

ДЕТСКАЯ ОНЛАЙН-ШКОЛА ПРОГРАММИРОВАНИЯ

HELLO WORLD



#ШПАРГАЛОЧКИ



ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

Базовый уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити в Telegram: https://t.me/hw_school





Пинг-понг часть 2





on_key_press - метод **arcade.Window**, срабатывающий при нажатии на клавишу клавиатуры. Принимает 2 аргумента - код нажатой клавиши и код нажатой клавиши-модификатора, которая меняет значение клавиши в сочетании с ней (**shift**, **ctrl**). Чтобы узнать, какая клавиша была нажата, нужно сравнить ее код и код нужной клавиши (все коды хранятся в константах библиотеки **arcade**):



```
def on_key_press(self, key, modifiers):
```

```
    if key == arcade.key.LEFT:
```

```
        self.bar.change_x = -SPEED_X
```



on_key_release - метод **arcade.Window**, срабатывающий при отпускании клавиши клавиатуры. Принимает 2 аргумента - код отпущенной клавиши и код отпущенной клавиши-модификатора, которая меняет значение клавиши в сочетании с ней (**shift**, **ctrl**). Чтобы узнать, какая клавиша была отпущена, нужно сравнить ее код и код нужной клавиши (все коды хранятся в константах библиотеки **arcade**):



```
def on_key_release(self, key, modifiers):
```

```
    if key == arcade.key.LEFT or key == arcade.key.RIGHT:
```

```
        self.bar.change_x = 0
```



arcade.check_for_collision - метод, проверяющий столкновение двух спрайтов. Принимает 2 спрайта, столкновение которых нужно проверить. Возвращает **True**, если столкновение было, **False** - если нет.

if arcade.check_for_collision(self.ball, self.bar):

self.ball.bottom = self.bar.top # ставим мяч на ракетку

self.ball.change_y = -self.ball.change_y



arcade.draw_text - метод, отрисовывающий текст на окне. Принимает следующие основные аргументы:

text - тот текст, который необходимо отобразить на экране

start_x - положение текста по горизонтали

start_y - положение текста по вертикали

color - цвет текста  HELLO WORLD

font_size - размер шрифта

```
arcade.draw_text(f"Score: {self.score}", 20, SCREEN_HEIGHT - 30, (0, 0, 0), 20)
```



stop - метод класса **arcade.Sprite**, останавливающий спрайт, например, когда игру нужно закончить. А чтобы отслеживать состояние игры, потребуется логическая переменная, в начале она равна True, когда игра должна остановиться, то в переменную состояния игры запишем False, метод обновления работает, если состояние игры True:

```
class Game(arcade.Window):
```

```
    def __init__(self, width, height, title):
```

```
        ...
```

```
        self.game = True
```

```
    def update(self, delta_time):
```

```
        if not self.game:
```

```
            ball.stop()
```