

Google Maps / Рисование / Анимация



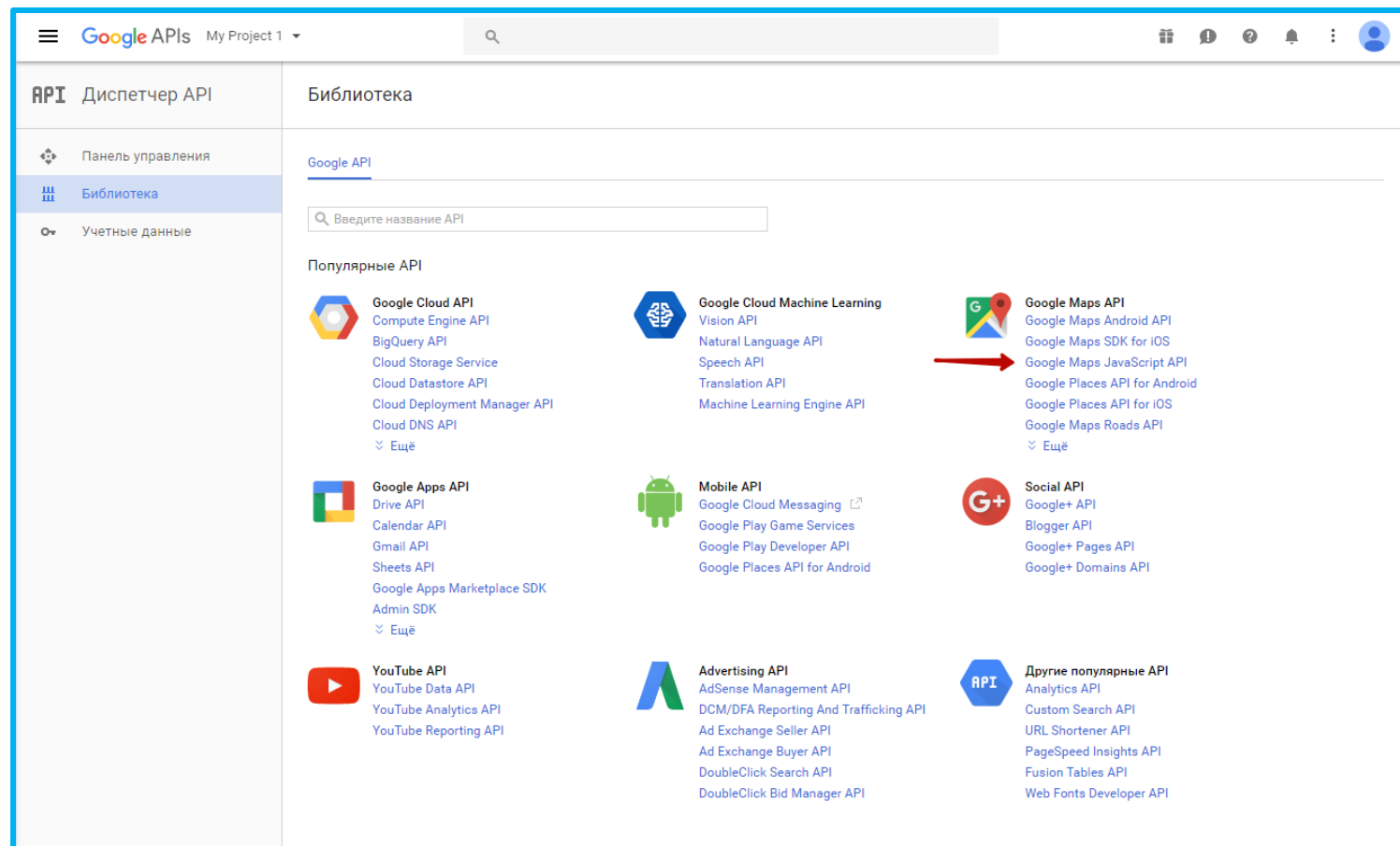
JavaScript
Courses

courses.dp.ua

Google Maps – самый популярный
сервис.

Но чтобы им
пользоваться нужно
иметь ключ...

Консоль управления API Google'a



<https://console.developers.google.com/>

Ключ для использования сервисов Google

Для использования API сервисов компании Google необходимо зарегистрироваться и получить ключ, для использования в своём коде.

AIzaSyASjcSgdcXxxFTy3uVsSBWw0N-Pdve736A

Пример ключа

Руководство по работе с Google Maps

Руководство по работе с Google Maps

The screenshot shows the Google Maps APIs documentation page for the JavaScript API. The page is in Russian and features a blue header with navigation links: Главная, Документация, and Цены и планы. A search bar and a user profile icon are also present. Below the header, the breadcrumb trail reads 'Веб > Maps JavaScript API'. Two buttons are visible: 'ПОЛУЧИТЬ КЛЮЧ' and 'ПРОСМОТРЕТЬ ЦЕНЫ И ПЛАНЫ'. A secondary navigation bar includes links for ОБЗОР, РУКОВОДСТВА (which is highlighted), ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ, ПРИМЕРЫ, ПОДДЕРЖКА, and ОТПРАВИТЬ ОТЗЫВ. The main content area is titled 'Начало работы' (Getting Started) and includes a star rating. It contains sections for 'Целевая аудитория' (Target Audience) and 'Приложение Hello World' (Hello World Application). The 'Целевая аудитория' section explains that the document is for those familiar with JavaScript and Google Maps, and mentions the availability of 'учебных материалов по JavaScript' (JavaScript learning materials). The 'Приложение Hello World' section introduces a simple example of using the API to display a map of Sydney. On the left, a sidebar lists various topics under 'Начало работы' and 'Учебные материалы'. On the right, a 'Содержание' (Table of Contents) section lists the document's structure, including 'Целевая аудитория', 'Приложение Hello World', 'Декларирование приложения в формате HTML5', 'Загрузка Google Maps JavaScript API', 'HTTPS или HTTP', 'Библиотеки', 'Синхронная загрузка API', 'Элементы DOM карты', 'Параметры карты', 'Уровни масштабирования', 'Объект Map', and 'Устранение неполадок'.

Google Maps APIs Главная Документация Цены и планы Поиск ВСЕ ПРОДУКТЫ

Веб > Maps JavaScript API ПОЛУЧИТЬ КЛЮЧ ПРОСМОТРЕТЬ ЦЕНЫ И ПЛАНЫ

ОБЗОР РУКОВОДСТВА ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ ПРИМЕРЫ ПОДДЕРЖКА ОТПРАВИТЬ ОТЗЫВ

Начало работы ☆☆☆☆☆

Целевая аудитория

Данный документ предназначен для тех, кто знаком с языком программирования JavaScript и понятиями объектно-ориентированного программирования. Также требуется знакомство с Google Maps с точки зрения пользователя. В Интернете доступно множество учебных материалов по JavaScript.

Этот концептуальный документ поможет вам быстро приступить к проектированию и разработке приложений с использованием Google Maps JavaScript API. Также мы публикуем Дополнительные материалы по Google Maps JavaScript API.

Приложение Hello World

Проще всего познакомиться с Google Maps JavaScript API на простом примере. На следующей веб-странице показана карта, в центре которой располагается город Сидней (штат Новый Южный Уэльс, Австралия):

Содержание

- Целевая аудитория
- Приложение Hello World
- Декларирование приложения в формате HTML5
- Загрузка Google Maps JavaScript API
- HTTPS или HTTP
- Библиотеки
- Синхронная загрузка API
- Элементы DOM карты
- Параметры карты
- Уровни масштабирования
- Объект Map
- Устранение неполадок

<https://developers.google.com/maps/documentation/javascript/tutorial?hl=ru>

Google Maps в теории

Чем нам могут помочь карты от Google

1. *Размещение карты на странице (выбор отображаемого места, масштаба, типа карты и т.п.);*
2. *Установка маркеров (со своими иконками) на карте;*
3. *Размещение информационных окон (и обработка событий на карте);*
4. *Расчёт расстояния между точками;*
5. *Рисование на карте;*
6. *Построение маршрута между двумя точками (на основе google навигатора);*

Координаты



Широта = Latitude

Долгота = Longitude

Для задания координат в Google Maps используется специальный объект, который можно создать при помощи конструктора:

```
var point = new google.maps.LatLng(48.47, 35.05);
```

В некоторых случаях хватит и подобного:

```
var point = { lat: 48.47, lng: 35.05 };
```

1. Размещение карты на странице

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Simple Map</title>
5   <meta name="viewport" content="initial-scale=1.0">
6   <meta charset="utf-8">
7   <style>
8     #map {
9       height: 100%;
10    }
11    html, body {
12      height: 100%;
13      margin: 0;
14      padding: 0;
15    }
16  </style>
17 </head>
18 <body>
19   <div id="map"></div>
20   <script>
21
22     var map;
23     function initMap() {
24       map = new google.maps.Map(document.getElementById('map'), {
25         center: {lat: 48.418162, lng: 35.043517},
26         zoom: 12,
27       });
28     }
29
30   </script>
31   <script src=
32     "https://maps.googleapis.com/maps/api/js?key=AIzaS...:FBK
33     anKczSd9xk&callback=initMap"
34   async defer></script>
35 </body>
36 </html>
```

*Шаблон из руководство по Google Maps.
Ключ подставляется в URL подключаемого JS-файла.*

1а. Настройка отображения карты

```
22     var map;  
23     function initMap() {  
24         map = new google.maps.Map(document.getElementById('map'), {  
25             center: {lat: 48.418162, lng: 35.043517},  
26             zoom: 12,  
27             mapTypeId: 'satellite',  
28             zoomControl: false,  
29             mapTypeControl: false,  
30             scaleControl: false,  
31             streetViewControl: false,  
32             rotateControl: false,  
33             fullscreenControl: false,  
34             draggable: false,  
35             scrollwheel: false,  
36             disableDoubleClickZoom: false  
37         });  
38     }
```

У разработчиков есть возможность тонко настроить то как карта будет выглядеть на странице и какие действия с ней будут доступны пользователю.

<https://developers.google.com/maps/documentation/javascript/controls?hl=ru>

2. Маркеры на карте

```
22 var map;  
23 function initMap() {  
24     map = new google.maps.Map(document.getElementById('map'), {  
25         center: {lat: 48.418162, lng: 35.043517},  
26         zoom: 12  
27     });  
28  
29     var airport = new google.maps.Marker({  
30         position: { lat: 48.368, lng: 35.0938 },  
31         title: "Аэропорт",  
32         icon: "http://files.courses.dp.ua/js/16/plane-ico.png"  
33     });  
34  
35     airport.setMap(map);  
36 }
```

Маркеры – метки на карте, а в коде - объекты, которые привязываются к карте.

<https://developers.google.com/maps/documentation/javascript/markers?hl=ru>



./source/images/plane-ico.png

3. Информационные окна

```
22  var map;  
23  function initMap() {  
24      map = new google.maps.Map(document.getElementById('map'), {  
25          center: {lat: 48.418162, lng: 35.043517},  
26          zoom: 12  
27      });  
28  
29      var airport = new google.maps.Marker({  
30          position: { lat: 48.368, lng: 35.0938 },  
31          title: "Аэропорт",  
32          icon: "http://files.courses.dp.ua/js/16/plane-ico.png"  
33      });  
34  
35      airport.setMap(map);  
36  
37      var info = new google.maps.InfoWindow({  
38          content: "<h1>DNK</h1><p>Dnipro International Airport</p>"  
39      });  
40  
41      airport.addListener("click", function() {  
42          info.open(map, airport);  
43      });  
44  }  
45
```

Информационные окна – фрагмент разметки, который может отобразиться на карте, как правило связаны с событиями. События на картах Google как и в целом для JS – возможность связать выполнение функции с действием пользователя.

<https://developers.google.com/maps/documentation/javascript/infowindows>

Один важный момент: геолокация

Геолокация

```
45 navigator.geolocation.getCurrentPosition(function(result) {  
46     console.dir(result);  
47     var user = new google.maps.Marker({  
48         position : {  
49             lat: result.coords.latitude,  
50             lng: result.coords.longitude  
51         },  
52         title: "You are here"  
53     });  
54     user.setMap(map);  
55  
56 }, function(error){  
57     alert("Geolocations fail: " + error.message);  
58 });
```

У браузера есть возможность узнать координаты пользователя на местности. *Но важно проверять поддерживает ли браузер геолокацию!*

https://www.w3schools.com/html/html5_geolocation.asp

4. Расстояние между точками

```
58 <script src=  
    "https://maps.googleapis.com/maps/api/js?key=AIzaSyASjcSgdcXxx  
    SBWw0N-Pdve736A&callback=initMap&libraries=geometry" ←  
59 async defer></script>
```

```
45 navigator.geolocation.getCurrentPosition(function(result) {  
46     console.dir(result);  
47     var user = new google.maps.Marker({  
48         position : {  
49             lat: result.coords.latitude,  
50             lng: result.coords.longitude  
51         },  
52         title: "You are here!"  
53     });  
54     user.setMap(map);  
55  
56     var distance = google.maps.geometry.spherical.computeDistanceBetween(airport.position, user.position);  
57     user.title += " airport in " + (distance/1000).toFixed(2) + " km";  
58  
59 }, function(error) {  
60     alert("Geolocations fail: " + error.message);  
61 });  
62 }
```

Google Maps даёт возможность узнать расстояние между двумя точками (в метрах). Необходимо подключить модуль (библиотеку) **geometry**.

<https://developers.google.com/maps/documentation/javascript/geometry>

5. Построение маршрута

```
56 var distance = google.maps.geometry.spherical.computeDistanceBetween(airport.position, user.position);
57 user.title += " airport in " + (distance/1000).toFixed(2) + " km";
58
59 var road_drawer = new google.maps.DirectionsRenderer({
60   map: map
61 });
62 var road_calc = new google.maps.DirectionsService();
63
64 road_calc.route({
65   origin: user.position,
66   destination: airport.position,
67   travelMode: "DRIVING"
68 }, function(result) {
69   alert("Distance : " + result.routes[0].legs[0].distance.value/1000 + " km");
70   road_drawer.setDirections(result);
71 });
```

Построение маршрута состоит в перечислении точек через которые должен пройти маршрут, отправить их на обработку серверу, отрисовать полученный результат. **Это API необходимо дополнительно включить в панели управления сервисами Google.**

<https://developers.google.com/maps/documentation/javascript/directions?hl=ru>

Рисование/Canvas

JavaScript и рисование



Воспользуйтесь заготовкой: [./source/ex01.html](#)

Рисование, Графика, Canvas

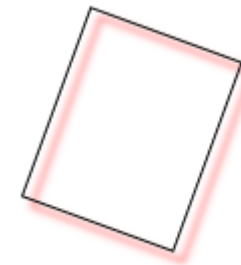
```
31 <canvas id="paint-canvas"></canvas>
32 <script>
33
34     var canvas      = document.getElementById("paint-canvas");
35     canvas.width    = canvas.clientWidth;
36     canvas.height   = canvas.clientHeight;
37
38     var context     = canvas.getContext("2d");
39
40     context.moveTo(200, 200);
41     context.lineTo(300, 250);
42     context.lineTo(200, 300);
43     context.closePath();
44     context.stroke();
45
46 </script>
```

*Тег **canvas** – представляет собой «холст», прямоугольную область в которой можно рисовать. Контекст canvas'a – объект который содержит множество методов для рисования на «холсте».*

Рисование, Графика, Canvas

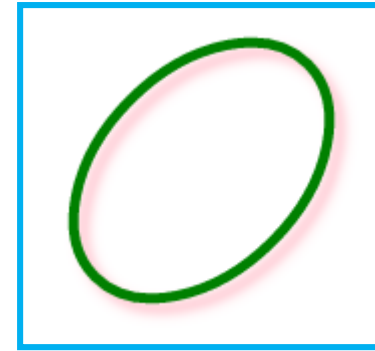
*Рисование на **canvas**'е основано на отрисовке примитивов.*

- 1) Штриховых (контурных фигур) – в названии методов и свойств есть слово **stroke**;*
- 2) Заполненных фигур, в названии методов и свойств есть слово **fill**;*
- 3) Наложении спецэффектов (тени, развороты, искажения и т.п.).*

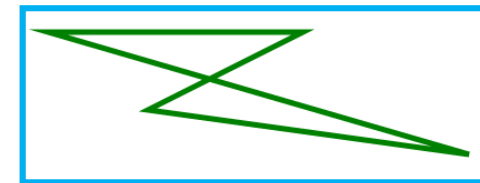
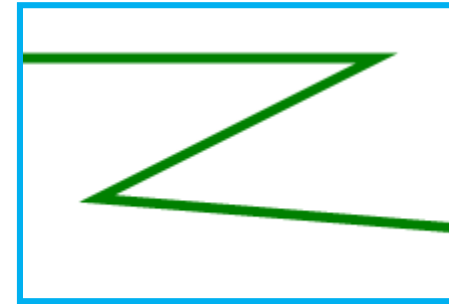


Рисование, Графика, Canvas

Примитивы можно рисовать при помощи функций-заготовок: прямоугольник (**rect()**), эллипс (**ellipse()**) и т.п.



Либо самостоятельно задав контур фигуры состоящей из множества линий. Для этого есть функции **beginPath()** и **closePath()** – для случаев когда нужно замкнуть контур (между первой и последней точкой фигуры).



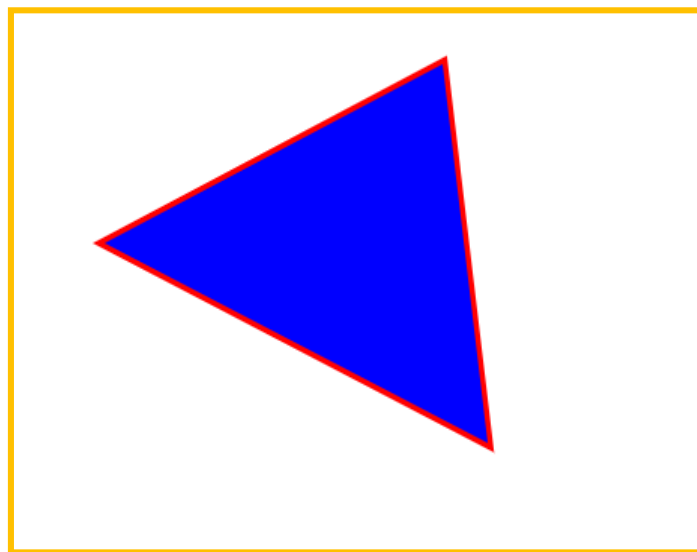
Рисование примитивов

```
context.beginPath();  
context.moveTo(175, 225);  
context.lineTo(400, 113);  
context.lineTo(430, 350);  
//context.closePath();  
context.stroke();
```

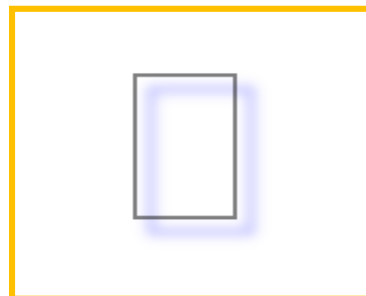


```
context.beginPath();  
context.moveTo(175, 225);  
context.lineTo(400, 113);  
context.lineTo(430, 350);  
//context.closePath();  
context.fill();
```

```
context.beginPath();  
context.moveTo(175, 225);  
context.lineTo(400, 113);  
context.lineTo(430, 350);  
context.closePath();  
context.lineWidth = 7;  
context.strokeStyle = "red";  
context.fillStyle = "blue";  
context.stroke();  
context.fill();
```



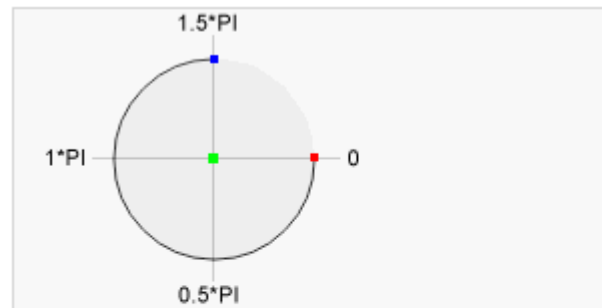
```
context.rect(300, 200, 50, 80);  
context.shadowBlur = 10;  
context.shadowOffsetY = 8;  
context.shadowOffsetX = 8;  
context.shadowColor = "blue";  
context.stroke();
```



Рисование примитивов

```
context.strokeStyle = "green";  
context.rect(300, 200, 50, 80);  
//rect(x,y, width, height)  
context.stroke();
```

```
context.arc(200,300,45, 1.8*Math.PI, 2*Math.PI, false);  
//arc(x, y, radius, startAngle, finishAngle, direction);  
context.stroke();
```



Center	arc(100 , 75 ,50,0*Math.PI,1.5*Math.PI)
Start angle	arc(100,75,50, 0 ,1.5*Math.PI)
End angle	arc(100,75,50,0*Math.PI, 1.5*Math.PI)

Прямоугольник и дуга

Свойства (графические атрибуты «холста»)

Paths

Method	Description
fill()	Fills the current drawing (path)
stroke()	Actually draws the path you have defined
beginPath()	Begins a path, or resets the current path
moveTo()	Moves the path to the specified point in the canvas, without creating a line
closePath()	Creates a path from the current point back to the starting point
lineTo()	Adds a new point and creates a line to that point from the last specified point in the canvas
clip()	Clips a region of any shape and size from the original canvas
quadraticCurveTo()	Creates a quadratic Bézier curve
bezierCurveTo()	Creates a cubic Bézier curve
arc()	Creates an arc/curve (used to create circles, or parts of circles)
arcTo()	Creates an arc/curve between two tangents
isPointInPath()	Returns true if the specified point is in the current path, otherwise false

Transformations

Method	Description
scale()	Scales the current drawing bigger or smaller
rotate()	Rotates the current drawing
translate()	Remaps the (0,0) position on the canvas
transform()	Replaces the current transformation matrix for the drawing

http://www.w3schools.com/tags/ref_canvas.asp

Paint на JavaScript

Paint на JavaScript

```
31 <canvas id="paint-canvas"></canvas>
32 <script>
33
34     var canvas      = document.getElementById("paint-canvas");
35     canvas.width    = canvas.clientWidth;
36     canvas.height   = canvas.clientHeight;
37
38     var context      = canvas.getContext("2d");
39     context.lineJoin = "round";
40     context.lineWidth = 10;
41     context.strokeStyle = "red";
42
43     var x;
44     var y;
45     var painting = false;
46
47     canvas.onmousedown = function(e) {
48         painting = true;
49         x = e.offsetX;
50         y = e.offsetY;
51         context.moveTo(x, y);
52     };
53
54     canvas.onmousemove = function(e) {
55         if(painting) {
56             context.lineTo(e.offsetX, e.offsetY);
57             context.closePath();
58             context.stroke();
59             x = e.offsetX;
60             y = e.offsetY;
61             context.moveTo(x, y);
62         }
63     };
64
65     canvas.onmouseup = canvas.onmouseleave = function(e) {
66         painting = false;
67     };
68
69 </script>
```

Простой paint на базе JavaScript и canvas.

Отрисовка изображений

Загрузка изображений на холст

```
var img1 = new Image();  
img1.src = "http://js.courses.dp.ua/files/15/1.jpg";  
img1.onload = function(){  
    context.drawImage(img1, 100, 100);  
}  
  
var img2 = new Image();  
img2.src = "http://js.courses.dp.ua/files/15/2.jpg";  
img2.onload = function(){  
    context.drawImage(img2, 50, 50, 250, 185);  
}
```



Загрузка изображений на холст / метод `drawImage()`

Метод `.drawImage()` выводит ранее загруженное изображение на «холст»

`drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)`

image — объект `Image()`;

sx, sy — координаты верхнего левого вставляемого изображения;

sWidth, sHeight — размеры вставляемого изображения;

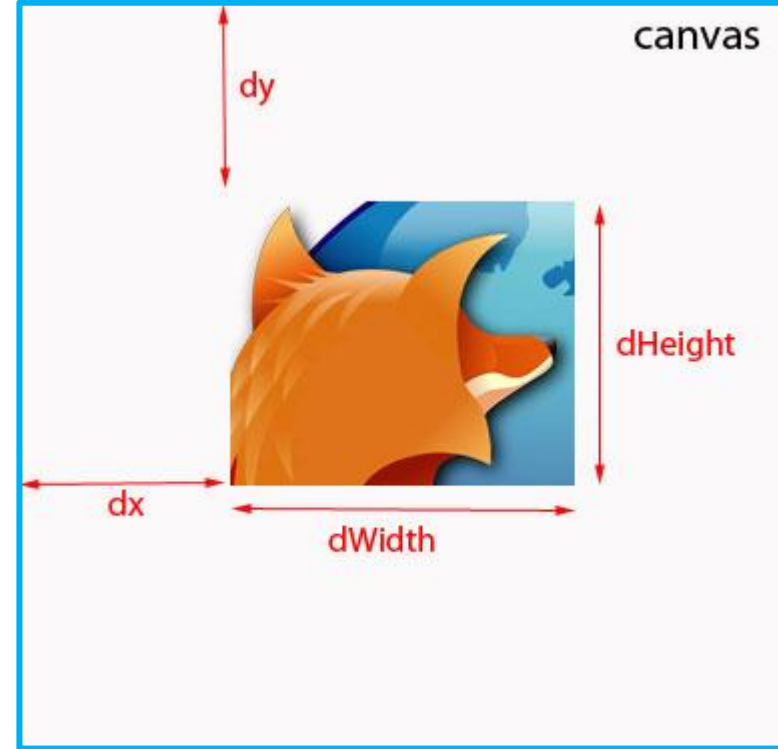
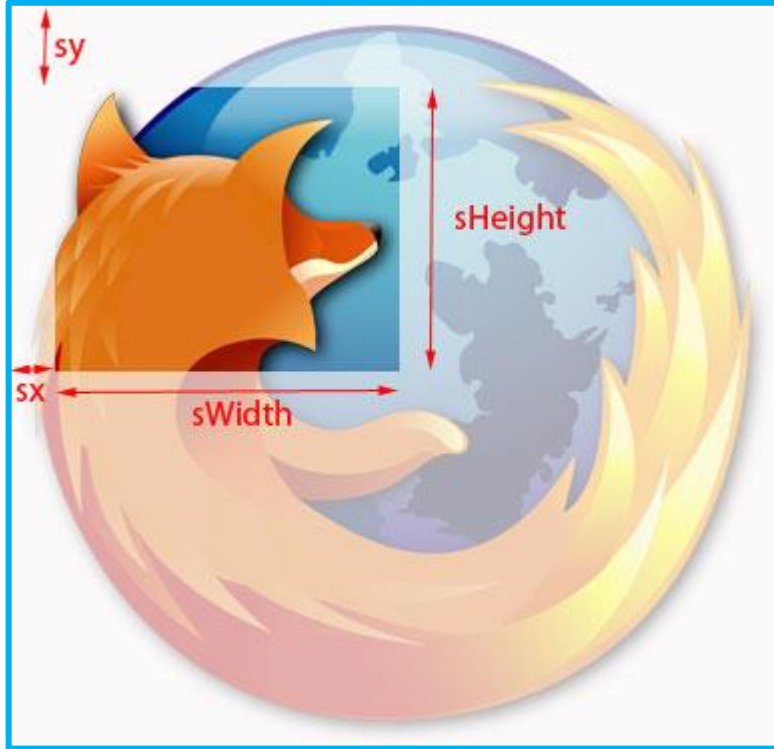
dx, dy — координаты размещения верхнего левого угла обрезанного изображения на холсте;

dWidth, dHeight — размеры обрезанного изображения на холсте.

<https://developer.mozilla.org/ru/docs/Web/API/CanvasRenderingContext2D/drawImage>

Загрузка изображений на холст / метод `drawImage()`

Метод `.drawImage()` выводит ранее загруженное изображение на «холст»



`drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)`

Анимация и JavaScript

*Анимация на веб-странице это,
как правило, изменение во времени
того или иного стилевого
свойства элемента во времени.*

Немного практики: краткая история анимации в браузере.

```
1 <html>
2 <head>
3 <style>
4     div {
5         width: 200px;
6         height: 200px;
7         border: 2px solid red;
8         background-color: yellow;
9         position: fixed;
10        left: 20px;
11        top: 20px;
12    }
13 </style>
14 <script>
15     window.onload = function() {
16
17     }
18 </script>
19 </head>
20 <body>
21     <div></div>
22 </body>
23 </html>
```

Воспользуйтесь заготовкой: <./source/ex02.html>

Скопируйте код примера в *notepad++*

Анимация при помощи setInterval()

```
14 <script>
15     window.onload = function() {
16         var offset = 0;
17         var step    = 3;
18
19         setInterval(function() {
20             tag.style.left = offset + "px";
21             tag.style.top  = offset + "px";
22
23             offset += step;
24
25             step = ((offset > 400) || (offset < 1)) ? -step : step;
26
27         }, 30);
28     }
29 </script>
```

Вроде и работает, но «дёргано» и ресурсоёмко.

Анимация при помощи `requestAnimationFrame()`

```
14 <script>
15     window.onload = function() {
16         var offset = 0;
17         var step = 3;
18
19         function move() {
20             requestAnimationFrame(move);
21
22             tag.style.left = offset + "px";
23             tag.style.top = offset + "px";
24
25             offset += step;
26
27             step = ((offset > 400) || (offset < 1)) ? -step : step;
28         }
29
30         move();
31     }
32 </script>
```

Уже лучше, теперь отрисовка происходит в те моменты когда браузер «готов» к перерисовке страницы.

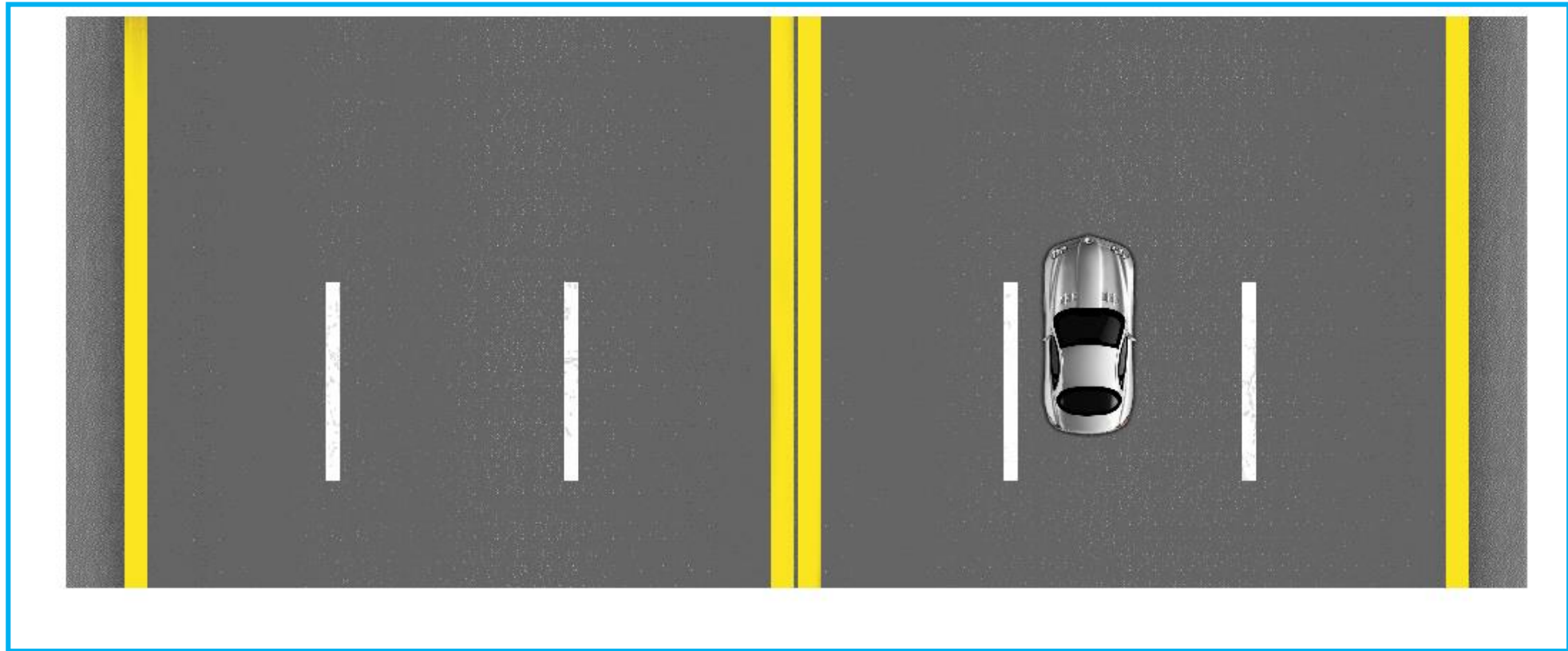
Анимация при помощи `requestAnimationFrame()` и FPS

```
14 <script>
15     window.onload = function() {
16         var offset = 0;
17         var step = 3;
18
19         var fps = 15;
20
21         function move() {
22             setTimeout(function() {
23                 requestAnimationFrame(move);
24             }, 1000/fps);
25
26             tag.style.left = offset + "px";
27             tag.style.top = offset + "px";
28
29             offset += step;
30
31             step = ((offset > 400) || (offset < 1)) ? -step : step;
32         }
33
34         move();
35     }
36 </script>
```

С помощью такой техники мы можем делать анимацию с желаемой частотой кадров.

Механика игровой анимации

Немного практики: механика игровой анимации.



Воспользуйтесь заготовкой: [./source/ex03.html](#)

Немного практики: механика игровой анимации.

```
11 <script>
12     var context      = document.querySelector("canvas").getContext("2d");
13
14     var road_url      = "http://files.courses.dp.ua/js/21/road_2.jpg";
15     var road_image    = new Image();
16     road_image.src    = road_url;
17
18     var car_url       = "http://files.courses.dp.ua/js/21/car.png";
19     var car_image     = new Image();
20     car_image.src     = car_url;
21
22
23     var offset        = 1;
24     var step          = 3;
25     var car_x         = 200;
26     var car_y         = 200;
27
28     function go(){
29         offset += step;
30         if(offset + 511 > 1024) offset = 0;
31         requestAnimationFrame(go);
32         context.drawImage(road_image, 0, 511-offset, 1024, 400, 0, 0, 1024, 400);
33         context.drawImage(car_image, car_x, car_y, 90, 140);
34     }
35     requestAnimationFrame(go);
36
37     document.body.onkeydown = function(e){
38         if(e.code == "ArrowLeft")   car_x -= 4;
39         if(e.code == "ArrowRight")  car_x += 4;
40         if(e.code == "ArrowUp")     car_y -= 4;
41         if(e.code == "ArrowDown")   car_y += 4;
42         if(e.code == "ControlLeft") step = 10;
43         if(e.code == "Space")       step = 1;
44     }
45
46     document.body.onkeyup = function(e){
47         if(e.code == "ControlLeft") step = 2;
48         if(e.code == "Space")       step = 2;
49     }
50
51 </script>
```

Игровая механика сводится к перерисовке на странице изображений-заготовок (спрайтов) на экране в зависимости от «ситуации» в игре.

Домашнее задание
/сделать

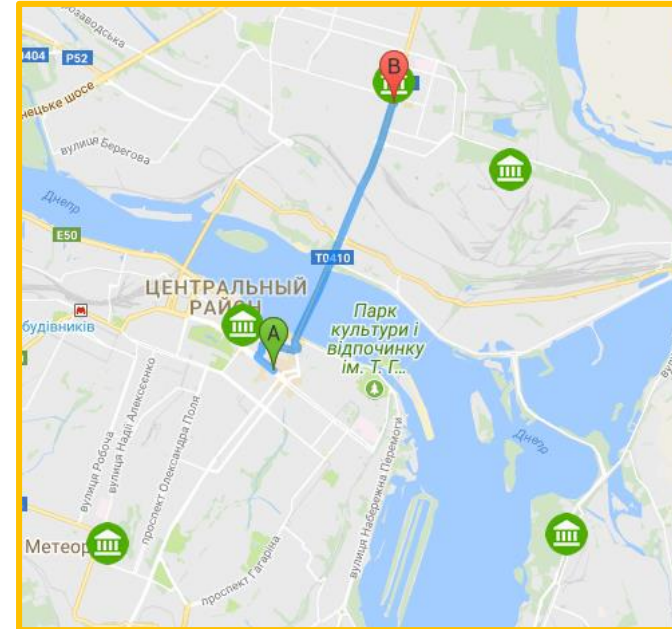
Домашнее задание #К.1

Маркеры и маршруты

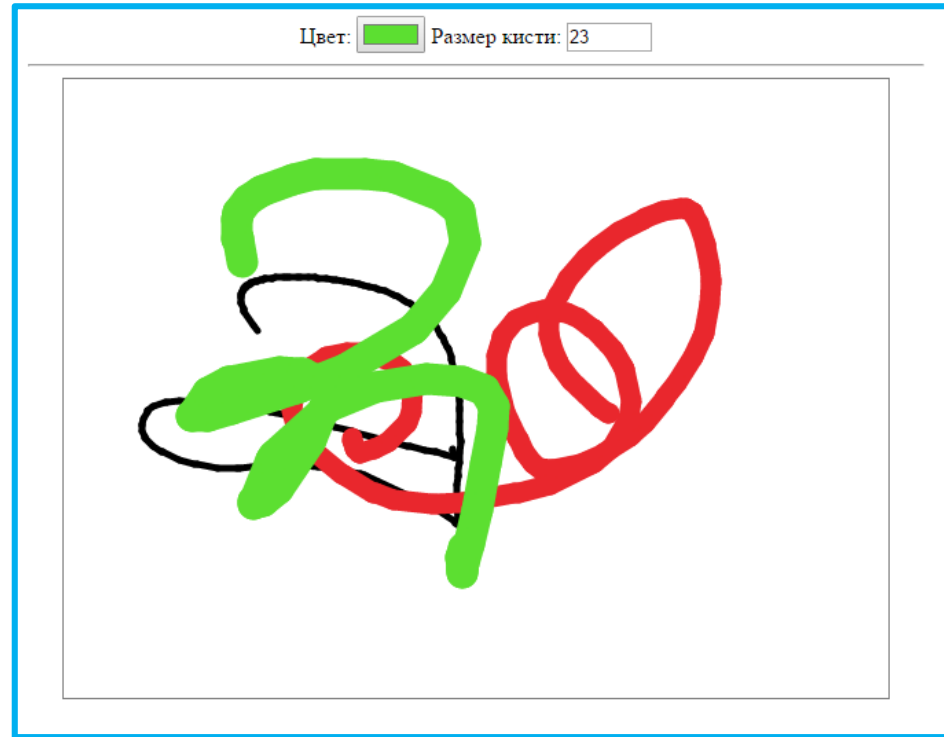
<http://courses.dp.ua/api/branches.php> - JSON список филиалов;
[./source/images/branche-ico.png](#) - иконка филиала.

Наша задача: Получить список отделение в виде JSON где указаны координаты отделений. Нанести отделения на карту, указать положение пользователя на карте. Проложить маршрут между пользователем и ближайшим отделением.

Доп. Сложность: при клике по другому отделению маршрут перестраивается от пользователя к этому другому отделению.

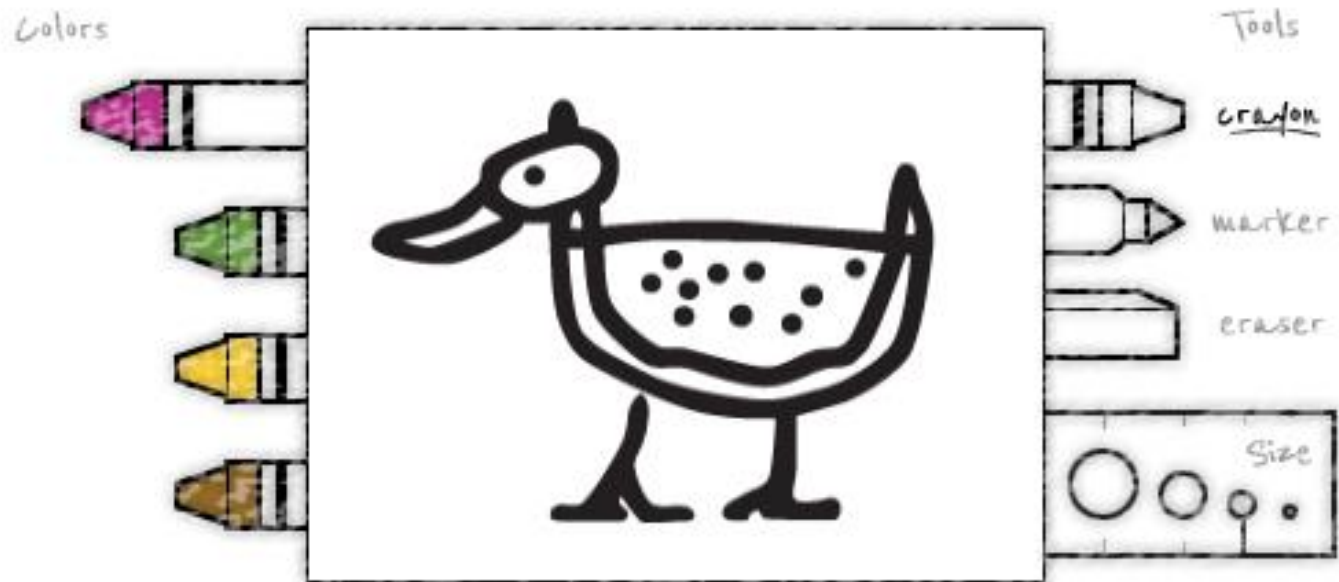


Домашнее задание #К.2



Воспользуйтесь заготовкой: [./homework/hw_k2.html](http://homework/hw_k2.html) и сделайте Paint настраиваемым, т.е. необходимо сделать чтобы пользователь мог выбрать цвет линии и толщину линии. Например как в примере: [./homework/hw_k2_demo.html](http://homework/hw_k2_demo.html)

При решении ДЗ #К.2 вам поможет



<http://www.williammalone.com/articles/create-html5-canvas-javascript-drawing-app/>