

**Project: Browser-Based Drawing Tool**

**Performed by**: Zhangazy Nurshapagat

**Checked by:** Shoiynbek Temirlan

**Almaty, 2024**

**Introduction**

The purpose of this project is to provide a simple and accessible drawing tool through a web application. The project allows users to draw directly in their browser without the need to install any additional software. No special skills or professional knowledge are required to use it.

This project aims to create an optimized tool for the average user by integrating an intuitive interface and essential functionality using HTML, CSS, and JavaScript technologies.

1. **Purpose of the project.**

The primary goal of this project is to learn how to practically apply web technologies (HTML, CSS, JavaScript) and provide users with an intuitive and convenient drawing tool. Allows the user to draw in different colors and sizes, use the eraser, and use the "undo" and "redo" functions.

1. **Used programming languages and methods**

**HTML** - is used to create the structure of the project. All control elements and the canvas element for drawing are added using HTML.

**CSS** - was used to enhance the visual appeal of the project and adjust the positioning of elements.

**JavaScript** - was used to provide the functionality of the project. Features such as drawing, erasing, changing the brush color or size, "undo," and "redo" were implemented using JavaScript.

**document.getElementById() and the DOM:**  
Used to manage HTML elements in JavaScript. For example, changing the brush color and size or accessing the <canvas> element.

**Arrays:**  
**undoStack** and **redoStack:**  
Used to store the previous and subsequent states of the drawing. These arrays enable the Undo/Redo functionality.

1. **Main Features**

**Drawing**: The user can draw on the canvas using the mouse.

**Change color:** The brush color can be changed using the color picker selector.

**Change brush size:** The brush diameter can be adjusted using the size control slider.

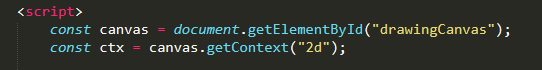
**Eraser:** Allows the user to erase certain parts of the drawing.

**Undo:** Cancels the last action.

**Redo:** Reverts the "undone" action.

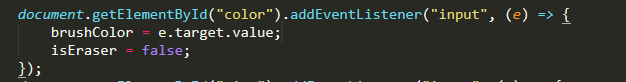
1. **Using JavaScript elements:**

**Canvas:** The <canvas> element is the main area for drawing. All actions, such as drawing or erasing, are performed using the 2D context.

****

**Color selection:**

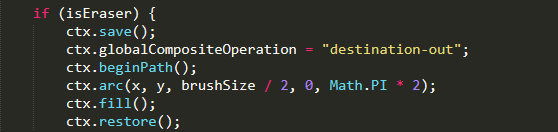
The <input type="color"> element allows the user to select the brush color. The color is continuously updated through the input event.

****

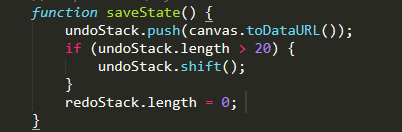
**Changing the brush size:** The <input type="number"> element allows the user to adjust the brush diameter. This value is updated through the input event.



**Eraser:** The eraser function is implemented by setting the globalCompositeOperation mode to destination-out. This allows the brush to erase parts of the drawing.

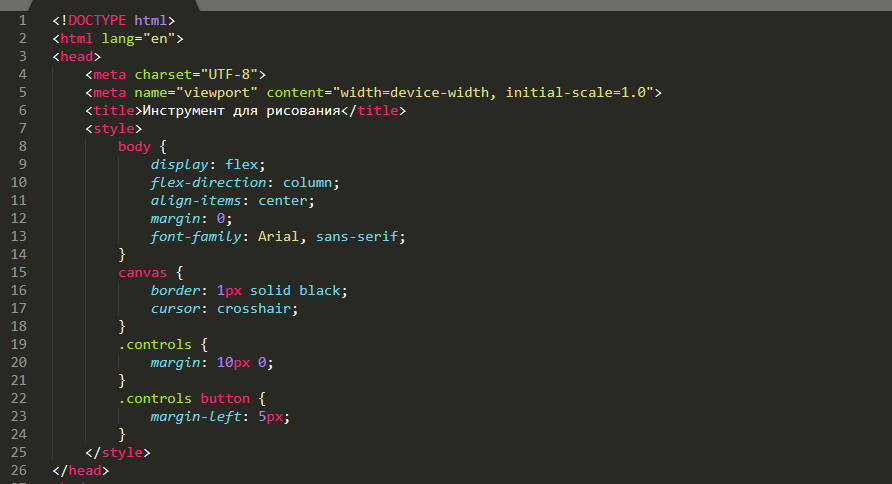


**Undo and Redo:** The undoStack and redoStack stacks are used to store the states of the canvas. The toDataURL method is used to save the canvas state.



1. **Individual explanation for each code**

**CSS**

****

<!DOCTYPE html> This declaration specifies the document type as HTML5.

<html lang="en"> specifies the primary language of the webpage, which in this case is English.

<meta charset="UTF-8"> This meta tag specifies the character encoding of the document, i.e., the character set used for the text.

<meta name="viewport" content="width=device-width, initial-scale=1.0"> The viewport ensures that the page width matches the device's screen width and sets the initial scale of the page to 100%.

The <style> tag is used to define internal CSS styles.

Body{} The body selector affects the entire page.

* display: flex; — Ensures that elements are placed within a flexible container. Here, the elements are arranged vertically.
* flex-direction: column; — Specifies the vertical arrangement of the elements.
* align-items: center; — Centers all inner elements along the horizontal axis.
* margin: 0; — Removes the outer margins of the page.
* font-family: Arial, sans-serif; — Sets the font to Arial, or another sans-serif font if Arial is unavailable.

**Canvas {}:**  
The <canvas> tag is an HTML5 element used for creating dynamic graphics.

* border: 1px solid black; — Adds a thin black border around the drawing area.
* cursor: crosshair; — Changes the mouse pointer to a "crosshair," visually indicating a drawing action.

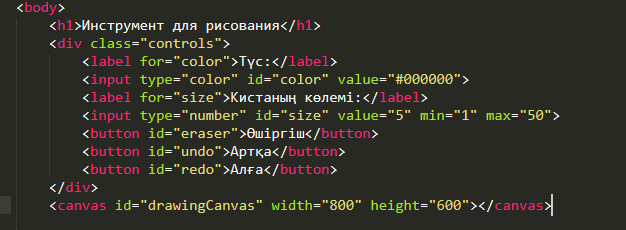
**.controls class:**  
The .controls class defines the container for the control elements.

* margin: 10px 0; — Leaves a 10px gap on the top and bottom of the elements.

**.controls button {}:**  
This selector applies to all <button> tags, intended for control elements.

* margin-left: 5px; — Adds a 5px space to the left of each button, separating them from each other.

**HTML**



<div class="controls">

* The attribute assigns the class "controls" to the container, allowing it to be targeted and styled or manipulated using CSS or JavaScript. This is where the control elements will be placed.

<label for="color">Түс:</label>

* <label the tag is used to define a label for form elements, making the form more understandable to the user by displaying its name.
* The for="color" attribute indicates that the label is associated with the element that has the id="color". In this case, it labels the input element used for selecting a color.

<input type="color" id="color" value="#000000">

* The type="color" attribute displays a color selection panel in the browser, enabling the user to pick a color.
* The id="color" attribute uniquely identifies the element, making it possible to select and manipulate it through JavaScript.
* The value="#000000" attribute initially sets the color to black (#000000).

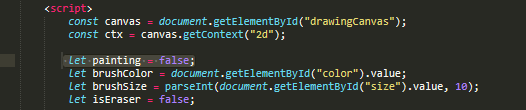
<input type="number" id="size" value="5" min="1" max="35">

* The type="number" attribute specifies that only numbers can be entered in this field.
* The id="size" attribute uniquely identifies the element, enabling it to be used with JavaScript.
* The value="5" attribute sets the initial brush size to 5 pixels.
* The min="1" and max="50" attributes allow the brush size to be adjusted between 1 and 50 pixels.

<canvas id="drawingCanvas" width="800" height="600"></canvas>

* The <canvas> tag is an HTML5 element used for dynamically displaying graphics and images.
* The id="drawingCanvas" attribute allows the canvas to be accessed and controlled through JavaScript.
* The width="800" and height="600" attributes set the canvas dimensions to 800 pixels in width and 600 pixels in height, providing a space for drawing.

**JS**

****

****



1. **User Guide**

**Interface**  
The drawing area is located in the center of the webpage.  
The control panel includes the following elements:

* Select the brush color.
* Adjust the brush size.
* Buttons: "Eraser", "Undo", "Redo".

**How to use it?**

* Select the brush color and size using the corresponding control elements.
* Move the cursor to the canvas and draw by pressing the left mouse button.
* To erase unwanted parts of the drawing, click the "Eraser" button and perform the erase action on the canvas.
* To undo the last action, click the "Undo" button.
* To redo the undone action, click the "Redo" button.

1. **Usage examples**

Example: Select a color and draw

* Choose red from the palette.
* Set the brush size to 10.
* Press the left mouse button and draw a circle.

Example: Erase and undo

* Click the "Eraser" button and erase a specific part of the drawing.
* Click the "Undo" button to restore the erased part.

**Note**

* The maximum history size (undoStack) is 20 steps.
* The eraser function depends on the brush size.

1. **Conclusion:**

This project offers users a simple and intuitive tool for drawing. Users can create drawings directly in their browser without any special skills or additional software. The project stands out with its straightforward interface and provides opportunities for further expansion beyond its basic functionalities. Overall, this project is an excellent solution that demonstrates the potential of web technologies while making drawing accessible to beginner users.