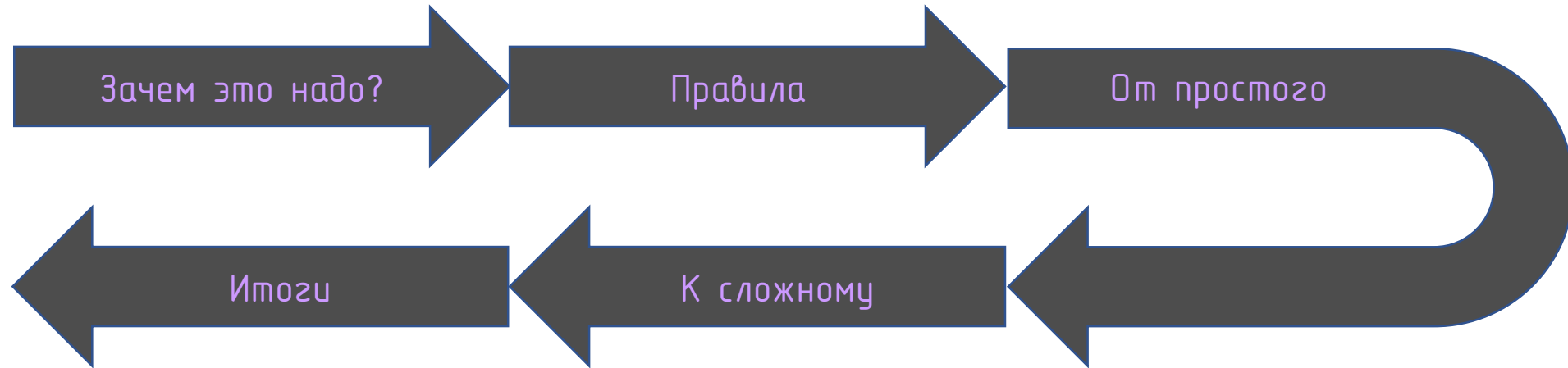


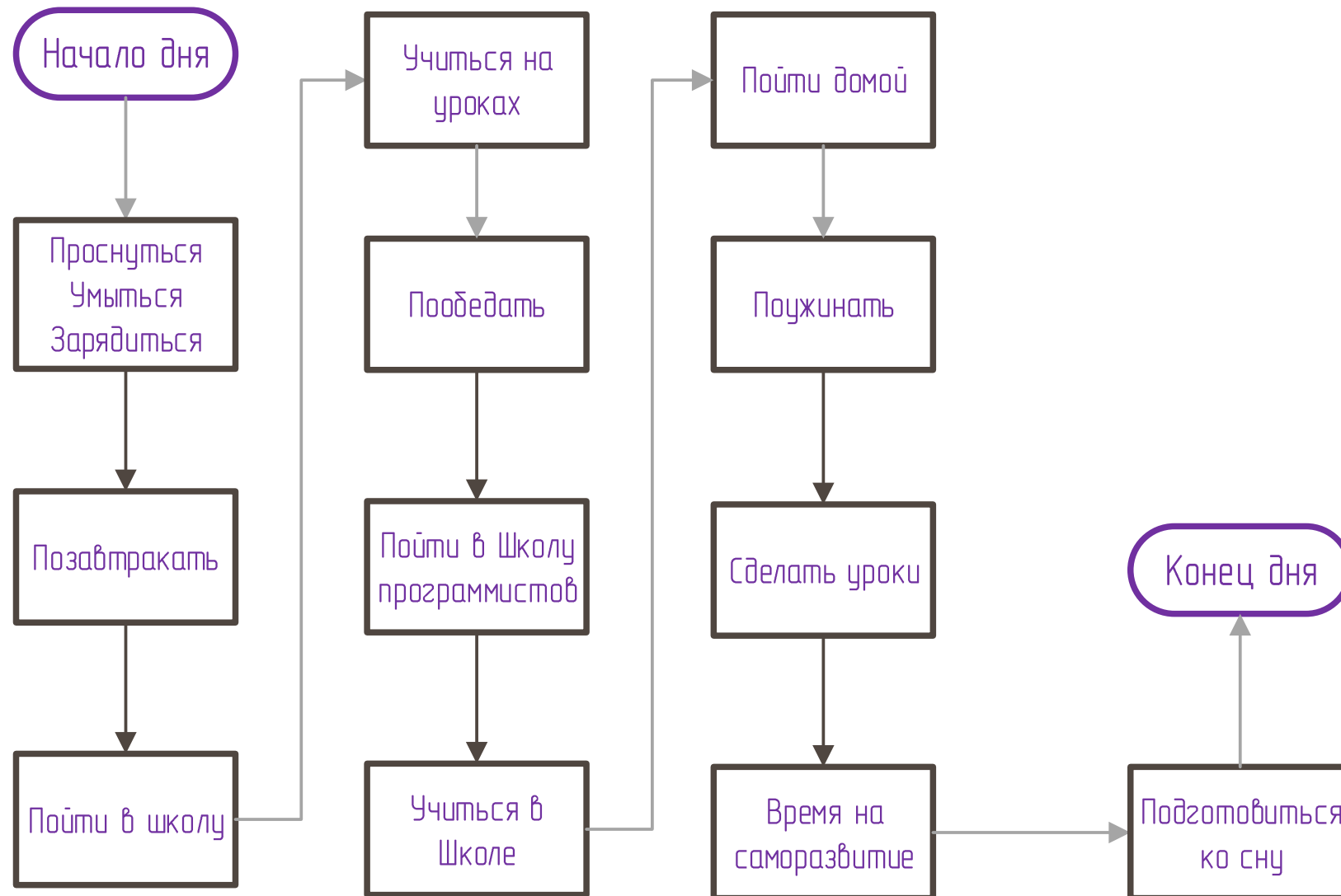
# Функции в JAVA SCRIPT

Амельченко Евгений Олегович  
evgenameo@gmail.com

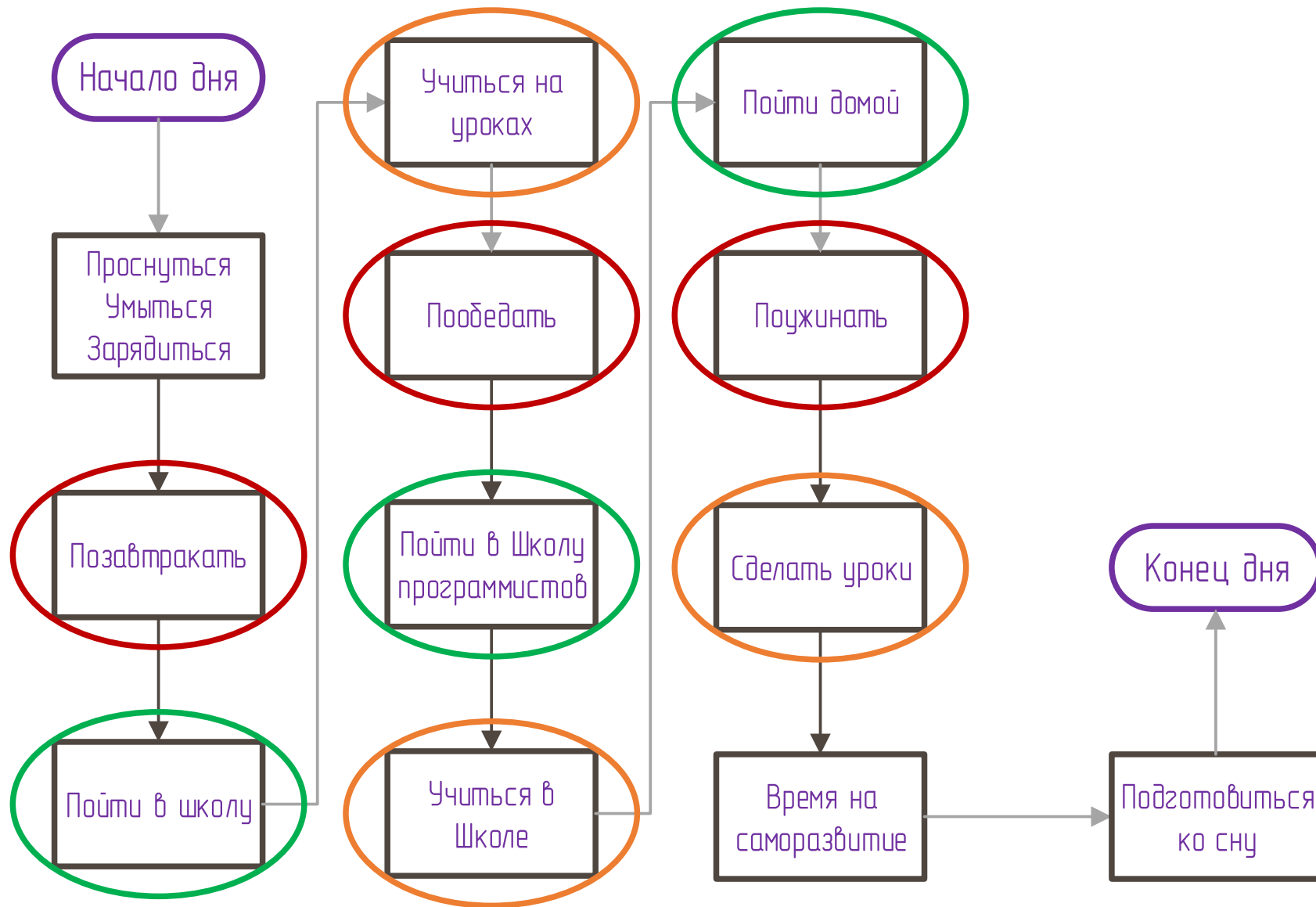
# Содержание



# Функции в нашей жизни: идеальный ребенок



# Функции в нашей жизни: идеальный ребенок



# Функции делятся

Встроенные

Пользовательские

Возвращающие

Не возвращающие

Принимающие

Не принимающие

# Синтаксис

```
function имя(параметры) {  
    ...тело...  
}
```

- Имена чувствительны к регистру (y и Y это две разных переменных)
- Имена должны начинаться с буквы или символов "\$" и "\_"
- Имя может состоять из любых цифр и букв латинского алфавита, а также символов "\$" и "\_"
- В качестве имени нельзя использовать зарезервированные и ключевые слова

# Рекурсия, или как мы искали клад

```
function pow(x, n) {  
    let result = 1;  
    for (let i = 0; i < n; i++) {  
        result *= x;  
    }  
    return result;  
}  
alert( pow(2, 3) ); // 8
```

```
function pow(x, n) {  
    if (n == 1) {  
        return x;  
    } else {  
        return x * pow(x, n - 1);  
    }  
}  
alert( pow(2, 3) ); // 8
```

# Лукбез: факториал

$$n! = n * (n - 1)!$$

$$0! = 1$$

$$1! = 1$$

$$2! = 2 * 1 = 2$$

$$3! = 3 * 2 * 1 = 6$$

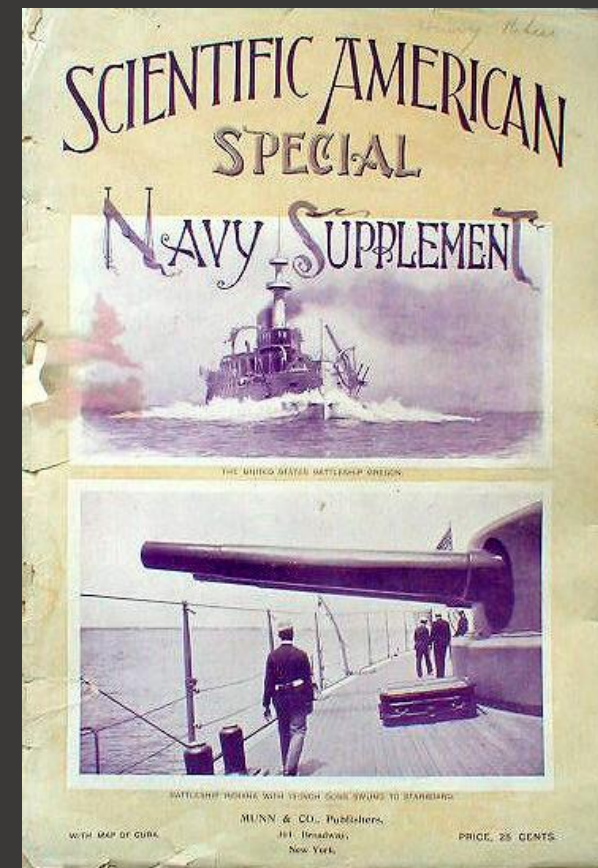
$$4! = 4 * 3 * 2 * 1 = 24$$

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$



# Поиграем в «Жизнь»

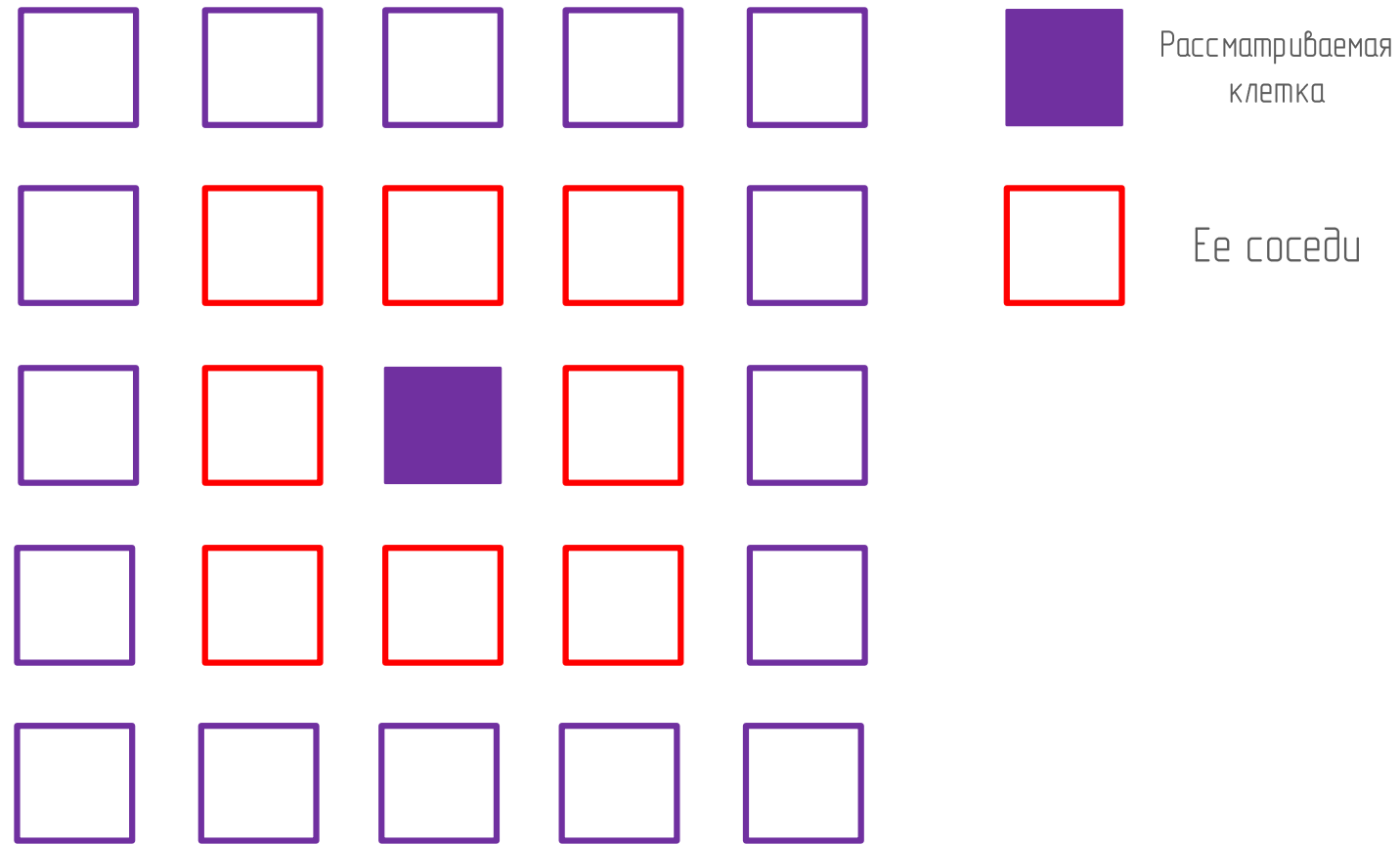
1. Место действия этой игры — «вселенная» — это размеченная на клетки поверхность или плоскость.
2. Каждая клетка на этой поверхности может находиться в двух состояниях: быть «живой» или быть «мёртвой». Клетка имеет восемь соседей, окружающих её.
3. Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:
  - в пустой (мёртвой) клетке, рядом с которой ровно три живые клетки, зарождается жизнь;
  - если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае, если соседей меньше двух или больше трёх, клетка умирает («от одиночества» или «от перенаселённости»).



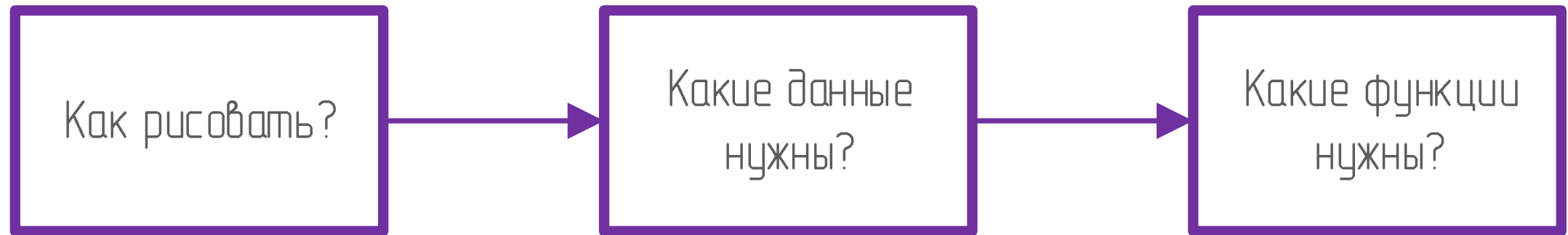
Источник:

[https://ru.wikipedia.org/wiki/Scientific\\_American](https://ru.wikipedia.org/wiki/Scientific_American)

# Поиграем в «Жизнь»



# Поиграем в «Жизнь»



# Поиграем в «Жизнь»: Отрисовка

В HTML:

```
<body onload="main()">
  <!--Когда загрузиться страница -- произойдет вызов функции main()-->
  <script src="./index.js"></script>
  <!--Подключаем файл с нашим будущим скриптом-->
  <canvas id="plot" width="1400" height="700"></canvas>
  <!--canvas используется для отрисовки пикселей (высота 700px, ширина 1400px)-->
</body>
```

В JS:

```
canvas = document.getElementById('plot');// Получаем элемент с id="plot" на нашей странице
ctx = canvas.getContext('2d');// «Говорим», что будем работать с 2D графикой
```

```
ctx.fillStyle = 'rgb(0, 0, 0)'; // Устанавливаем черный цвет для отрисовки фигур
ctx.fillRect(0, 0, width, height); // Рисуем квадрат
```

Подробное описание в листинге данной практики

# Поиграем в «Жизнь»: Переменные

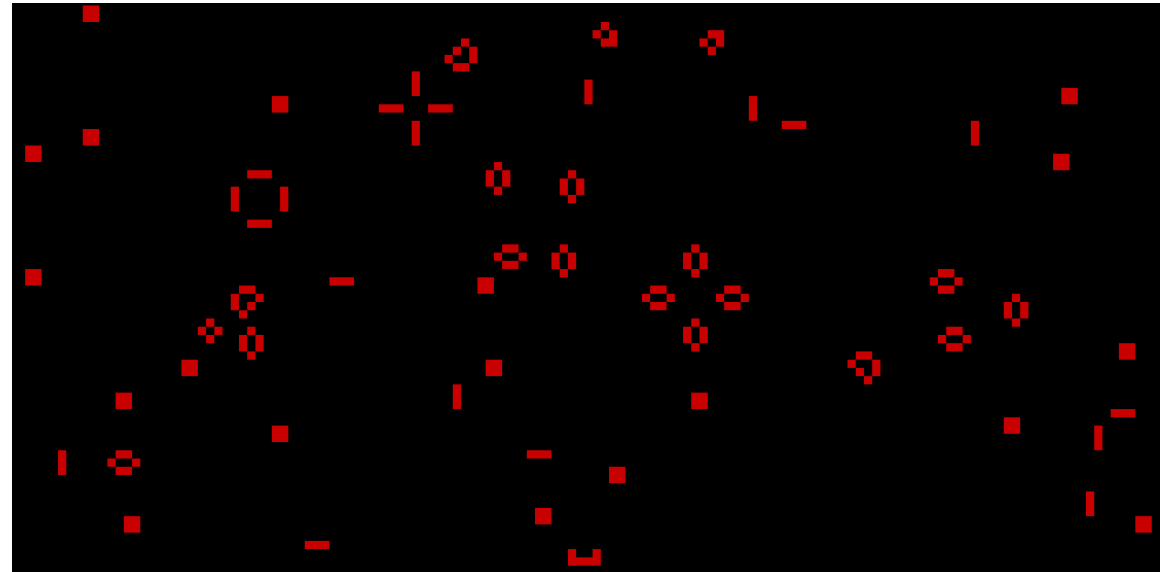
```
var density = 0.5; // Служит для задания первоначальной плотности живых клеток
var resolution = 10; // Сколько пикселей будет занимать одна клетка
var width = 1400; // Ширина поля пикселей
var height = 700; // Высота поля пикселей
var simSpeed = 150; // Скорость симуляции
var plot; // Переменная для хранения текущего состояния клеток
var inTimePlot; // Переменная для хранения промежуточного состояния клеток
```

# Поиграем в «Жизнь»: Функции

```
function init() {  
    // В данной функции реализуется то, что делается программой один раз  
    // Т.е. подготовительная часть для последующей симуляции "Жизни"  
}  
  
function startSimulation() {  
    // В данной функции реализуется циклическая часть программы  
}  
  
function main() {  
    // Из данной функции происходит вызов всех остальных функций  
    // Сама функция main() вызывается браузером после загрузки страницы  
    // благодаря <body onload="main()">  
}
```

# Поиграем в «Жизнь»: Итоги

- Как можно улучшить код?
- Что еще можно вывести на экран?
- Добавить новые правила???



# Функции в JAVA SCRIPT

```
function name(params) {  
    /* Имя -- обязательная уникальная составляющая,  
       задается как и для переменных, по тем же правилам*/  
  
    /* Параметры -- не обязательная часть, служит для получения информации*/  
  
    /*Тело функции -- не обязательная часть, но если ее нет,  
       стоит подумать, а зачем мы все это делаем?!) */  
  
    return // Служит для возвращения значений и завершения функции, не обязательная часть  
}
```



# Полезные ссылки

[Джон Хортон Конвей: Жизнь, как игра](#)

[Функции](#)

[Рекурсия и стек](#)

[<canvas>](#)

[conwaylife.com](#)

[Научно-популярный журнал "Квант"](#)

# Джон Хортон Конве́й

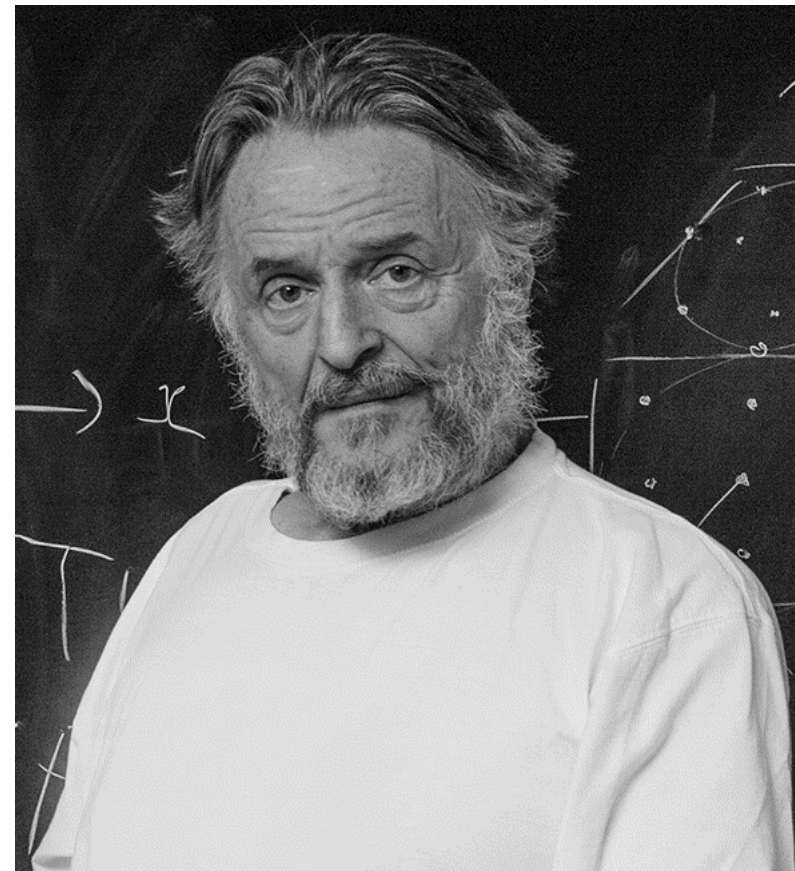
26 декабря 1937

11 апреля 2020

Известен в первую очередь как создатель игры «Жизнь». Однако его вклад в математику весьма многообразен и значителен. В теории групп он открыл группы Конвея и сформулировал гипотезу чудовищного вздора.

Совместно с соавторами заложил основы комбинаторной теории игр, попутно открыв surreальные числа.

Также внёс вклад в теорию узлов, теорию чисел. Многие работы Конвея лежат в области занимательной математики или близки к ней.



Источник:  
<https://habr.com/ru/post/274081/>

Благодарю Вас за внимание!

Амельченко Евгений Олегович  
evgenameo@gmail.com  
2021

Использование данной презентации не может осуществляться без  
соблюдения требований закона РФ об авторском праве и  
интеллектуальной собственности