

УДК 004.42
ББК 32.372
Р64

Розанов А. Н., Жуков К. Л.

Р64 Учимся рисовать на компьютере кодом в P5JS. – М.: ДМК Пресс, 2022. – 168 с.: ил.

ISBN 978-5-93700-160-3

Цифровое творчество развивает личность, делает мир ярче, богаче и веселее. Многие современные художники создают свои картины с помощью компьютерных программ. В этой книге подробно показано, как школьник может освоить программирование на языке JavaScript и научиться создавать картины, персонажей и красивые пейзажи для собственной игры. Советы в конце книги помогают развивать «мышление в стиле компьютера» – ключевую компетенцию XXI века.

Издание адресовано школьникам среднего и старшего возраста, будет полезно студентам и учителям, а также родителям, заботящимся о развитии своих детей.

УДК 004.42
ББК 32.372

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-5-93700-160-3

© Розанов А. Н., Жуков К. Л., 2022
© Оформление, издание,
ДМК Пресс, 2022

Содержание

От издательства	5
К читателю	7
1. Знакомство со средой программирования P5JS	9
2. Рисуем флаг России	20
3. Рисуем солнце с помощью цикла FOR	29
4. Рисуем лоллипоп, или Как работать с цветом	35
5. Картинки и анимация спрайтов	43
6. Звёздная система и процедурный подход	54
7. Рисуем персонажей в 3D-графике	65
8. Рисование сложных фигур с помощью циклов	77
9. Советский истребитель МИГ-25, или Как загрузить готовый 3D-объект из файла	90
10. Генерация 3D-ландшафта	95
11. Осеннее дерево Пифагора, или Как я перестал бояться и полюбил рекурсию	108
12. Немного о «мышлении в стиле компьютера», или Computational thinking	119
Вместо послесловия	126

4 Содержание

Справочник по командам * P5JS	128
Общие команды	128
frameRate – установка FPS	128
frameCount – кадров отрисовано.....	129
width, height – ширина и высота холста	129
saveCanvas – скачать изображение холста	129
text – вывод текста на экран	130
translate – смещение начала координат.....	132
rotate – вращение начала координат.....	133
random – случайное число.....	134
noise – шум Перлина	136
Работа с цветом	138
colorMode – цветовая модель	138
fill – цвет заливки.....	140
stroke – цвет карандаша.....	140
strokeWeight – толщина линии	141
background – заполнение холста.....	142
lerpColor – оттенок между цветами	143
2D-графика.....	144
circle – окружность.....	144
ellipse – овал	145
rect – прямоугольник	146
square – квадрат	147
quad – четырёхугольник	148
triangle – треугольник	149
line – линия	150
point – точка.....	151
arc – дуга.....	152
vertex – вершина в фигуре.....	154
bezier – кривая Безье	156
3D-графика.....	157
rotateX, rotateY, rotateZ – вращение системы координат вокруг оси.....	157
box – куб	158
sphere – сфера.....	159
cylinder – цилиндр	160
torus – бублик	161
ellipsoid – эллипсоид	162
cone – конус	163
orbitControl – осмотр сцены.....	164
loadModel – загрузка модели из файла.....	165
Контакты с авторами	167

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки всё равно случаются. Если вы найдёте ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остаётся насущной проблемой. Издательство «ДМК Пресс» серьёзно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

К читателю

*Спасибо моему другу и вдохновителю
Жукову Кириллу Леонидовичу.*

*Спасибо педагогам, благодаря которым
образование детей стало очень увлекательным:
Даниэлю Шиффману, Митчелу Резнику, Натали Раск,
а также Сеймуру Пайперту за его «Черепашку»*

Эта книга посвящена информатике, а точнее одному из её разделов – программированию графики, наверное, самому интересному и увлекательному делу.

Перед людьми, заинтересованными в изучении программирования, встаёт задача выбора языка. Языки программирования требуют от программиста различного уровня внимания к деталям при реализации алгоритма. В зависимости от области применения и поставленных задач подбирается подходящий язык программирования: удобный и простой. Правильный выбор языка сэкономит время и увеличит производительность. Каждый день не только совершенствуются старые языки, но и разрабатываются новые, поэтому выбор для изучения велик. Изучение языков программирования открывает новые возможности и перспективы для каждого человека. Развитие навыков поможет найти работу мечты в каждой стране мира или же работать удалённо.

Мы описали методы создания графики на компьютере с помощью популярного языка программирования JavaScript. А что-

бы читателю было проще пользоваться книгой, все исходники программ расположены в облаке на популярной платформе P5JS по адресу <https://p5js.org>.

Также все материалы из книги можно найти на странице в соцсети «ВКонтакте»: <https://vk.com/itkvant>.

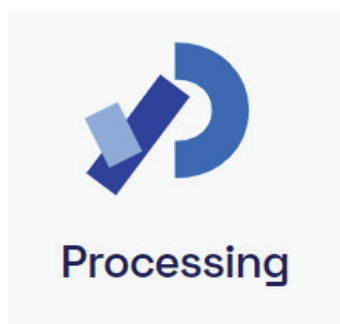
Цель авторов – увлечь читателя интересными и яркими примерами программ, развить такой важный аспект жизни современного человека, как computational thinking (мышление в стиле компьютера, или вычислительное мышление). Это подобие алгоритмического мышления из предмета информатики, однако применяется не только в программировании, но и в обыденной жизни.

Россия всегда славилась своими учёными и инженерами, пришло твоё время, дорогой читатель!

1

Знакомство со средой программирования P5JS

Родоначальником среды P5JS является среда **Processing**, которая в 2022 году вышла в 4-й версии.



Среда обладает большим набором команд, а для программирования используется популярный язык программирования **JavaScript**. Таким образом, программист одновременно учится рисовать линии, круги, прямоугольники и осваивает правила написания команд языка JavaScript. Такие правила называют синтаксисом языка программирования. Сами программы, написанные в среде, называют скетчами, от слова **sketch** – набросок, эскиз.

Для начала знакомства перейдём на сайт среды по адресу: <https://p5js.org/>.



Рис. 1. Главная страница среды P5JS

Сайт выполнен в стиле «минимализма», чтобы учиться было удобно на любом устройстве. На данный момент нет официальной поддержки русского языка, но можно воспользоваться онлайн-переводом, чтобы прочитать вводное слово на главной страничке.

Мы для учёбы в основном будем использовать редактор (**Editor**) и раздел справки (**Reference**), но начать изучать сайт хочется с раздела примеров (**Examples**).

Перейдите по ссылке <https://p5js.org/examples/>.



Рис. 2. Страница с примерами программ

Для каждой возможности и команды языка есть пример, наглядно поясняющий принцип работы. Все примеры объединены в блоки, например блок **Simulate** (Симуляция). В нем можно найти известную программу «клеточный автомат» **Game of Life** («Игра “Жизнь”»), которая некогда наглядно продемонстрировала учёным, что жизнь во Вселенной может подчиняться вполне понятным и простым законам. Внимательно посмотрите на рис. 2, найдите пункт **Game of Life** и откройте программу, или просто перейдите по ссылке <https://p5js.org/examples/simulate-game-of-life.html>.

Game of Life

[< Back to Examples](#)

A basic implementation of John Conway's Game of Life CA

(natureofcode.com)

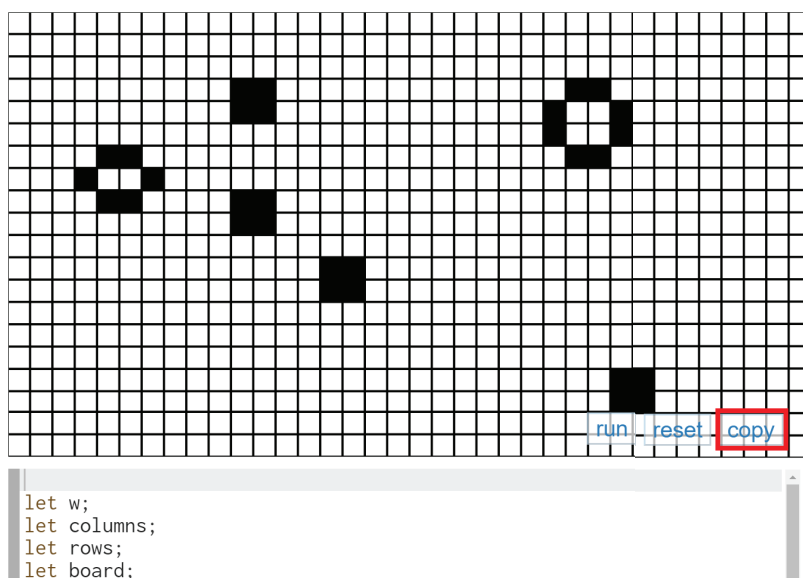


Рис. 3. Пример «Игры “Жизнь”»

Скопируйте программный код этого примера в **буфер обмена**. Это можно сделать с помощью кнопки **copy** (копировать) на рис. 3 справа или выделив весь текст и нажав одновременно сочетание клавиш **Ctrl+C**.

С главной странички сайта можно войти в редактор **Editor**: <https://editor.p5js.org>.

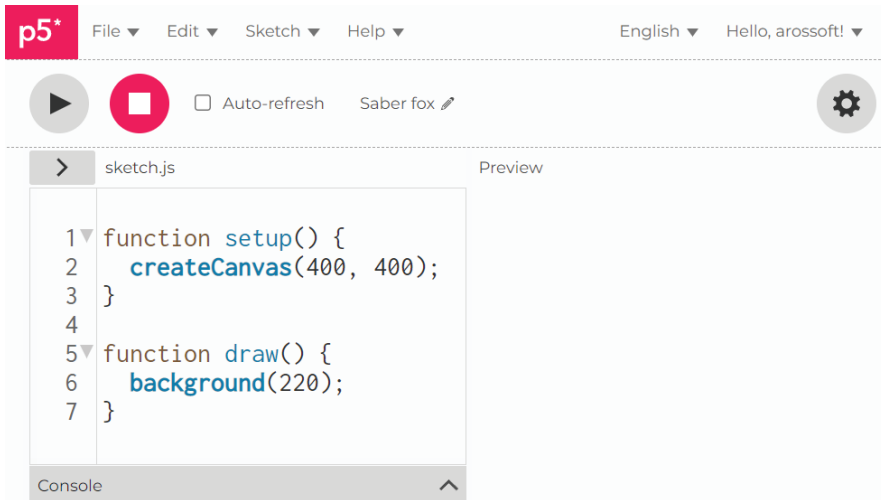


Рис. 4. Окно редактора и стартовый шаблон программы

Вы увидите стартовый шаблон с короткой программой, как на рис. 4. Удалите весь текст стартового шаблона и вставьте текст программы «Игра “Жизнь”» из буфера обмена, нажав сочетание клавиш **Ctrl+V**.

Запустите программу с помощью красной кнопки с треугольником. Такие кнопки по привычке называют **Play** (Играй). И действительно, «Игра “Жизнь”» запустится. Можно остановить программу соседней кнопкой **Stop** (Остановить) с квадратом внутри.

Попробуйте изменить параметр $w = 20$ в строке № 9 на большее и меньшее числа и полюбуйте игрой клеток.

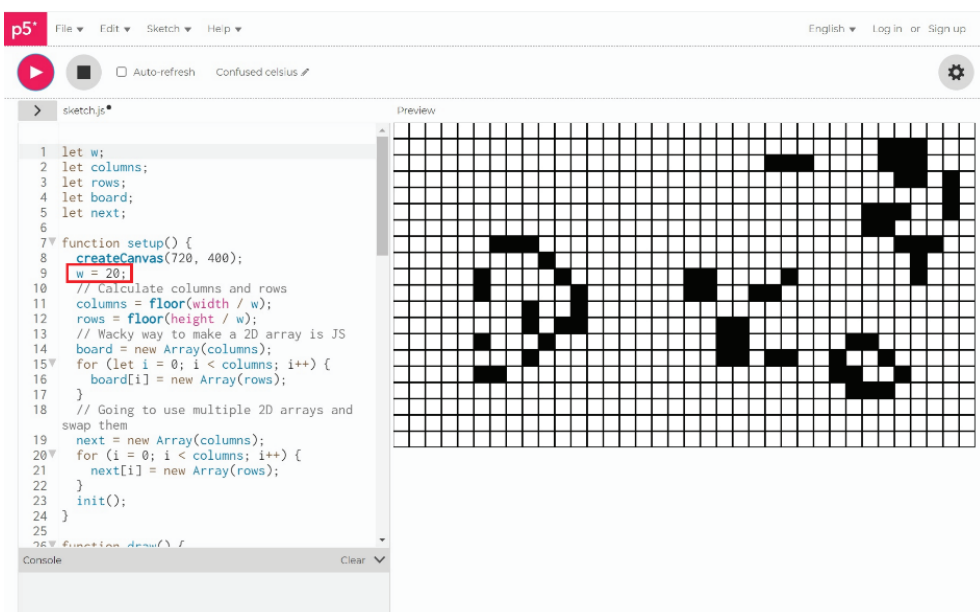
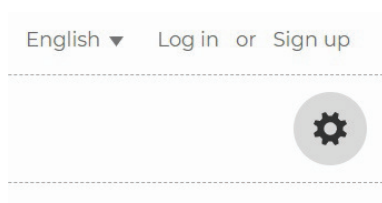
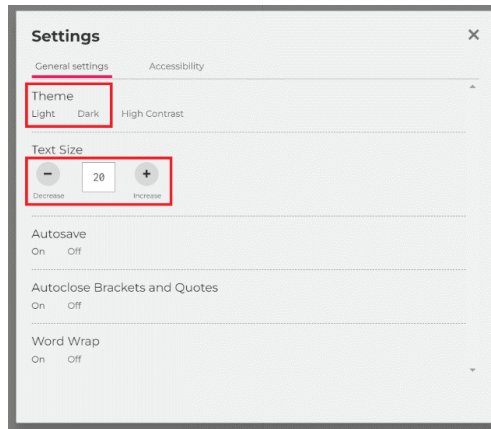


Рис. 5. «Игра «Жизнь»» в работе

Чтобы сохранить программу в облаке, необходимо авторизоваться или зарегистрироваться на сайте в правом углу удобным способом.



За кнопкой с шестерёнкой спрятаны настройки, которые помогают выбрать наиболее удобный режим отображения странички. Можно увеличить или уменьшить шрифт, добавить опции.



Можно переключить тему на **тёмную** для ночной работы и на **светлую**, когда работаете в освещённом помещении или днём.

Всегда старайтесь настроить среду под освещение и размер экрана, сделайте шрифт крупнее. Это сохранит ваше здоровье и уменьшит усталость.

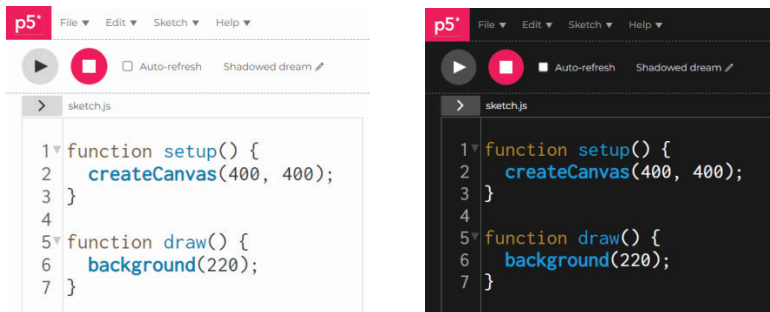


Рис. 6. Светлая и тёмная темы

После успешной регистрации и авторизации на сайте сохранить программу можно через верхнее меню **File** (Файл) ⇒ **Save** (Сохранить). Но прежде желательно дать программе понятное имя, для этого нажмите на карандаш.

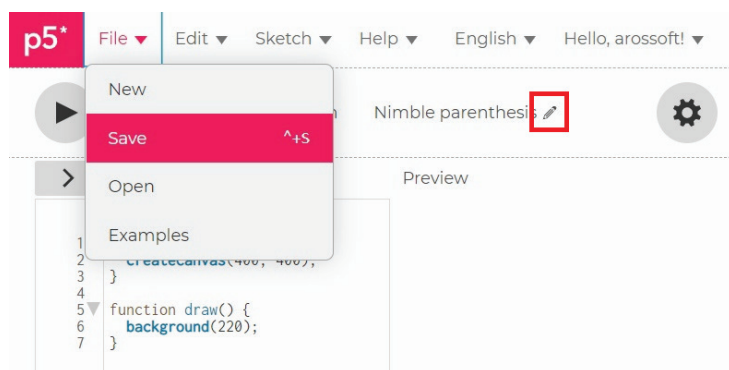


Рис. 7. Сохраните программу с вашим названием

Предупреждение!!! Старайтесь сохранять программу перед стартом, ведь она может работать неправильно и «подвесить» страницу браузера. Тогда набранный текст программы пропадет и его придется набирать заново.

Теперь нажмите на треугольник в правом углу и перейдите в меню **My Sketches** (Мои программы), там вы и найдете сохранённую программу.

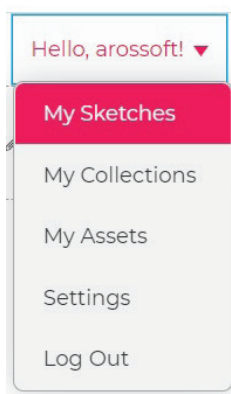
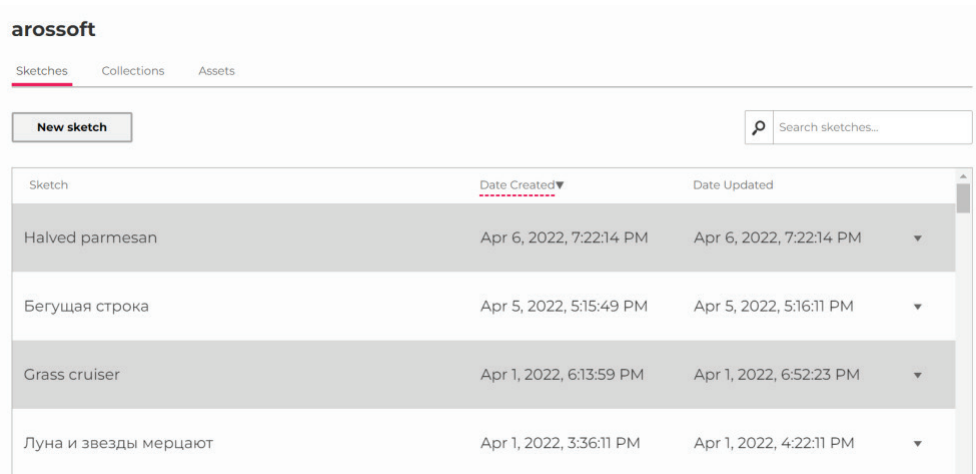


Рис. 8. Место хранения ваших программ



На моей памяти есть примечательный случай.

Один мой знакомый, очень уважаемый мной инженер, как-то позвонил. Тогда ему было 12 лет. Он сказал по телефону, что его компьютер завис и он не знает, что делать. Перепробовав все возможности вывести компьютер из «зависания», я предложил ему нажать и подержать кнопку включения 7 секунд. Он сказал, что экран погас. Компьютер выключился, и я предложил включить его снова.

К его изумлению, на экране появился пустой рабочий стол, и тогда он спросил меня: «А как же игра, которую я делал?» Я переспросил: «Ты сказал “игра”? Ты знаешь, как делать игры?» Виталий, так звали мальчика, сказал, что делает игру «Пять ночей с Фредди».

И тогда я поинтересовался, на каком языке он пишет игру. К моему изумлению, он ответил, что ещё не знает языков программирования. «Но тогда как же ты делаешь игру?» – не выдержал я. Виталий ответил, что делает её в программе **PowerPoint**. Оказалось, что для каждой игровой ситуации он создал отдельный слайд презентации с переходами между ними с помощью кликов мышью. Его игра состояла из более чем 400 слайдов!!!

Проблема была в том, что он начал делать игру, даже не думая сохранить файл презентации. Я был потрясён. На следующий день я был потрясён окончательно – тем, что Виталий без долгих раздумий взял и написал игру с начала, но теперь уже оптимизировал количество слайдов до около 200.

Надеюсь, этот пример из моей жизни всплывёт в вашей памяти, когда вы впервые напишите бесконечный цикл, не сохранив текст программы. Как и то, что это вас не остановит, как истинно творческого человека не останавливают тягости и невзгоды. А уж в жизни любого программиста их бывает навалом.

В завершение этого ознакомительного путешествия по сайту хочется отдельно выделить одну важную возможность. В жизни любого человека важно уметь «делиться» своим опытом. Например, вы можете поделиться с друзьями и другими программистами своей программой. Кнопка находится в верхнем меню **File** (Файл) ⇒ **Share** (Поделиться) и доступна после авторизации. Отправьте ссылку своим друзьям и спросите, что они думают о среде P5JS.

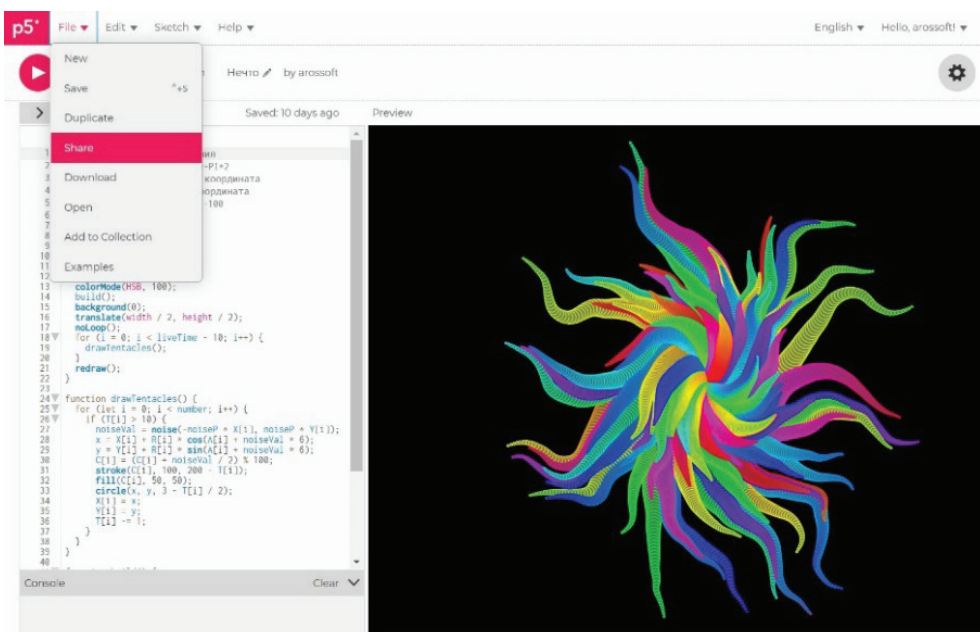
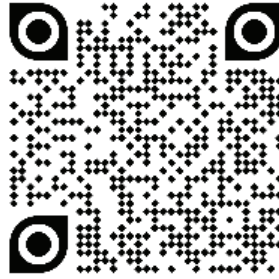


Рис. 9. Кнопка **Share** позволяет делиться исходным кодом или демонстрацией программы

Кстати, скетч на рисунке сделал мой ученик, школьник Костя, когда я поделился с ним моей версией. Теперь вы можете открыть этот пример по ссылке <https://editor.p5js.org/arossoft/sketches/ohLwd4lv3>.



Хочу пожелать вам успехов в изучении программирования и предлагаю написать свою первую программу на JavaScript в среде P5JS.

2

Рисуем флаг России

(Теги: Reference, setup, draw, createCanvas, RGB, line, stroke)

Многие начинающие программисты сделали свои первые шаги в программирование через графику с помощью простых примеров, например рисовали флаги своих стран. Российский флаг рисуется очень просто, для этого достаточно нарисовать три цветных прямоугольника. Давайте добавим движения, пусть флаг развевается на ветру.



Рис. 10. Флаг России, нарисованный в P5JS

Зайдите на сайт в раздел редактора **Editor** (<https://editor.p5js.org>).
Наберите листинг программы вместо строк стартового шаблона.

```

:1  function setup() { // функция настройки
:2      L = 400;      // размер флага
:3      createCanvas(400, 400); // размер окна
:4  }
:5
:6  function draw() { // функция рисования
:7      x = 50;
:8      background(220); // серый фон
:9
:10     while (x < L) { // цикл "пока"
:11         y = 25*sin((x + frameCount)/45);
:12
:13         stroke(255); // белый
:14         line(x, 100 + y, x, 150 + y);
:15         stroke(0, 0, 255); // синий
:16         line(x, 151 + y, x, 201 + y);
:17         stroke(255, 0, 0); // красный
:18         line(x, 202 + y, x, 252 + y);
:19         x++;
:20     }
:21 }

```

Программа состоит из двух блоков:

- функции настройки (**setup**);
- функции рисования (**draw**).

Если ваша программа не содержит анимацию и не меняет картинку на экране, тогда вполне можно обойтись одним блоком **setup**, без **draw**. Но большинство примеров в книге будут динамическими (т. е. с изменяющейся картинкой), и для них потребуется функция **draw**, которая будет вызываться до 60 раз

в секунду. Её задача – плавная перерисовка изменений на холсте (области рисования на экране).

Пока мы создаем 2D-графику, нам достаточно вызвать функцию **создания** (create) **холста** (Canvas) и указать размеры графического окна в пикселях. Точки на экране называются пикселями.

```
3 createCanvas(400, 400);
```

Этой командой мы сообщаем компьютеру, что хотим создать окно размером 400 пикселей в ширину и 400 пикселей в высоту.

По мере изучения языка вы будете встречать новые команды. Для всех команд в системе P5JS есть описание на английском и некоторых других языках. Чтобы найти описание команды **CreateCanvas** (Создать холст), можно в верхнем меню нажать пункт **Help** (Помощь) ⇒ **Reference** (Справочник). Достаточно ввести первые буквы названия команды в поле для поиска и выбрать нужный вариант, чтобы попасть на страницу с описанием команды: <https://p5js.org/reference/#/p5/createCanvas>.

p5.js

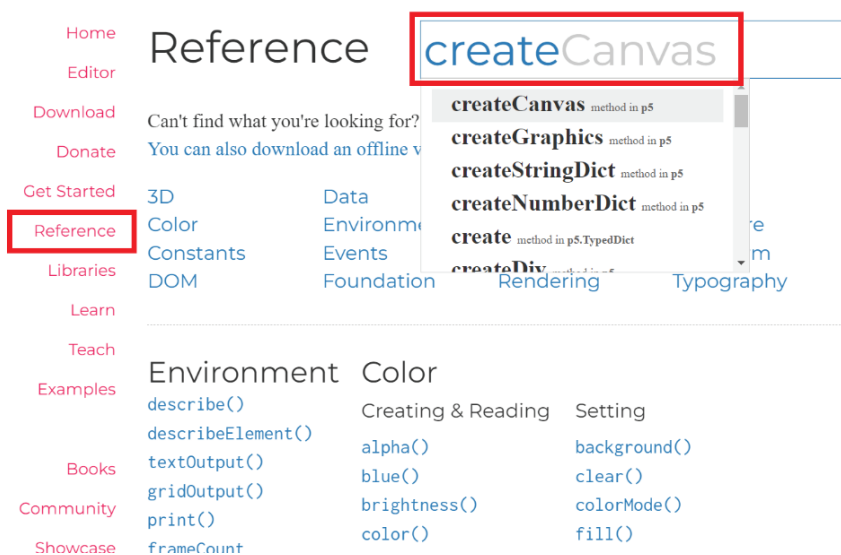


Рис. 11. Встроенный справочник по командам

Также в справочник с командами можно попасть и с главной страницы сайта с помощью кнопки **Reference** (Справочник) слева в меню. В строке поиска обычно начинают набирать название команды или находят интересующую команду в большом списке, разбитом на категории. Чтобы работать со справочником было проще, в конце книги есть список основных команд с описанием на русском языке.

Программы сохраняют в памяти компьютера разные величины (числа, имена и прочее), для этого используются так называемые *переменные*. Каждой переменной программист дает уникальное имя. Оно может отражать некоторый смысл или просто быть взято с потолка.

Совет. Как правильно назвать переменную?

1. Нельзя использовать служебные слова, которые применяются для команд языка, констант вроде числа π и встроенных атрибутов, например `width`.
2. Название не должно начинаться с цифры.
3. Можно объединять слова с помощью нижнего подчеркивания: `my_name`.
4. Можно дать переменной имя в «верблюжьем стиле» (Camel style): `myName`.
5. А можно просто так: `i`, `abc`, `x1`, `dy`.

О том, как лучше всего называть переменные, можно подробно почитать в Википедии: <https://ru.wikipedia.org/wiki/CamelCase>.

В нашем примере переменная **X** названа не случайно, она обозначает смещение флага по оси X, т. е. по горизонтали на холсте.

```
⋮ 7      x = 50;      ⋮
```

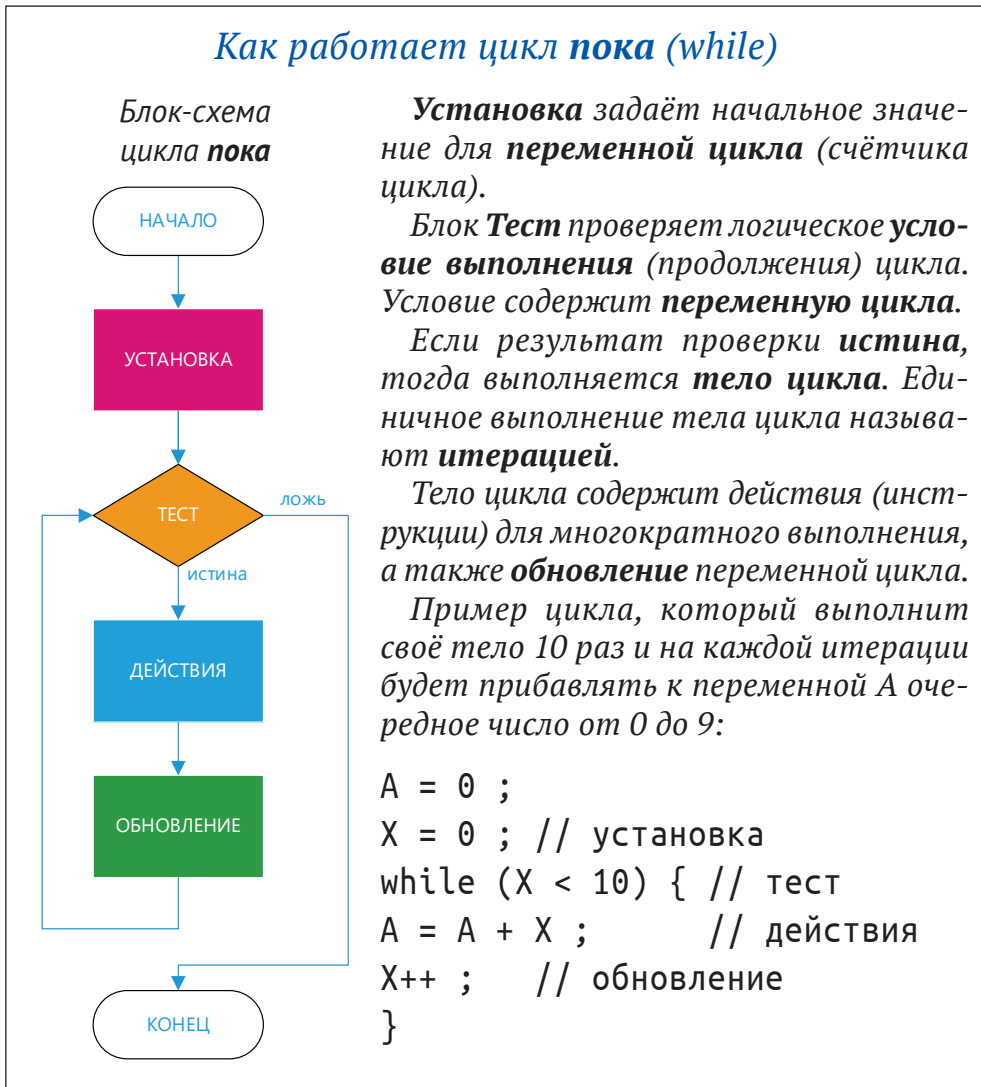
Таким образом, рисование флага начнётся с отступом от левого края холста в 50 пикселей.

За очистку экрана и заливку фона цветом отвечает следующая команда:

```
⋮ 8      background(220);      ⋮
```

Здесь светло-серый цвет задаётся числом 220, что соответствует градации одного из 253 оттенков серого. Белый цвет можно задать числом 255, а чёрный – 0.

Наш флаг будет состоять из отдельных вертикальных линий. Чтобы нарисовать их все по очереди слева направо, мы используем цикл **пока** (while). Знание, как работает цикл, – одно из самых важных в программировании. Постарайся перечитать следующий текст несколько раз, чтобы запомнить основные понятия.



```

:10 while (x < L) {
  
```

Цикл будет выполнять команды *тела цикла*, пока верно условие в круглых скобках. В нашем случае цикл будет выполнять

команды в фигурных скобках, пока x не вырастет до величины L (от 50 до 400). Зоркий глаз хакера заметит, что само число 400 не войдет в этот диапазон.

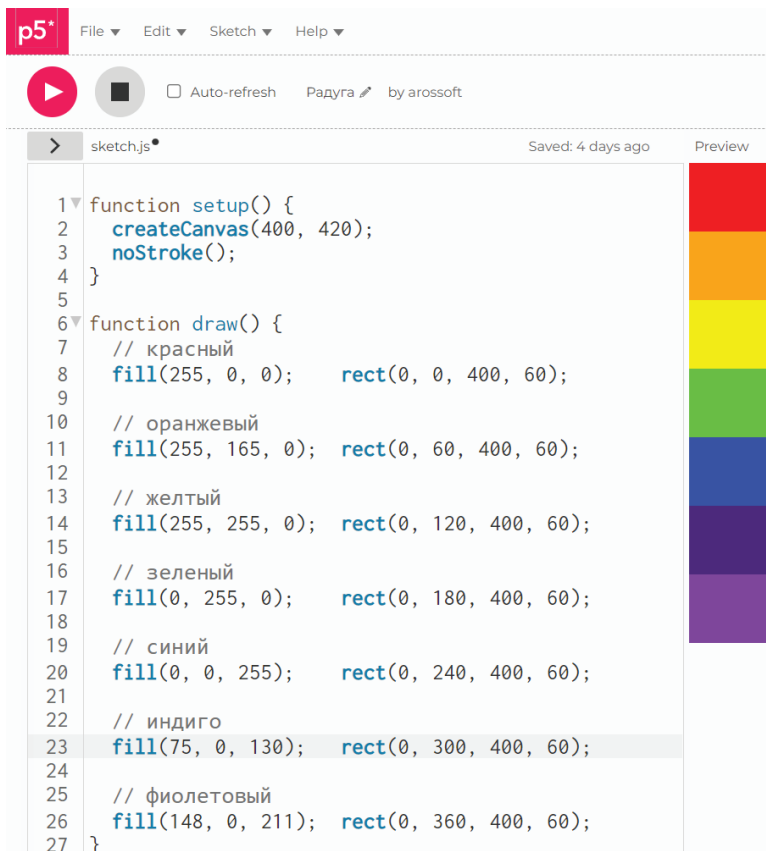
Если ты не успел познакомиться с тригонометрией в школе, не беда. Эта строка вычисляет изгиб флага с помощью математики.

```
11    y = 25 * sin((x + frameCount) / 45);
```

Попробуй заменить числа 25 и 45 на другие – и поймешь, для чего программисты вводят в формулы коэффициенты.

Чтобы задать цвет полос флага, применяется команда **stroke** с параметром. Им может быть значение в диапазоне 0–255 для выбора оттенка серого. Но если вы хотите задать один из цветов радуги, тогда нужно написать его код в формате **RGB**.

Пример для рисования радуги:



```

1 function setup() {
2   createCanvas(400, 420);
3   noStroke();
4 }
5
6 function draw() {
7   // красный
8   fill(255, 0, 0);   rect(0, 0, 400, 60);
9
10  // оранжевый
11  fill(255, 165, 0);  rect(0, 60, 400, 60);
12
13  // желтый
14  fill(255, 255, 0);  rect(0, 120, 400, 60);
15
16  // зеленый
17  fill(0, 255, 0);    rect(0, 180, 400, 60);
18
19  // синий
20  fill(0, 0, 255);    rect(0, 240, 400, 60);
21
22  // индиго
23  fill(75, 0, 130);   rect(0, 300, 400, 60);
24
25  // фиолетовый
26  fill(148, 0, 211);  rect(0, 360, 400, 60);
27 }

```



```

stroke(255);           // белый
stroke(0, 0, 255);    // синий
stroke(255, 0, 0);    // красный

```

Три числа в скобках задают яркость компонентов цвета, соответственно: **Red** – красный, **Green** – зелёный, **Blue** – синий. Цвет линий (обводки) задаётся командой **stroke**, а цвет заливки фигур – командой **fill**.

Чтобы нарисовать на экране линию из одной точки в другую, нужно воспользоваться командой **line**. Координаты точек на экране задаются двумя парами (x_1, y_1, x_2, y_2).

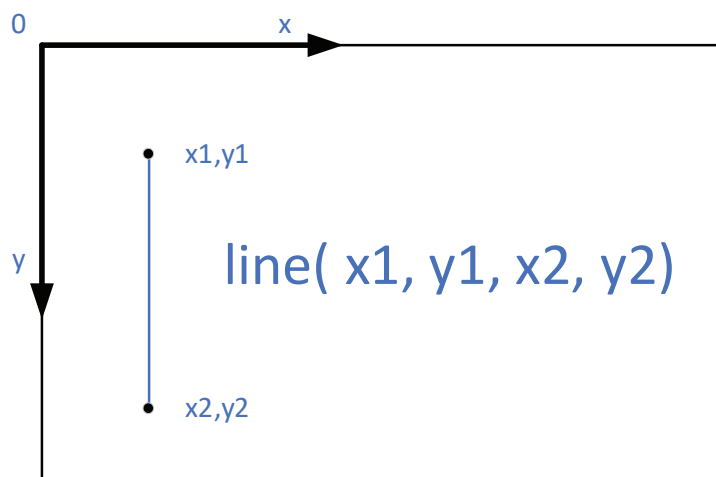


Рис. 12. Модель рисования линии на холсте

Хотите знать больше – ищите описание команды **line** в справочнике, во второй половине книги в разделе **2D-графика**.

```

14 line(x, 100 + y, x, 150 + y);

```

Чтобы нарисовать полосу другого цвета ниже, достаточно добавить смещение по координате y для точек на концах линии. В следующей команде рисования синей линии 100 увеличили на 51, 150 увеличили до 201, поэтому на холсте она рисуется ниже.

Если мы не изменим переменную x в конце тела цикла, то он, как говорят некоторые программисты, «зациклится», и программа зависнет. Правильно говорить, что цикл станет беско-

нечным. Скорее всего, зависнет и вкладка браузера. Увеличим переменную `x` на 1 с помощью операции `++`.

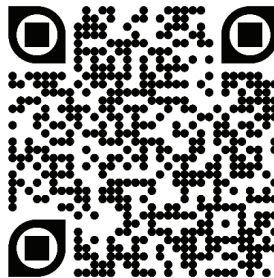
```
19     x++;
```

Программа завершается фигурной скобкой. Постарайтесь ответить: для чего она нужна?

```
21     }
```

Маленькая подсказка: количество открытых и закрытых скобок всегда должно быть равно. Спросите других начинающих или профессиональных программистов об этом. Они с радостью поделятся своими познаниями по поводу скобок.

Рабочая версия программы доступна по адресам <https://editor.p5js.org/arossoft/sketches/o50bLSZXU>; <https://clck.ru/enaRK> или по QR-коду:



Один из наших учеников предложил интересный эффект тени. Костя добавил в команды `stroke` смещение с помощью переменной `y`.

```
stroke(255 - y * 2);          // белый
stroke(0, 0, 255 - y * 2);   // синий
stroke(255 - y * 2, 0, 0);   // красный
```

Чем ниже находится вертикальная линия на флаге, тем она темнее.

Вопрос для эксперимента: если твой монитор формата 4К, какие ширину и высоту холста в пикселях можно написать в команде `createCanvas()`?

Задания для самопроверки

1. Нарисуй флаг другой страны.
2. Нарисуй флаги нескольких стран на одном экране один за другим.

Челлендж («Тебе слабо это сделать?»): измени программу, чтобы вместо флага получилось море.

Свое решение можно опубликовать на нашей страничке в соц-сети «ВКонтакте» (<https://vk.com/itkvant>) с пометкой «Я смог: нарисовать море». Ну или как-то ещё.

Продвинутый пример флага: <https://editor.p5js.org/arossoft/sketches/Z44fOVB2Q>.

