

# HTML5 Canvas (холст)

## Низкоуровневая графика в веб

- Canvas (холст) это путь к графике в веб
  - Используется JavaScript для вывода графики
  - Включает высококачественную графику
- Canvas (холст) является частью спецификации HTML5
  - Поддерживается большинством браузеров
    - Как для ПК, так и для мобильных устройств

## Использование холста

- Холст это прямоугольная таблица
  - Все присвоение происходит внутри этой таблицы
- Холст включает
  - `<canvas>` HTML элемент
  - JavaScript API для рисования

```
<canvas id="the-canvas" width="100" height="100" />

var canvas = document.getElementById('the-canvas');
var canvasCtx = canvas.getContext('2d');
canvasCtx.fillRect(10, 10, 30, 30);
```

- Установить ширину и высоту

## HTML5 контекст холста

- Холст предоставляет множество способов для рисования
  - Плоское 2d рисование
  - 3d рисование
  - WebGL рисование
- HTML5 предоставляет API для всех этих типов рисования
  - Способ использовать специфический API в соответствии с необходимым контекстом

## Рисование фигур - Прямые, Дуги, Окружности

### Фигуры

- Холст позволяет отображать любые виды фигур
  - Прямоугольники, дуги, окружности, линии
- Каждая из этих фигур может:
  - Быть закрашенной сплошным цветом (заливка)
  - Отрисованной только по границам

```
var canvas = document.getElementById('the-canvas');
var canvasCtx = canvas.getContext('2d');
canvasCtx.fillRect(10, 10, 25, 25);
canvasCtx.strokeRect(10, 10, 25, 25);
```

## Рисование прямоугольников

- Рисование прямоугольников - это самый простой способ рисования в canvas
  - Встроенный функционал
  - `context.fillRect (x, y, width, height)`
    - Создает прямоугольную фигуру в позиции (x, y) от левого верхнего края холста
    - Фигура рисуется полноцветной
  - `context.strokeRect (x, y, width, height)`
    - То же что и `fillRect`

- Отрисовываются только границы
- Нарисовать прямоугольник залить светло-синим с темно-синими границами

```
<pre><code><canvas id="rects-canvas"> </canvas></code></pre>
<pre><code>var canvas = document.getElementById('rects-canvas'),
ctx = canvas.getContext('2d');
ctx.fillStyle = 'rgb(107, 187, 201)';
ctx.strokeStyle = 'rgb(2, 55, 155)';
ctx.fillRect(20, 20, 140, 90);
ctx.strokeRect(20, 20, 140, 90);</code></pre>
```

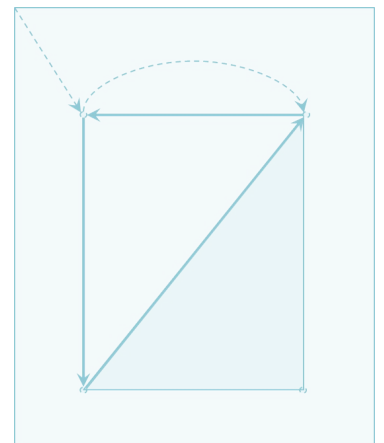
## Рисование дуг и линий

- В холсте могут рисоваться фигуры более сложные чем прямоугольники
  - Кривые Безье, эллипсы, дуги
  - Могут вырисовываться траектории отображения точек
- Путь это просто последовательность точек
  - В зависимости от используемого метода, точки могут соединяться между собой с использованием прямых линий или кривых
- Контекст холста содержит несколько методов для рисования путей:
  - beginPath()
    - Начало пути
  - moveTo(x, y)
    - Изменение пути у маркера пути
  - lineTo(x, y)
    - Рисование прямой линии от точки расположения маркера до позиции (x, y)
  - fill() / stroke()
    - Закрашивается линию пути

## Как это работает?

- Холст маркирует точки, которые располжены в таблице холста
  - И запоминает как эти точки соединяются
  - Когда вызывается метод fill() или stroke(), все точки соединяются вместе

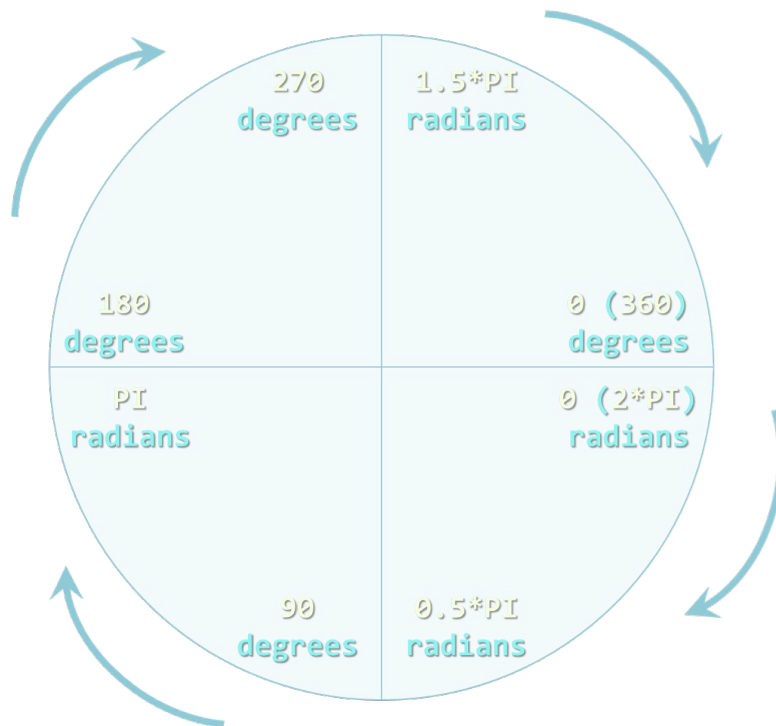
```
ctx.beginPath();
ctx.lineTo(200, 50);
ctx.lineTo(50, 50);
ctx.stroke();
ctx.beginPath();
ctx.moveTo(200, 50);
ctx.lineTo(200,300);
ctx.lineTo(50, 300);
ctx.closePath();
ctx.fill();
```



## Рисование окружностей

- Холст имеет методы для рисования окружностей
- arc(x, y, радиус, от, до, counterclockwise)
- Рисует окружность с центром в точке (x,y) со стартовой позиции до конечной
- Позиции в окружностях определяются в радианах (градусах)
- Формула перевода градусов в радианы следующая:
  - radians = degrees \* PI/180

## Градусы и радианы



- Инициализация пути рисования:

```
ctx.beginPath();
```

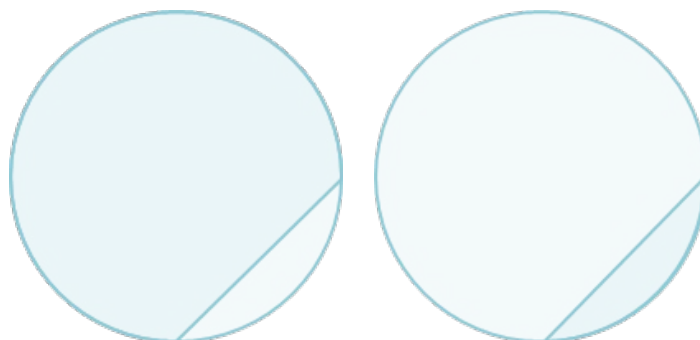
- Рисование целой окружности:

```
//По часовой стрелке
ctx.arc(x, y, r, 0, 2- Math.PI);
//Против часовой стрелке
ctx.arc(x, y, r, 2- Math.PI, 0, true);
```

- Рисование сегмента окружности:

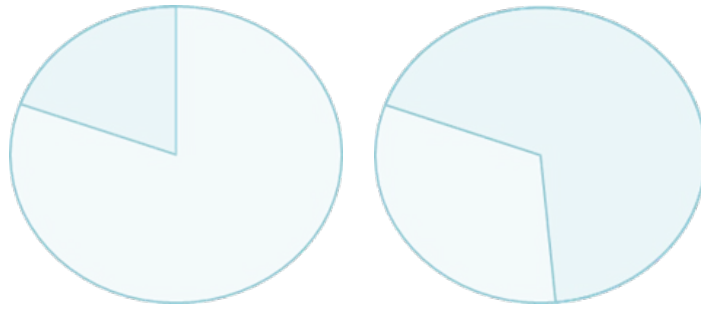
```
//Меньшую часть (по часовой стрелке)
ctx.arc(x, y, r, Math.PI/2, 2- Math.PI);

//Большую часть (против часовой стрелки)
ctx.arc(x, y, r, Math.PI/2, 2- Math.PI, true);
```



## Рисование секторов окружностей

- Сектор окружности это часть дуги окружности замыкающаяся двумя радиусами
- Холст имеет несколько методов для создания секторов окружностей
  - Тем не менее холст так же поддерживает рисование путями
- Сектор окружности может быть получен созданием сектора и линиями из центра окружности к его крайним точкам



- `context.closePath()` соединяет начальную и конечную точки пути

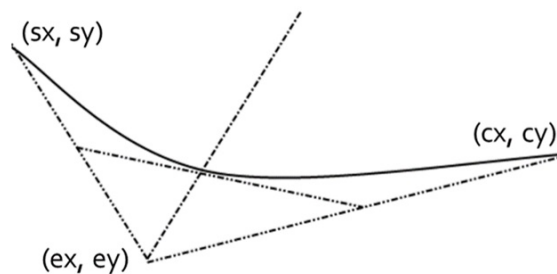
```
function drawSector(x, y, r, from, to, isCounterClockwise) {
  ctx.beginPath();
  ctx.arc(x, y, r, from, to, isCounterClockwise);
  ctx.lineTo(x, y);
  ctx.closePath();
  ctx.stroke();
}
```

## Рисование кривых

- Холст поддерживает два способа рисования кривых
  - **Квадратичные** кривые
    - Простое рисование кривой основанное на контрольной точке
  - Кривые **Безье**
    - A more complex curve based on two control points
    - Более комплексный подход, основанный на двух контрольных точках
- Оба типа кривых могут быть получены с помощью рисованием путей

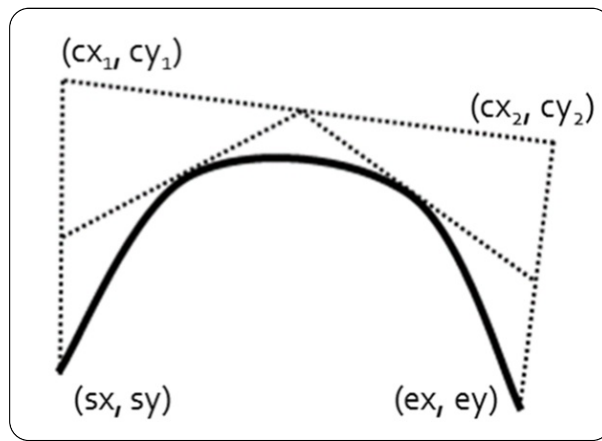
## Квадратичные кривые

- Квадратичные кривые являются базовыми кривыми
  - Применяются **две точки контекста** и **контрольная точка**
    - The first is the last point from the path (sx, sy)
    - The second is the one from the curve (cx, cy)
  - `context.quadraticCurveTo(cx, cy, ex, ey)`



## Кривые Безье

- Кривые Безье подобны квадратичным кривым, только они имеют **две точки контекста** и **две контрольные точки**



## Рисование текста

- HTML5 холст, так же может **отрисовывать текст**:
  - Методы:
    - `context.fillText (text, x, y)` — заполнить цветом пространство текста
    - `context.strokeText (text, x, y)` — заполнить цветом только границы шрифта текста
  - Свойства:
    - `context.font` — установить размер и семейство шрифта
    - `context.fillStyle` — цвет заливки шрифта
    - `context.strokeStyle` — цвет границ шрифта
- Вывести текст 'Курсы Веб программирования'
  - Заполнить **желто-зеленым цветом**, с границами шрифта **темно-зеленого цвета**
  - Семейство шрифтом — Arial
  - Размер шрифта от 28px до 48px

```
var minFontSize = '28';
var currentFontSize = 48;
while (minFontSize <= currentFontSize) {
  ctx.font = currentFontSize + 'px ' + 'Arial';
  ctx.fillText(text, x, y);
  ctx.strokeText(text, x, y);
  y += currentFontSize + offset;
  currentFontSize -= 4;
}
```

Курс веб программирования  
 Курс веб программирования  
 Курс веб программирования  
 Курс веб программирования  
 Курс веб программирования  
 Курс веб программирования

## Стили холста

- Холст поддерживает два стиля
  - Стили для **заливки** и **контура**
    - Может быть как сплошным цветом, так и паттерном
  - Стили для **отображения контура**
    - Прерывистая и сплошная

## Преобразования в холстах

- Объект холста можно трансформировать
  - т.е может быть повернутым, масштабированным или преобразованным
- `context.scale(dx, dy)` — все координаты и точки масштабируются
  - `fillRect(X, X, W, H)` будет отрисован прямоугольник
  - до позиции (**dx X, dy Y**)
  - С шириной (**dx\* W**) и высотой (**dy\* H**)
- `context.rotate(D)` — все отрисованное будет повернуто на угол **D градусов**

## Попиксельные манипуляции

- Холст поддерживает **попиксельную манипуляцию**
  - Над каждым пикселем может производиться **индивидуальная манипуляция**
- Используется `context.getImageData(x, y, w, h)`
  - Возвращает данные отрисованного объекта
    - Данные по прямоугольнику содержат отступ от **верхнего левого угла на (x, y)** с шириной **w** и высотой **h**
  - Данные изображения содержат **массив чисел** являющихся набором всех пикселей

## Попиксельные манипуляции (2)

- массив точек содержит значения в диапазоне от 0 до 255
  - Каждое значение определяет **цветовую компоненту** в пространстве **RGBA**
  - Пиксели сгруппированы в триплеты
  - Значения цвета для **i-ого пикселя** находятся в позиции:
    - `pixels[i]` содержит **RED** компоненту
    - `pixels[i+1]` содержит **GREEN** компоненту
    - `pixels[i+2]` содержит **BLUE** компоненту
- Инвертировать все цвета на холсте
  - Изменить значение каждого цвета сс на 255-сс

```
var i,
    len,
    width = canvas.width,
    height = canvas.height,
    imageData = ctx.getImageData(150, 150, width, height),
    data = imageData.data;
for(i = 0, len = data.length; i < len; i+=4){
    data[i+1] = 255 - data[i+1];
    data[i] = 255 - data[i];
    data[i+2] = 255 - data[i+2];
}
ctx.putImageData(imageData, 0, 0);
```