

# Canvas API

Canvas API (Холст API) предоставляет средства для рисования графики с помощью JavaScript и элемента HTML `<canvas>`. Помимо прочего, его можно использовать для анимации, игровой графики, визуализации данных, манипулирования фотографиями и обработки видео в реальном времени.

Canvas API в основном ориентирован на 2D-графику. API WebGL, который также использует этот `<canvas>` элемент, рисует 2D- и 3D-графику с аппаратным ускорением.

Базовый пример Этот простой пример рисует зеленый прямоугольник на холсте.

HTML [HTML Играть](#) [Скопировать в буфер обмена](#)

`<canvas id=«canvas»></canvas>` JavaScript Метод `Document.getElementById()` получает ссылку на `<canvas>` элемент HTML. Затем `HTMLCanvasElement.getContext()` метод получает контекст этого элемента — объект, на котором будет отображаться рисунок.

Фактический рисунок выполняется с использованием `CanvasRenderingContext2D` интерфейса. Свойство `fillStyle` делает прямоугольник зеленым. Метод `fillRect()` помещает верхний левый угол в точку (10, 10) и присваивает ему размер 150 единиц в ширину и 100 в высоту.

JS [Играть](#) [Скопировать в буфер обмена](#)

```
const canvas = document.getElementById(«canvas»); const ctx = canvas.getContext(«2d»);
```

```
ctx.fillStyle = «green»; ctx.fillRect(10, 10, 150, 100);
```

 Результат [Играть](#)

Ссылка `HTMLCanvasElement` `CanvasRenderingContext2D` `CanvasGradient` `CanvasPattern` `ImageBitmap` `ImageData` `TextMetrics` `OffscreenCanvas` `Path2D` Экспериментальный `ImageBitmapRenderingContext` Экспериментальный [Примечание](#). Интерфейсы, связанные с ними `WebGLRenderingContext`, указаны в разделе `WebGL`.

Примечание: `OffscreenCanvas` также доступно в веб-воркерах.

`CanvasCaptureMediaStreamTrack` это связанный интерфейс.

Руководства и учебные пособия Учебное пособие по холсту Подробное руководство, охватывающее как базовое использование Canvas API, так и его расширенные функции.

Глубокое погружение в HTML5 Canvas Практическое введение в Canvas API и WebGL длиной в книгу.

Справочник по холсту Удобный справочник по Canvas API.

Манипулирование видео с помощью холста Объединение `<video>` и `<canvas>` манипулирование видеоданными в реальном времени.

Библиотеки Canvas API чрезвычайно мощный, но не всегда простой в использовании. Перечисленные ниже библиотеки могут ускорить и упростить создание проектов на основе холста.

EaselJS — это библиотека холстов с открытым исходным кодом, которая упрощает создание игр, генеративного искусства и других высокографических задач. Fabric.js — это библиотека холстов с открытым исходным кодом и возможностями анализа SVG. Heatmap.js — это библиотека с открытым исходным кодом для создания тепловых карт данных на основе холста. JavaScript InfoVis Toolkit создает интерактивную визуализацию данных. Konva.js — это библиотека 2D-холста для настольных и мобильных приложений. p5.js имеет полный набор функций рисования на холсте для художников, дизайнеров, преподавателей и новичков. Paper.js — это среда разработки сценариев векторной графики с открытым исходным кодом, работающая поверх HTML Canvas. Phaser — это быстрая, бесплатная и увлекательная платформа с открытым исходным кодом для браузерных игр на основе Canvas и WebGL. Pts.js — это библиотека для творческого кодирования и визуализации в Canvas и SVG. Rekapі — это API создания ключевых кадров анимации для Canvas. Scrawl-canvas — это библиотека JavaScript с открытым исходным кодом для создания и управления 2D-элементами холста. Платформа ZIM предоставляет удобства, компоненты и элементы управления для творческого кодирования на холсте, включая специальные возможности и сотни красочных руководств. Sprig — это удобная для начинающих библиотека разработки игр на основе плиток с открытым исходным кодом, использующая Canvas. Примечание. См. API WebGL для 2D- и 3D-библиотек, использующих WebGL.

From:  
<http://timerus.ru/> - **book51.ru**

Permanent link:  
[http://timerus.ru/doku.php?id=software:development:web:docs:web:api:canvas\\_api](http://timerus.ru/doku.php?id=software:development:web:docs:web:api:canvas_api)

Last update: **2023/08/22 21:03**

