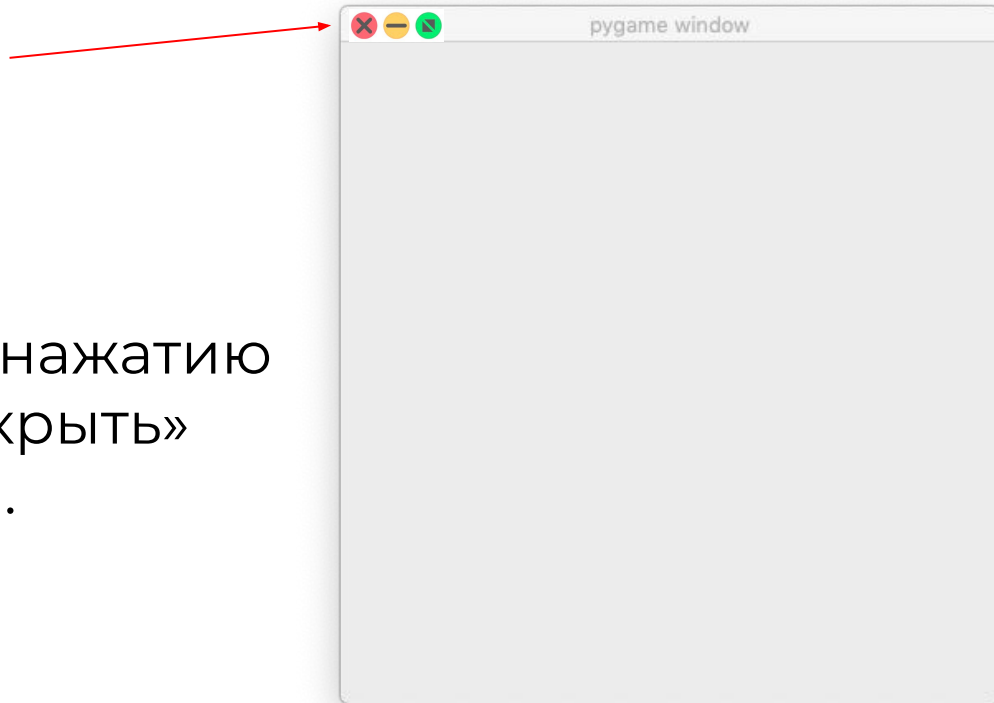
Нажать  
кнопку  
закрыть (x)



# Событие закрытие окна

---

`pygame.QUIT`



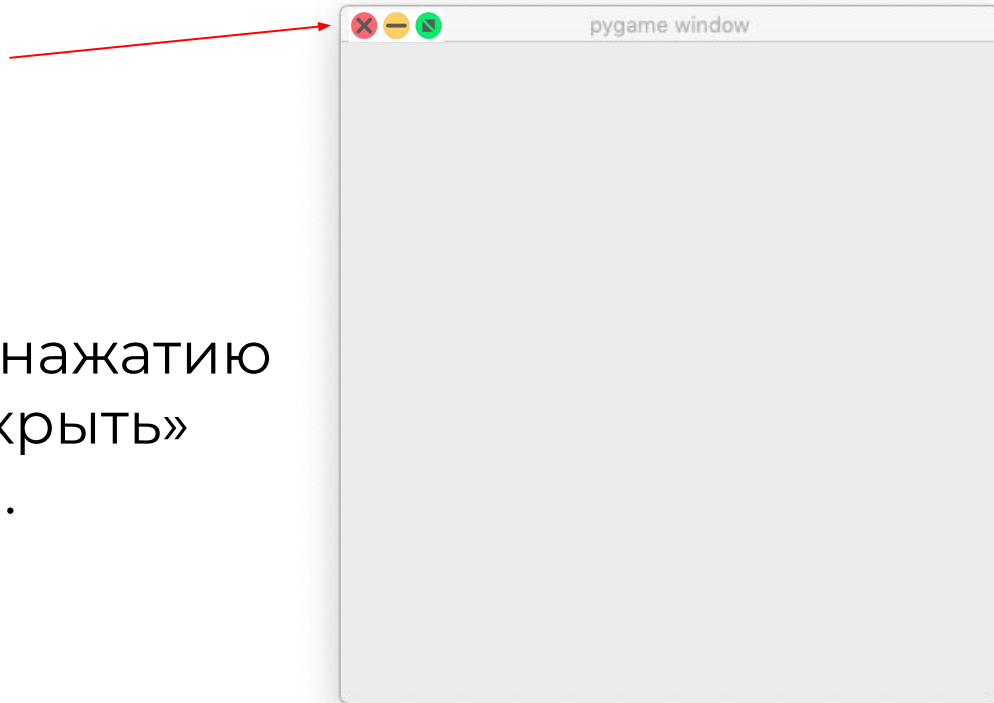
Событие **QUIT**  
соответствует нажатию  
на кнопку «Заккрыть»  
игрового окна.



# Событие закрытие окна

---

`pygame.QUIT`



Событие **QUIT**  
соответствует нажатию  
на кнопку «Заккрыть»  
игрового окна.



# Тип события

---

У каждого события есть свой тип, который возможно получить с помощью метода **type**.

**i.type** - возвращает тип события объекта i.



# Вопрос

---

Какую конструкцию необходимо использовать, чтобы совершить действие только после выполнения определённого условия?



# Вопрос-ответ

---

Какую конструкцию необходимо использовать, чтобы совершить действие только после выполнения определённого условия?

Условные оператор

if условие:  
    команды



# Задание

---

Добавьте в программу получение всех действий, которые совершает пользователь и обработку этих событий в программе.

```
game=True
while game:
    for i in pygame.event.get():
        if i.type == pygame.QUIT:
            print("Exit")
```





# Вопрос

---

Программа выводит сообщение «Exit», но окно не закрывается. Как заставить цикл прекратить работать?



# Вопрос-ответ

---

Программа выводит сообщение «Exit», но окно не закрывается. Как заставить цикл прекратить работать?

Сделать его условие  
ЛОЖНЫМ.



# Вопрос

---

Что проверяется в условии цикла?



# Вопрос-ответ

---

Что проверяется в условии цикла?

Значение переменной **game**. Пока она истинная, цикл выполняется.



# Вопрос

---

Если пользователь нажал на кнопку закрытия окна, какой должна стать переменная?



# Вопрос-ответ

---

Если пользователь нажал на кнопку закрытия окна, какой должна стать переменная?

Ложной.  
`game=False`



# Задание

---

Добавьте в программу изменение значения переменной **game**. После того, как пользователь нажал на кнопку «Заккрыть», переменная должна указывать на **False**.



# Задание. Решение

---

Добавьте в программу изменение значения переменной **game**. После того, как пользователь нажал на кнопку «Заккрыть», переменная должна указывать на **False**.

```
game=True
while game:
    for i in pygame.event.get():
        if i.type == pygame.QUIT:
            print("Exit")
            game=False
```





# Вопрос

---

Назовите основные составные части любой игры.



# Вопрос-ответ

---

Назовите основные составные части любой игры.

1. Персонаж/Герой.
2. Мир.
3. Сюжет.
4. Правила.



# Спрайт

---

**Спрайт** – это интерактивная объект-картинка, который выступает в роли героя или персонажа игры. Поведение спрайта зависит от действия пользователя или заранее определено в программе.



# Наша первая игра

---

???



# ГОНКИ



# Создание Спрайта для игры

---

**Pygame** не знает обо всех персонажах игр, но позволяет их создать самому разработчику!

## `pygame.sprite.Sprite`

Класс **Sprite** хранит все методы, которые необходимы для работы с объектом-спрайтом: управление, изменение, отслеживание и т.п.



# Создание класса-спрайта

---

Все спрайты игры будут объектами класса **Car**, который, в свою очередь, получил методы для работы с ними от класса **Sprite**. **Car** – дочерний класс, **Sprite** – родительский.

```
class Car(pygame.sprite.Sprite):
```



# Вопрос

---

Что такое функция?





# Вопрос-ответ

---

Что такое функция?

Это правило, которое  
описывает поведение объекта  
программы.



# Правила Спрайта

---

Функция-конструктор объектов класса:

```
def __init__():
```

Содержит в себе описание создания каждого объекта-спрайта.



# Конструктор Спрайта

---

```
def __init__(self, x, filename):
```

**self** – контейнер для объекта-спрайта.



**x** – положение спрайта на экране.

**filename** – имя файла изображения спрайта.



# Конструктор Спрайта

---

```
class Car(pygame.sprite.Sprite):
```

```
    def __init__(self, x, filename):
```

```
        pygame.sprite.Sprite.__init__(self)
```

1. Создать объекта класса Sprite.



# Конструктор Спрайта

---

```
class Car(pygame.sprite.Sprite):  
    def __init__(self, x, filename):  
        pygame.sprite.Sprite.__init__(self)  
        self.image = pygame.image.load(filename). convert_alpha()
```

2. Присоединить к спрайту изображение из указанного файла (filename).

Файл должен находиться в том же месте, что и программа игры!

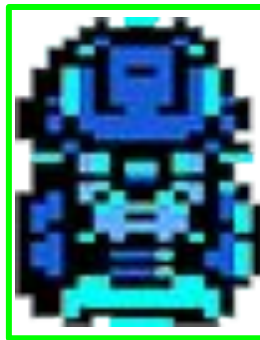
**convert\_alpha** – изменить формат пикселей изображения.



# Конструктор Спрайта

```
class Car(pygame.sprite.Sprite):  
    def __init__(self, x, filename):  
        pygame.sprite.Sprite.__init__(self)  
        self.image = pygame.image.load(filename).convert_alpha()  
        self.rect = self.image.get_rect(center=(x, 0))
```

3. Создать вокруг  
загруженного изображения  
прямоугольник, для  
дальнейшего управления и  
изменения спрайтом.



# Задание

---

Добавьте в программу создание класса Car и конструктор спрайта. В качестве спрайта будет использоваться картинка Car.png. Проверьте, что она находится в той же папке, что и программа.

```
class Car(pygame.sprite.Sprite):  
    def __init__(self, x, filename):  
        pygame.sprite.Sprite.__init__(self)  
        self.image = pygame.image.load(filename).convert_alpha()  
        self.rect = self.image.get_rect(center=(x, 0))
```



# Задание. Решение

---

Добавьте в программу создание класса Car и конструктор спрайта. В качестве спрайта будет использоваться картинка Car.png. Проверьте, что она находится в той же папке, что и программа.

```
class Car(pygame.sprite.Sprite):  
    def __init__(self, x, filename):  
        pygame.sprite.Sprite.__init__(self)  
        self.image = pygame.image.load(filename).convert_alpha()  
        self.rect = self.image.get_rect(center=(x, 0))
```

**Car.png**





# Вопрос

---

Почему на игровом поле мы не увидели машину?



# Вопрос-ответ

---

Почему на игровом поле мы не увидели машину?

Мы только создали правило, как  
создавать спрайт.

**НО** сам спрайт в программе ещё **не  
создали** и **не разместили** его в  
игровом окне.



# Задание

---

Добавьте в программу создание объекта car1 класса Car. X=20, файл – 'Car.png'.



# Задание. Решение

---

Добавьте в программу создание объекта car1 класса Car. X=20, файл – 'Car1.png'.

```
car1 = Car(20, 'Car1.png')
```

```
def __init__(self, x, filename):
```



# Вопрос

---

Почему на игровом поле мы не увидели машину?



# Вопрос-ответ

---

Почему на игровом поле мы не увидели машину?

Мы создали объект и правило,  
**НО** сам спрайт в программе ещё  
**не разместили** в игровом окне



# Вопрос

---

Что в программе отвечает за обработку изменения игрового поля?



# Вопрос

---

Что в программе отвечает за обработку изменения игрового поля?

Бесконечный  
цикл





# Обработка игрового окна

---

sc.**fill**(цвет\_фона)

sc.**blit**(car1.image, car1.rect)

pygame.**display.update**()

**fill(цвет)** – задает цвет фона игрового поля.

**blit(файл\_1, файл\_2)** – рисует файл\_1 сверху файл\_2.

**display.update()** – обновляет игровой экран.

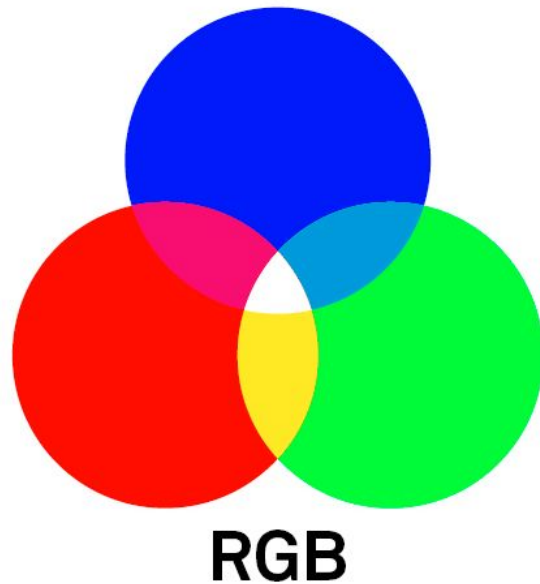


# Вопрос

---

Как задать белый цвет с помощью палитры RGB?

Вклад каждого цвета выражается числовым значением от 0 до 255.



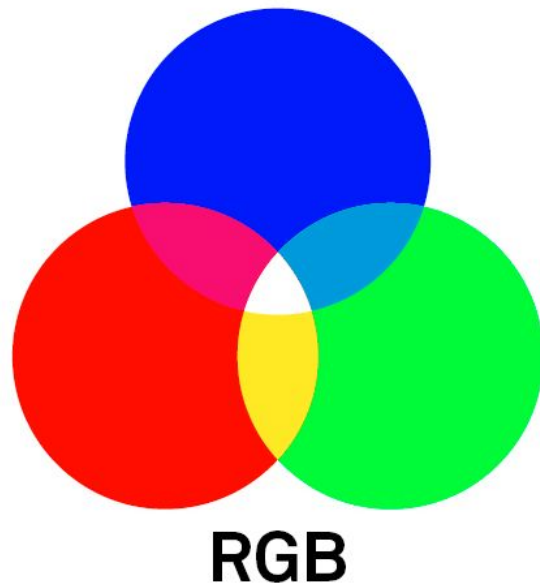
# Вопрос-ответ

---

Как задать белый цвет с помощью палитры RGB?

Вклад каждого цвета выражается числовым значением от 0 до 255.

**WHITE (Белый) = (255, 255, 255)**



# Задание

---

Создайте переменную WHITE, которая будет указывать на значение (255, 255, 255).



# Задание. Решение

---

Создайте переменную WHITE, которая будет указывать на значение (255, 255, 255).

WHITE = (255, 255, 255)



# Задание

---

Добавьте в программу обработку игрового окна.

```
while game:
    for i in pygame.event.get():
        if i.type == pygame.QUIT:
            print("Exit")
            game=False
```

```
sc.fill(WHITE)
sc.blit(car1.image, car1.rect)
pygame.display.update()
```

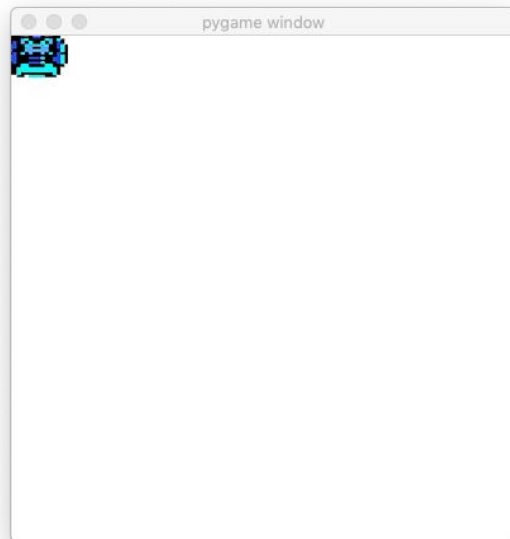


# Задание. Решение

---

Добавьте в программу обработку игрового окна.

```
while game:
    for i in pygame.event.get():
        if i.type == pygame.QUIT:
            print("Exit")
            game=False
    sc.fill(WHITE)
    sc.blit(car1.image, car1.rect)
    pygame.display.update()
```



# Вопрос

---

Чего не хватает нашему спрайту-машине?





# Вопрос-ответ

---

Чего не хватает нашему спрайту-машине?

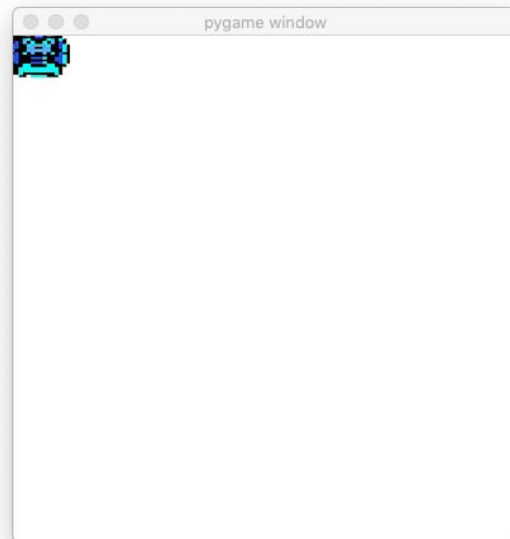
Движения



# Вопрос

---

Как должен двигаться автомобиль?

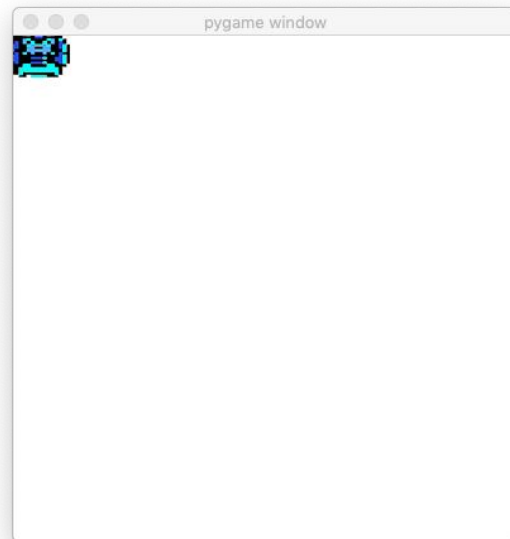


# Вопрос-ответ

---

Как должен двигаться автомобиль?

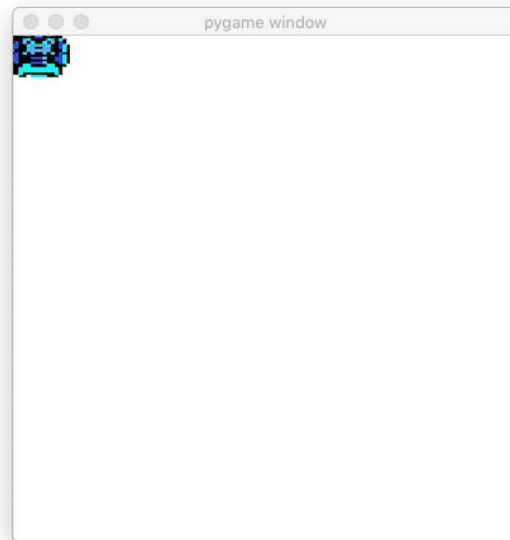
Сверху-вниз



# Вопрос

---

Какая координата отвечает для изменение положения объекта по вертикали?

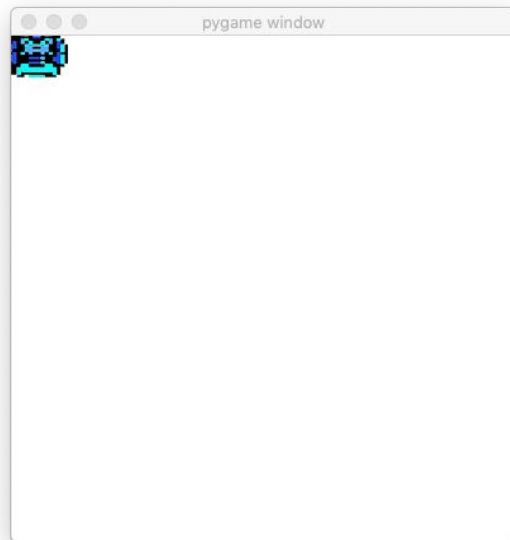


# Вопрос-ответ

---

Какая координата отвечает для изменение положения объекта по вертикали?

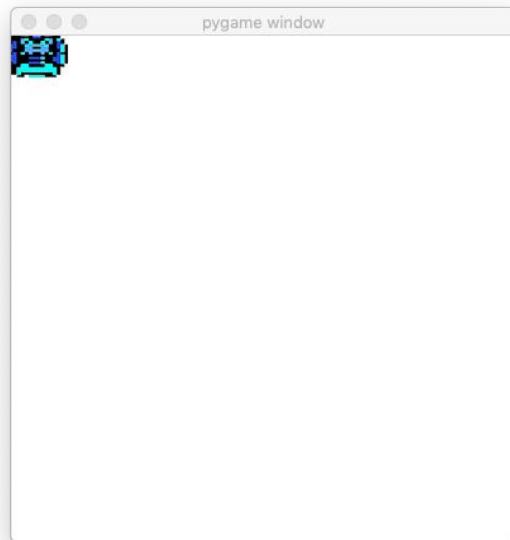
Y



# Вопрос

---

Чему равна координата **Y** внизу экрана?



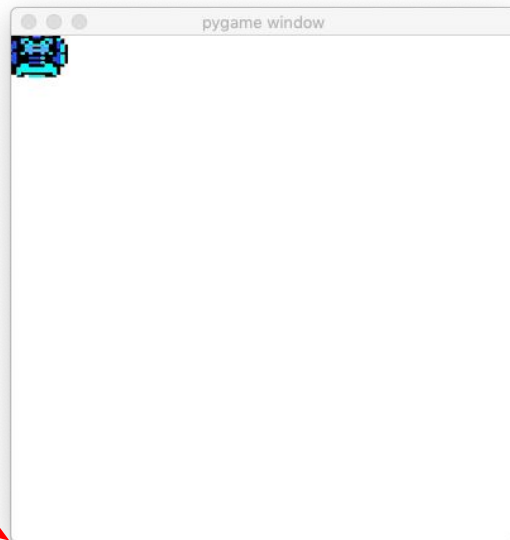
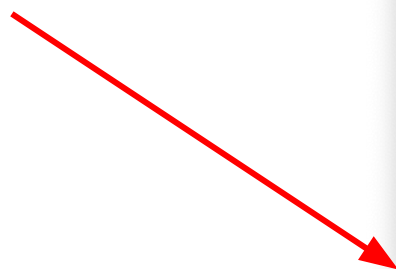
# Вопрос-ответ

---

Чему равна координата **Y** внизу экрана?

$H=400$

$Y=H$

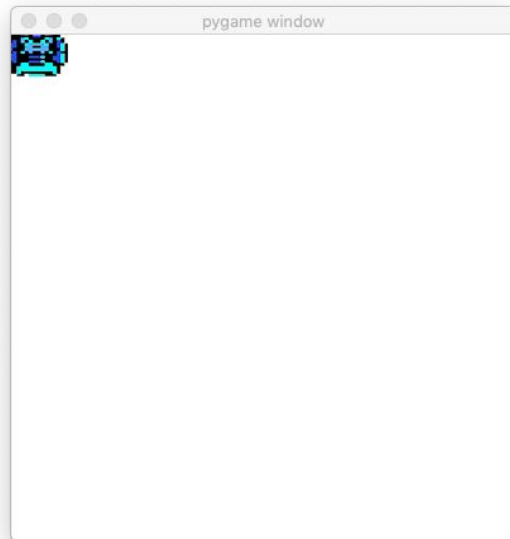


# Вопрос

---

Чему равна координата **Y** вверху экрана?

Y=?



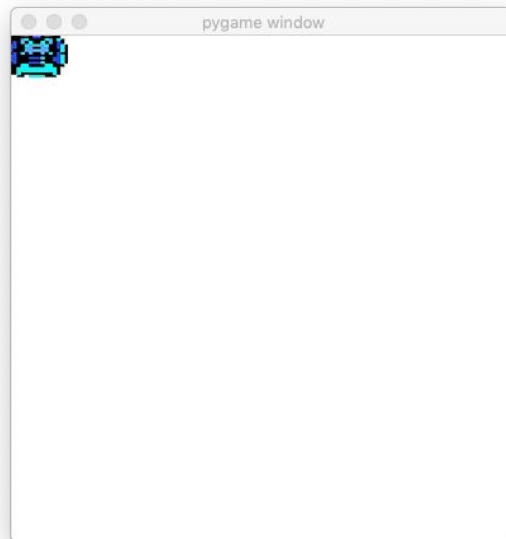


# Вопрос-ответ

---

Чему равна координата **Y** вверху экрана?

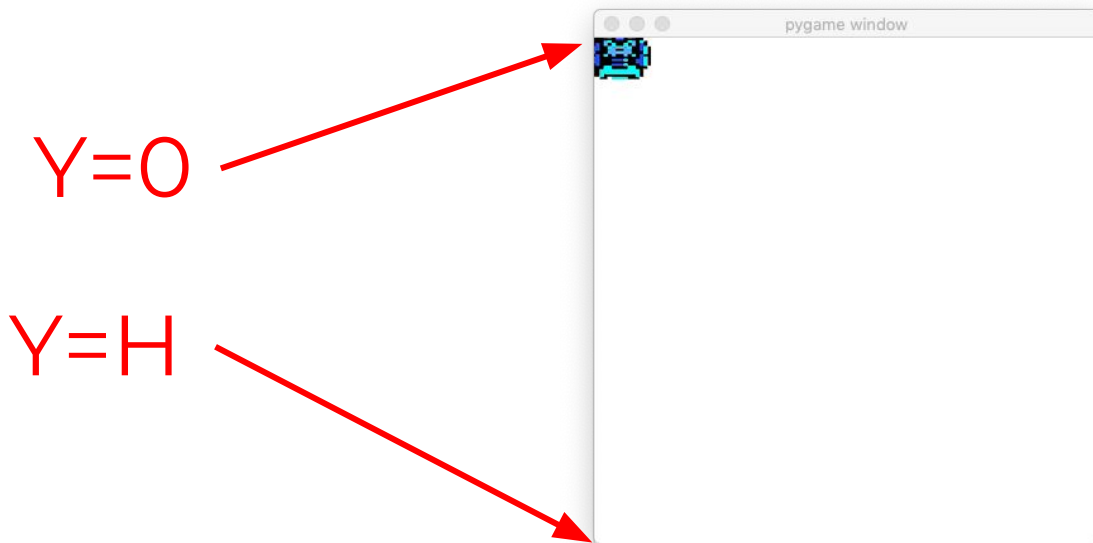
$Y=0$



# Вопрос

---

Как должна измениться координата **Y** спрайта car1, чтобы пройти путь от верха до низа окна?

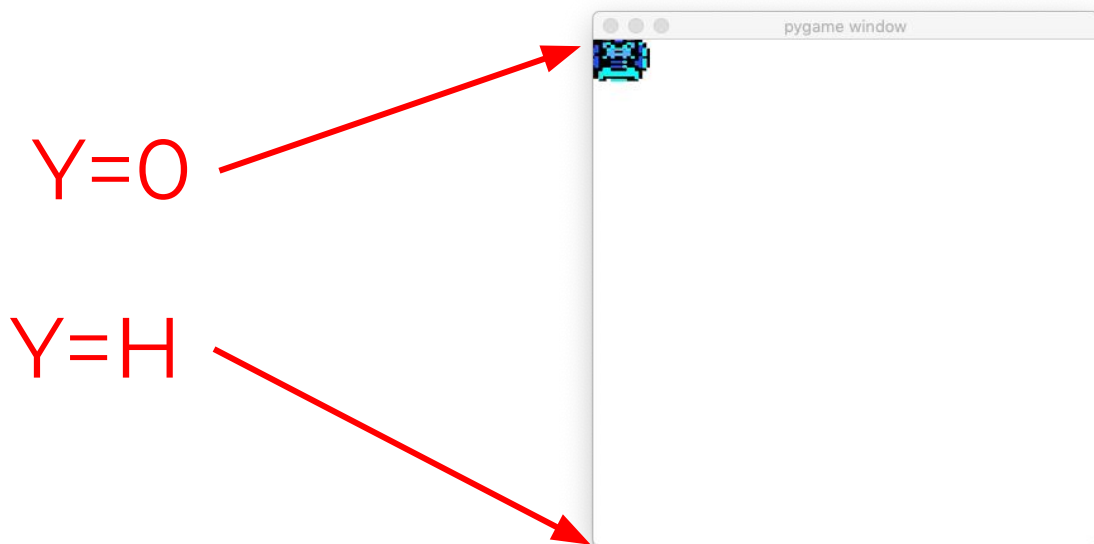


# Вопрос-ответ

---

Как должна измениться координата **Y** спрайта car1, чтобы пройти путь от верха до низа окна?

От 0 до H



# Задание

---

Добавьте в бесконечный цикл движение спрайта. Опишите работу программы.

```
if car1.rect.y < H:  
    car1.rect.y = car1.rect.y+2  
else:  
    car1.rect.y = 0  
    car1 = Car(20, 'Car1.png')
```



# Задание. Решение

---

Добавьте в бесконечный цикл движение спрайта. Опишите работу программы.

```
while game:
    for i in pygame.event.get():
        if i.type == pygame.QUIT:
            print("Exit")
            game=False
    sc.fill(WHITE)
    sc.blit(car1.image, car1.rect)
    pygame.display.update()
    if car1.rect.y < H:
        car1.rect.y = car1.rect.y+2
    else:
        car1.rect.y = 0
        car1 = Car(20, 'Car1.png')
```



# Вопросы

---

1. Что такое бесконечный цикл?
2. Что такое объект?
3. Что такое класс?
4. Как создать объект класса?
5. Для чего нужна функция `init()`?
6. За что отвечает `self`?
7. Что такое спрайт?
8. Как загрузить картинку для спрайта?

# Задание на дом. Уровень 1

---

Добавьте в проект еще одну машину с таким же изображением. Разместите второй спрайт в верхнем правом углу. Добавьте движение сверху-вниз.



## Задание на дом. Уровень 2

---

Добавьте в проект еще одну машину с таким же изображением. Разместите второй спрайт в нижнем правом углу. Добавьте движение снизу-вверх.

