**Работа с изображениями в HTML**

**Вставка изображения**

Для вставки изображения в html-документ используется непарный тег **<img>**, который помещается в то место документа, где должно быть вставлено изображение.

У этого тега есть обязательный параметр **src**, значение которого указывает путь к нужному изображению в каталоге вашего сайта.

В качестве изображений могут выступать GIF-, PNG-, JPG-, JPEG – файлы.

Атрибуты width – ширина изображения, height – высота изображения.

*<img alt = “описание” src = “путь к файлу”>*

*<!doctype html>*

*<html>*

*<head>*

*<body>*

*<title>Изображение</title>*

*<img src="1.png"> (выбрать картинку с папки)*

*</body>*

*</head>*

*</html>*

Установим высоту и ширину изображения:

*<img src="2.jpg" width="500" height="500">*

У тега **<img>** есть еще один необязательный параметр **border**. С его помощью можно задать толщину рамки вокруг изображения. По умолчанию толщина рамки изображения нулевая, т.е. рамки нет.

Установим рамку к изображению:

*<img src="2.jpg" width="500" height="500" border="3">*

Цвет рамки совпадает с цветом текста на странице, заданным с помощью параметра **text** тега **<body>** , по умолчанию это черный цвет.

Для изображения, как и для параграфа, существует понятие выравнивания относительно текста и других изображений на странице. Задать тип выравнивания можно также с помощью параметра **align** тега **<img>**.

Ниже приведена таблица возможных значений параметра **align**:

align="**texttop**" - Верхняя граница изображения выравнивается по самому высокому текстовому элементу текущей строки.

align="**top**" - Верхняя граница изображения выравнивается по самому высокому элементу текущей строки.

align="**middle**" - Выравнивание середины изображения по базовой линии текущей строки.

align="**baseline**" - Выравнивание изображения по базовой линии текущей строки.

align="**bottom**" - Выравнивание нижней границы изображения по окружающему тексту.

align="**left**" - Выравнивает изображение по левому краю окна.

align="**right**" - Выравнивает изображение по правому краю окна.

Чтобы при обтекании текст не прилегал близко к изображению, можно задать отступы вокруг изображения. Сделать это можно с помощью параметров **hspace** (отступы слева и справа) и **vspace** (отступы сверху и снизу) тега **<img>**.

**Элемент Canvas**

Элемент **<сanvas>** позволяет рисовать на веб-страницах произвольные фигуры с помощью JavaScript (*или других клиентских скриптов*).

При создании элемента canvas необходимо задать атрибут id (*определяет имя элемента для доступа к нему из скриптов*) и его размеры с помощью атрибутов width (*ширина*) и height (*высота*).

*<canvas id="draw" width="300" height="200"></canvas>*

сам по себе canvas ничего не рисует. Он представляет собой холст, который предоставляет возможности для рисования клиентским скриптам.

**Рисование прямоугольников**

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| **fillRect(x,y,ширина,высота)** | Рисует закрашенный прямоугольник. |
| **strokeRect(x,y,ширина,высота)** | Рисует не закрашенный прямоугольник. |
| **clearRect(x,y,ширина,высота)** | Очищает указанную зону делая ее полностью прозрачной. |

параметры x и y задают величину смещения прямоугольника по горизонтали (x) и вертикали (y) от верхнего левого угла холста в пикселях.

*<!doctype html>*

*<html>*

*<head><title>Основы</title></head>*

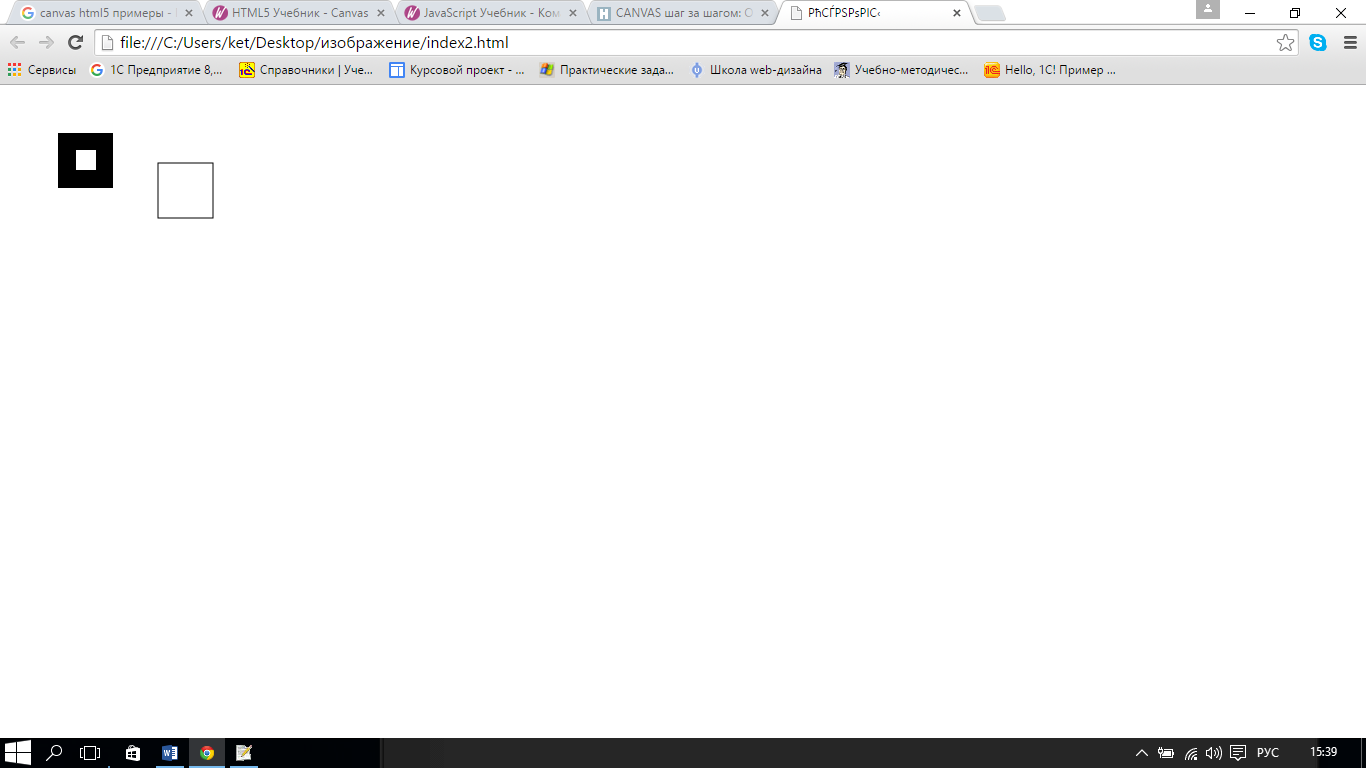
*<body>*

*<canvas id="draw" width="300" height="200"></canvas>*

*<script>*

*var canvas=document.getElementById("draw");* (находим нужный холст;)

*var x=canvas.getContext ("2d");* (обращаемся ко встроенному объекту, который содержит различные методы для рисования (первые два шага являются стандартными для рисования любого объекта в canvas);)

*x.fillRect (50,40,55,55);* (рисуем закрашенный прямоугольник;)

*x.strokeRect (150,70,55,55);* (рисуем не закрашенный прямоугольник;)

*x.clearRect (68,57,20,20);* (очищаем зону в закрашенном прямоугольнике.)

*</script>*

*</body>*

*</html>*

**Рисование составных фигур**

Составные фигуры состоят из нескольких соединенных простых объектов (таких как линии, круги и т.д.).

Таблица простых объектов.

|  |  |
| --- | --- |
| **Объекты** | **Описание** |
| **moveTo(x,y)** | Устанавливает координаты точки, из которой начнется рисование следующего объекта. |
| **lineTo(x,y)** | Рисует прямую линию. |
| **arc(x,y,радиус,нач\_угол,конеч\_угол)** | Рисует круг. Угол необходимо задавать в радианах, а не в градусах (*радианы=(Math.PI/180)\*градусы*). |
| **rect(x, y, ширина, высота)** | Рисует прямоугольник. |

*beginPath();*

*/\* Простые объекты помещаются здесь \*/*

*closePath(); // Автоматически завершает фигуру (соединяет конечную точку с начальной)*

*//Теперь необходимо вызвать один из методов для рисования фигуры определенной выше*

*stroke(); //нарисует фигуру не закрашенной*

*fill(); //нарисует фигуру закрашенной*

*<!doctype html>*

*<html>*

*<head><title>Основы</title></head>*

*<body>*

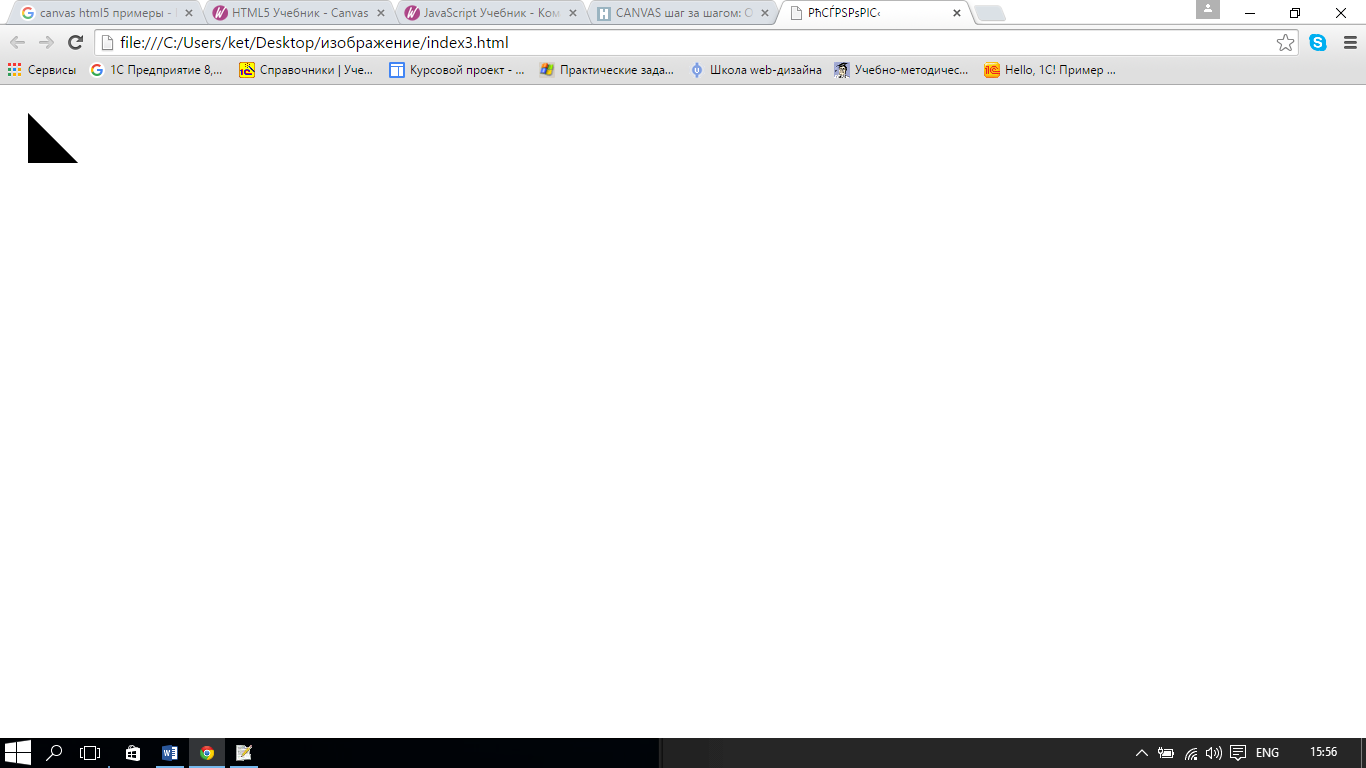
*<canvas id="draw" width="300" height="200"></canvas>*

*<script>*

*var canvas=document.getElementById("draw");*

*var x=canvas.getContext ("2d");*

*x.beginPath();*

*x.moveTo (20,20);*

*x.lineTo (70,70);*

*x.lineTo (20,70);*

*x.closePath();*

*x.fill ();*

*</script>*

*</body>*

*</html>*

*<html>*

*<body>*

*<canvas id='draw' width='200' height='200'></canvas>*

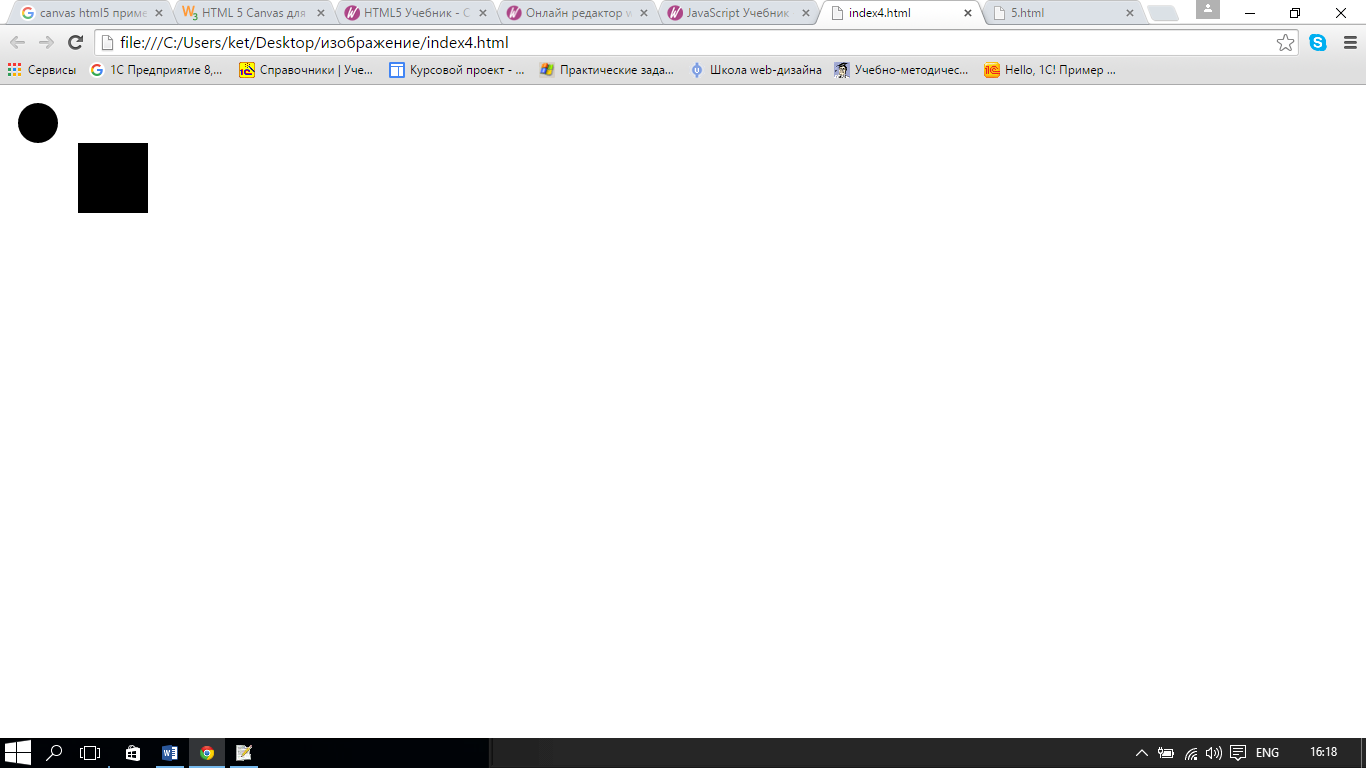
*<script>*

*var x1=(Math.PI/180)\*0;*

*var x2=(Math.PI/180)\*360;*

*var canvas=document.getElementById ("draw")*

*var x=canvas.getContext ("2d");*

*x.beginPath();*

*x.arc (30,30,20,x1,x2);*

*x.moveTo (100,100);*

*x.rect (70,50,70,70);*

*x.fill ();*

*</script>*

*</body>*

*</html>*

**Цвета**

Для раскрашивания нарисованных в canvas фигур предусмотрены свойства: **fillStyle** и **strokeStyle**.

Свойство **fillStyle** используется для применения цвета к закрашенным, а strokeStyle для применения цвета к не закрашенным фигурам.

*<html>*

*<head>*

*<body>*

*<title></title>*

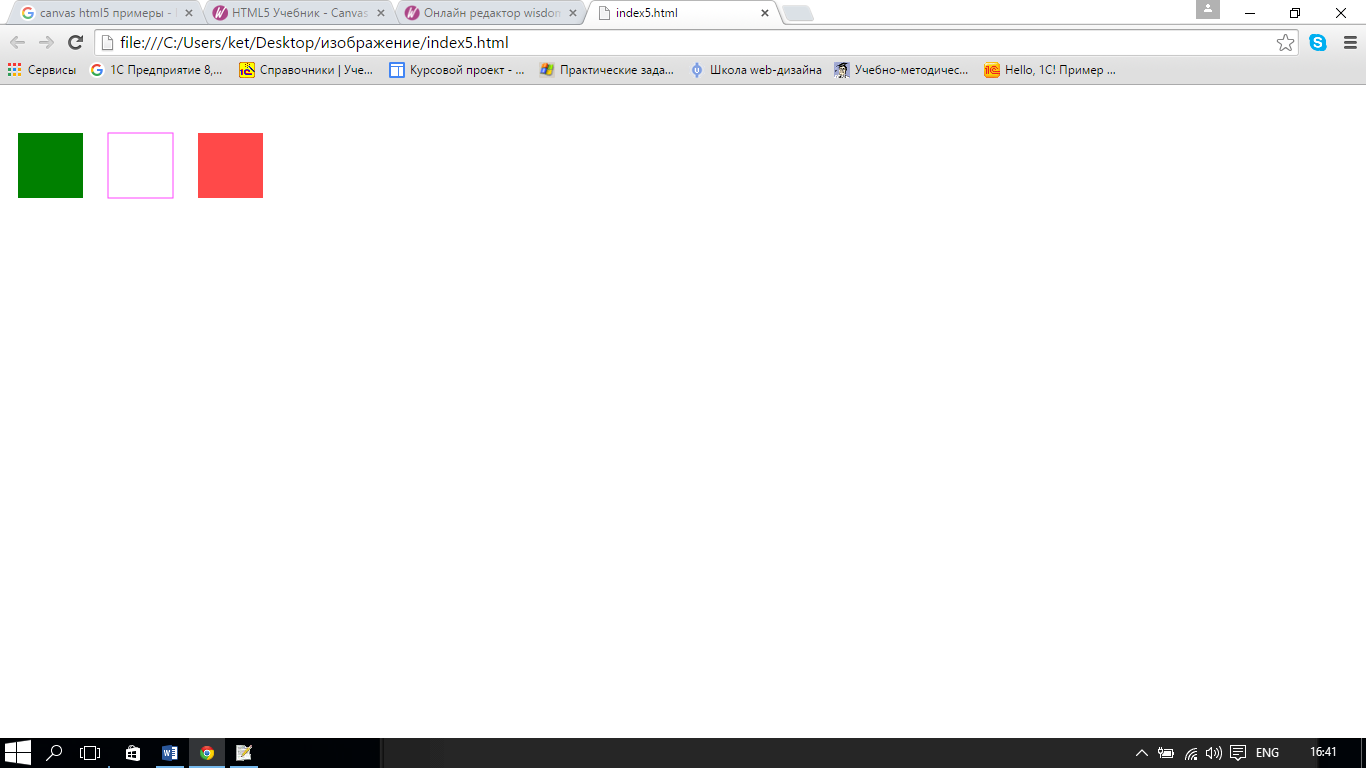
*<canvas id="draw" width="300" height="200"> </canvas>*

*<script>*

*var canvas=document.getElementById("draw");*

*var x=canvas.getContext("2d");*

*x.fillStyle="green";*

*x.fillRect(10,40,65,65);*

*x.strokeStyle="#FF45FF";*

*x.strokeRect (100,40,65,65);*

*x.fillStyle="rgb(255,73,73)";*

*x.fillRect(190,40,65,65);*

*</script>*

*</body>*

*</head>*

*</html>*

Помимо этих способов в canvas цвет может задавать с помощью rgba (*задается цвет и прозрачность элемента*).

*html>*

*<head>*

*<body>*

*<title></title>*

*<canvas id="draw" width="300" height="200"> </canvas>*

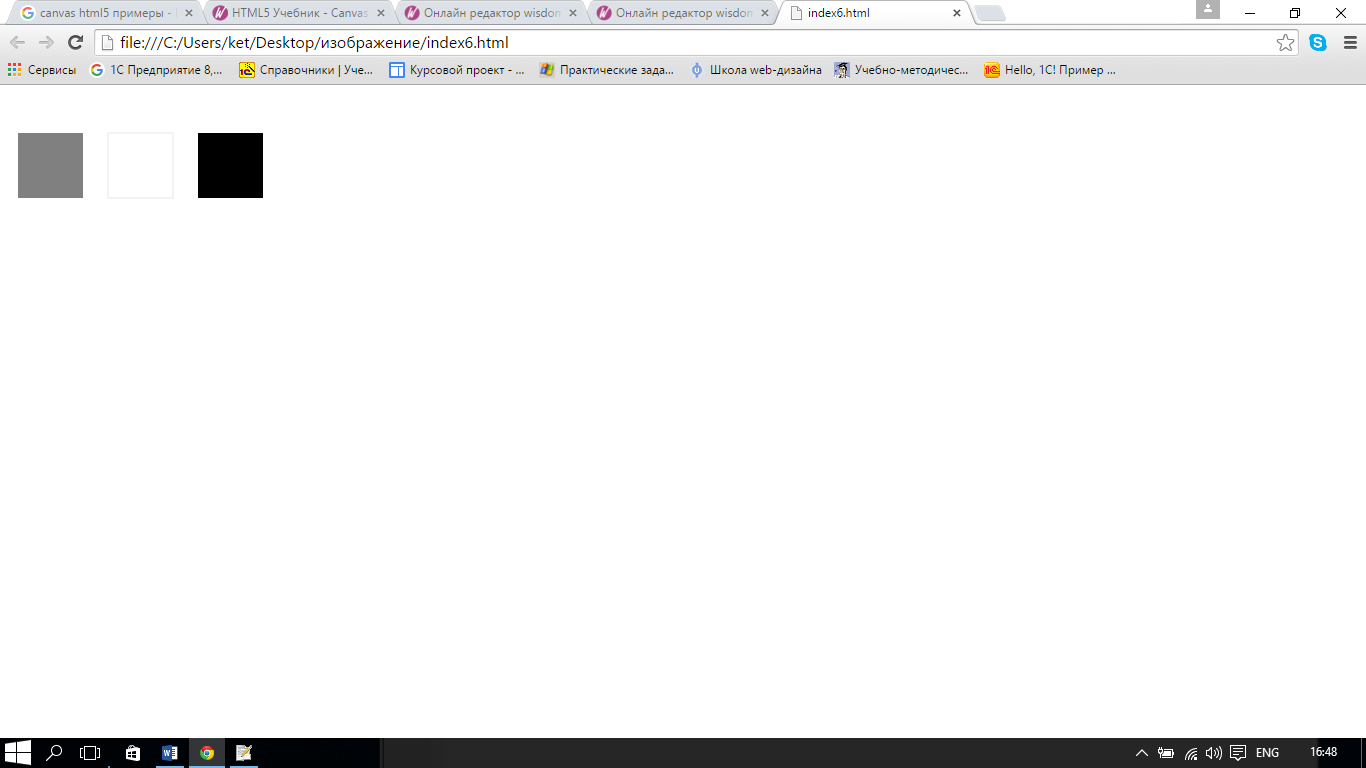
*<script>*

*var canvas=document.getElementById("draw");*

*var x=canvas.getContext("2d");*

*x.fillStyle="rgba(0,0,0,0.5)";*

*x.fillRect(10,40,65,65);*

*x.strokeStyle="rgba(0,0,0,0.1)";*

*x.strokeRect (100,40,65,65);*

*x.fillStyle="rgba(0,0,0,2)";*

*x.fillRect(190,40,65,65);*

*</script>*

*</body>*

*</head>*

*</html>*

**Оформление линий**

В HTML5 имеются несколько свойств для оформления линий нарисованных с помощью метода lineTo():**lineWidth**, **lineCap**, **lineJoin**.

С помощью свойства **lineWidth** Вы можете установить ширину линии (*по умолчанию линии имеют ширину 1 пиксель*).

*<html>*

*<head>*

*<body>*

*<title></title>*

*<canvas id="draw" width="300" height="200"> </canvas>*

*<script>*

*var canvas=document.getElementById("draw");*

*var x=canvas.getContext("2d");*

*// рисуем линию толщиной 3 пикселя*

*x.beginPath()*

*x.moveTo(10,10);*

*x.lineWidth=3;*

*x.lineTo (200,10);*

*x.stroke();*

*// рисуем линию толщиной 10 пикселей*

*x.beginPath()*

*x.moveTo(10,40);*

*x.lineWidth=10;*

*x.lineTo (200,40);*

*x.stroke();*

*// рисуем линию толщиной 1 пиксель*

*x.beginPath()*

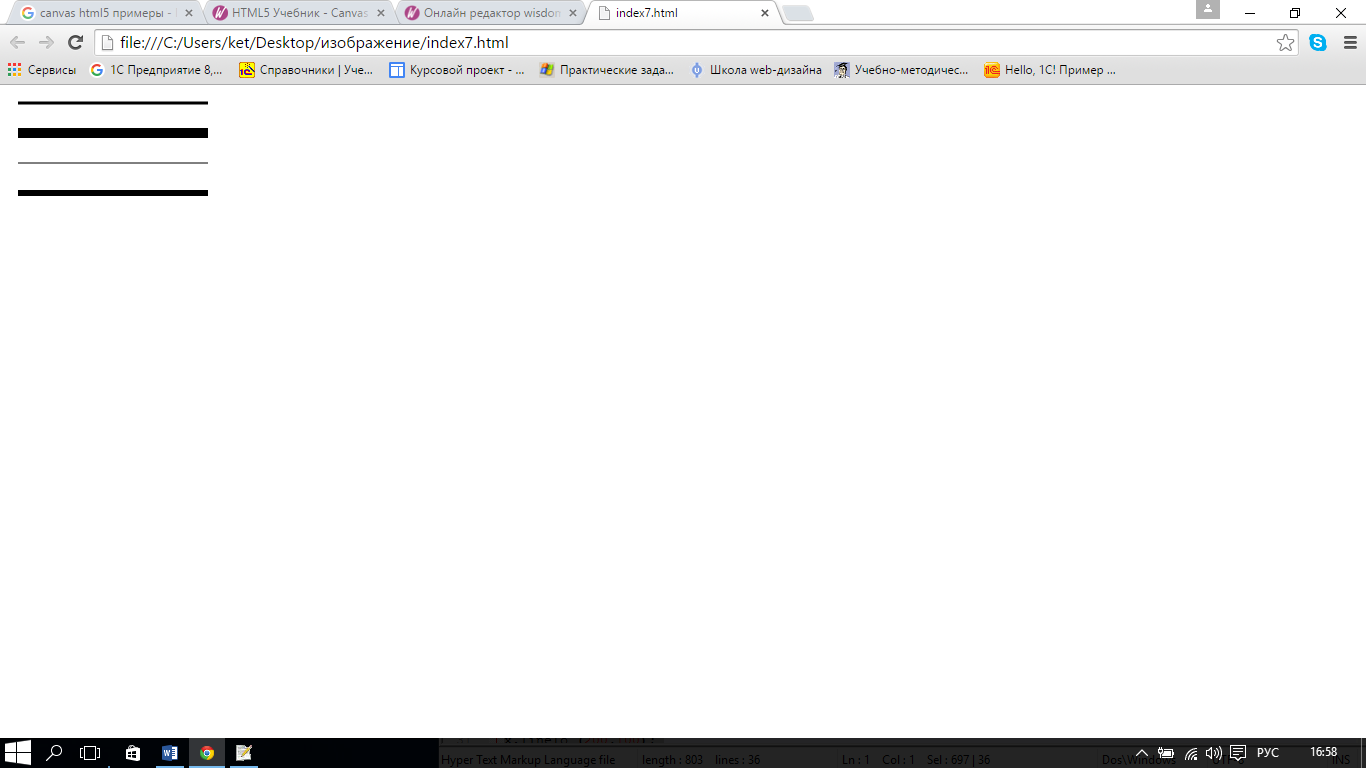
*x.moveTo(10,70);*

*x.lineWidth=1;*

*x.lineTo (200,70);*

*x.stroke();*

*// рисуем линию толщиной 6 пикселей*

*x.beginPath()*

*x.moveTo(10,100);*

*x.lineWidth=6;*

*x.lineTo (200,100);*

*x.stroke();*

*</script>*

*</body>*

*</head>*

*</html>*

С помощью свойства **lineCap** Вы можете оформлять кончики линии.

*<html>*

*<head>*

*<body>*

*<title></title>*

*<canvas id="draw" width="300" height="200"> </canvas>*

*<script>*

*var canvas=document.getElementById("draw");*

*var x=canvas.getContext("2d");*

*// нарисуем линию с закругленным кончиком*

*x.beginPath()*

*x.moveTo(10,10);*

*x.lineWidth=10;*

*x.lineCap="round";*

*x.lineTo (200,10);*

*x.stroke();*

*// нарисуем линию с квадратным кончиком*

*x.beginPath()*

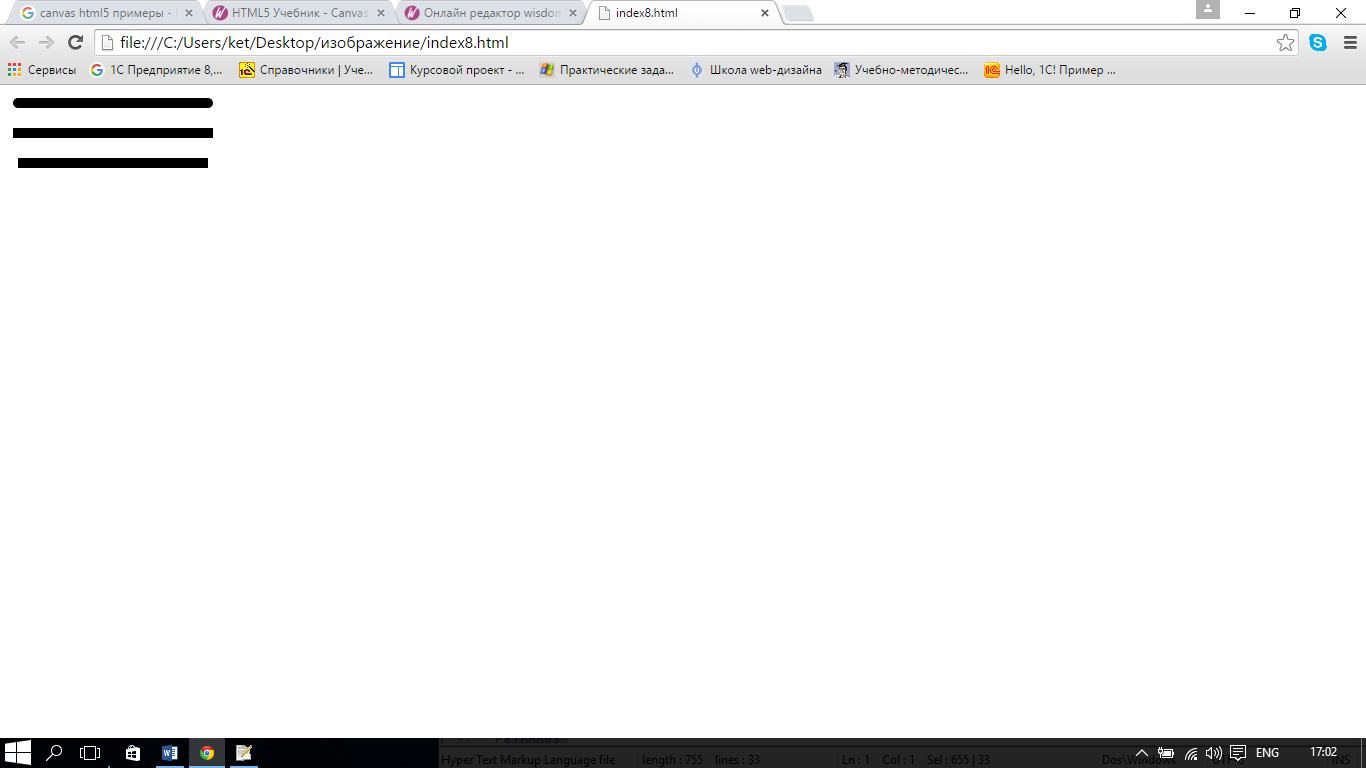
*x.moveTo(10,40);*

*x.lineWidth=10;*

*x.lineCap="square";*

*x.lineTo (200,40);*

*x.stroke();*

*// нарисуем линию со стандартным кончиком*

*x.beginPath()*

*x.moveTo(10,70);*

*x.lineWidth=10;*

*x.lineCap="butt";*

*x.lineTo (200,70);*

*x.stroke();*

*</script>*

*</body>*

*</head>*

*</html>*

С помощью свойства **lineJoin** Вы можете сглаживать стыки двух линий.

*<html>*

*<head>*

*<body>*

*<title></title>*

*<canvas id="draw" width="300" height="200"> </canvas>*

*<script>*

*var canvas=document.getElementById("draw");*

*var x=canvas.getContext("2d");*

*// Сгладим стыки линий*

*x.beginPath()*

*x.moveTo(10,10);*

*x.lineWidth=15;*

*x.lineJoin="round";*

*x.lineTo (50,50);*

*x.lineTo (100,10);*

*x.lineTo (170,80);*

*x.lineTo (210,40);*

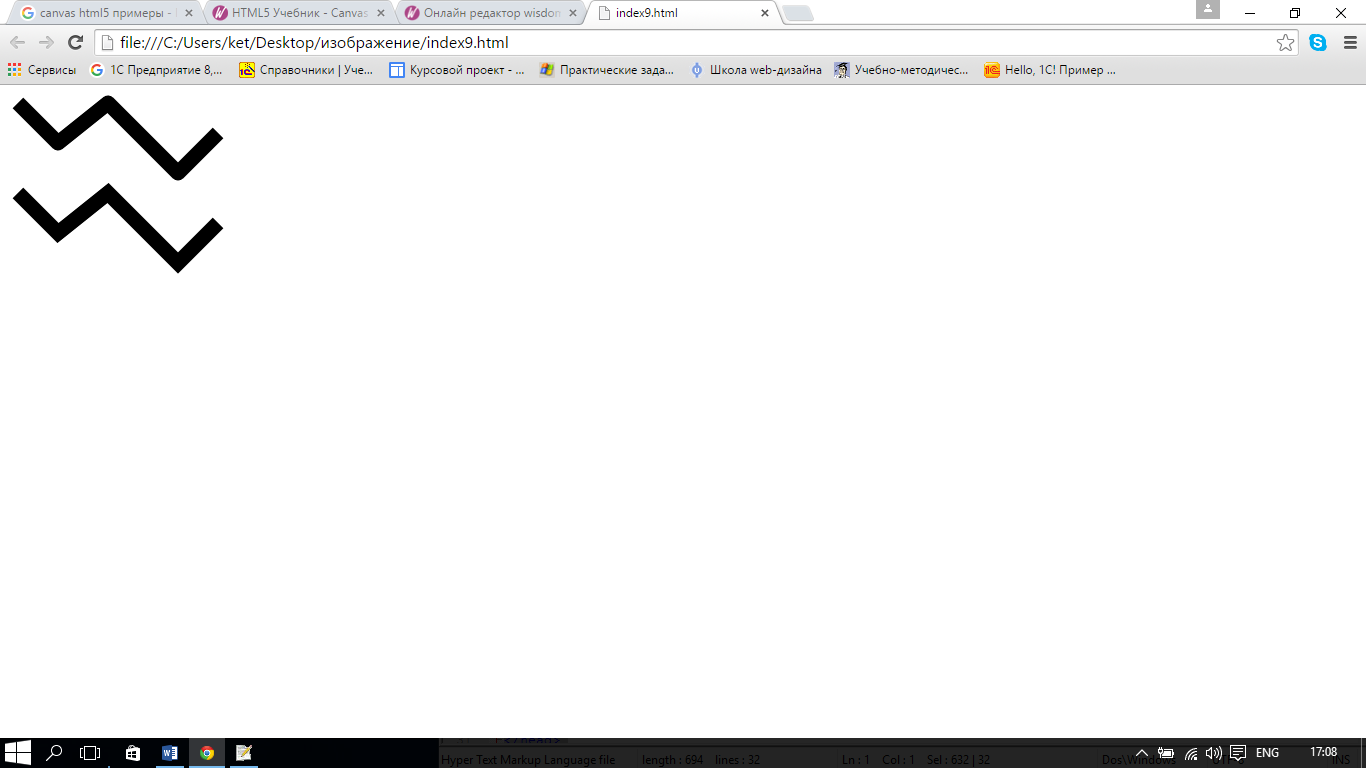
*x.stroke();*

*// нарисуем стыки не сглаженными (значения по умолчанию)*

*x.beginPath()*

*x.moveTo(10,100);*

*x.lineWidth=15;*

*x.lineJoin="miter";*

*x.lineTo (50,140);*

*x.lineTo (100,100);*

*x.lineTo (170,170);*

*x.lineTo (210,130);*

*x.stroke();*

*</script>*

*</body>*

*</head>*

*</html>*