# Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И. И. Ползунова»

Университетский технологический колледж

Отчет защищен с оценкой_	
Преподаватель	С.В. Умбетов
«»	2025 г.

# Отчёт по лабораторной работе № 1 «Реализация шифра ROT13 на JavaScript» <u>ЛР 09.03.01.14.002</u>

Студент группы <u>1ИСП-21</u>	А.А. Кайль
группа	и.о., фамилия
Преподаватель ассистент, к. т. н.	С. В. Умбетов
должность, ученая степень	и.о., фамилия

# Лабораторная работа №1

Цели и задачи р	<b>аботы:</b> необхо	димо создать h	tml страницу и ј	s код, страница
должна быть валидной	стандарту НТ	ML5. В коде реа	ализовать шифр	ование ROT13
для четного варианта и	и расшифровку	для нечетного.		
Задание принял:				Кайль А.А.
эадапис припял	Подпись		ФИО	канль А.А.

### Ход работы

ROT13 («поворот на 13 позиций») — это простой и широко известный шифр замены букв, который используется на онлайн-форумах и в текстовых файлах для сокрытия текста, например предупреждений о спойлерах или оскорбительном контенте. Этот метод заменяет букву на 13-ю по счёту в алфавите. А становится N, B становится O, C становится P и так далее.

ROT13 является обратным к самому себе, то есть для отмены ROT13 применяется тот же алгоритм, поэтому одно и то же действие можно использовать для кодирования и декодирования.

Алгоритм не обеспечивает криптографическую безопасность и часто приводится в качестве канонического примера слабого шифрования.

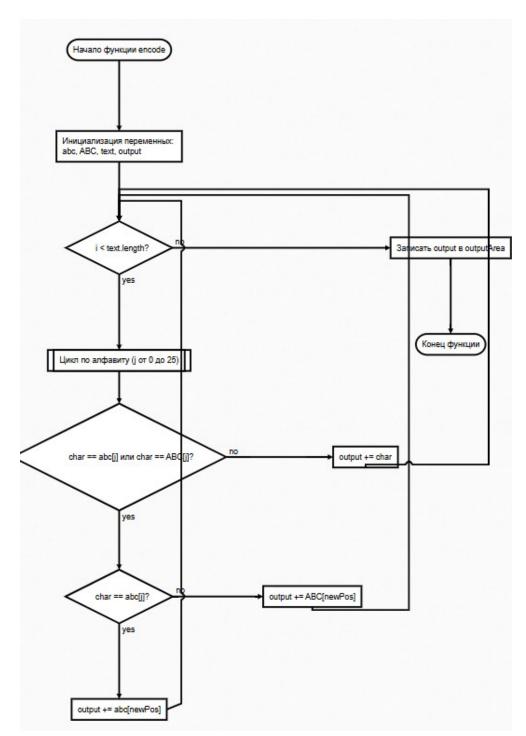


Рисунок 1 — Блок-схема

```
function encode() {
    let abc = 'abcdefghijklmnopqrstuvwxyz';
    let ABC = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    let text = document.getElementById('inputArea').value;
    let output = '';
```

Сначала создадим основную функцию encode(), abc – строчные буквы, ABC – заглавные буквы, text — для шифрования из текстового поля с id="inputArea". и output чтобы записать сюда зашифрованный текст.

```
for (let i = 0; i < text.length; i++) {
    let char = text[i];
    let found = false;</pre>
```

Это цикл, который проходит по каждому символу введённого текста. char — текущий символ (например, h в слове hello) found — флаг, который показывает была ли буква найдена в алфавите.

```
for (let j = 0; j < abc.length; j++) {
    if (char = = = abc[j] || char = = = ABC[j]) {
```

for ((let j = 0; j < abc.length; j++) вложенный цикл проверяет есть ли текущий символ (char) в алфавите (abc или ABC).

if (char === abc[j]  $\parallel$  char === ABC[j]) если символ совпадает с буквой в алфавите (строчной или заглавной) выполняется шифрование.

```
let shift = 13;
let newPos = (j + shift) % abc.length;
```

newPos = (j + shift) % abc.length вычисляет позицию буквы после сдвига. Если j + shift больше 26 позиция следует в начало.

```
Буква n (индекс 13): (13 + 13) \% 26 = 0 \longrightarrow a
```

Если строчный то берём новую букву из abc[newPos], если заглавный ABC[newPos]

Для цифр знаков препинаний и тд.

```
if (!found) {
          output += char;
}
```

Если символ не был найден в алфавите то выводится без изменений

```
index.html M
                  # style.css M
                                   JS script.js M X
practice > script > JS script.js > ...
       function encode() {
           let abc = 'abcdefghijklmnopqrstuvwxyz';
           let ABC = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
           let text = document.getElementById('inputArea').value;
           let output = '';
           for (let i = 0; i < text.length; i++) {</pre>
               let char = text[i];
               let found = false;
               for (let j = 0; j < abc.length; j++) {</pre>
                   if (char === abc[j] || char === ABC[j]) {
                       let newPos = (j + 13) \% 26;
                        if (char === abc[j]) {
                            output += abc[newPos];
                           output += ABC[newPos];
                        found = true;
                        break;
               if (!found) {
                   output += char;
           document.getElementById('outputArea').value = output;
      document.addEventListener('DOMContentLoaded', function() {
 32
           document.getElementById('inputArea').addEventListener('input', encode);
 34
      });
```

Рисунок 2 — Код JS

# Тестирование

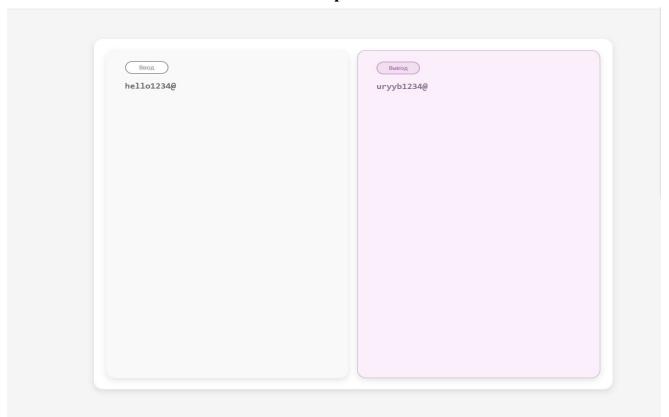


Рисунок 3 — Тестирование шифровка

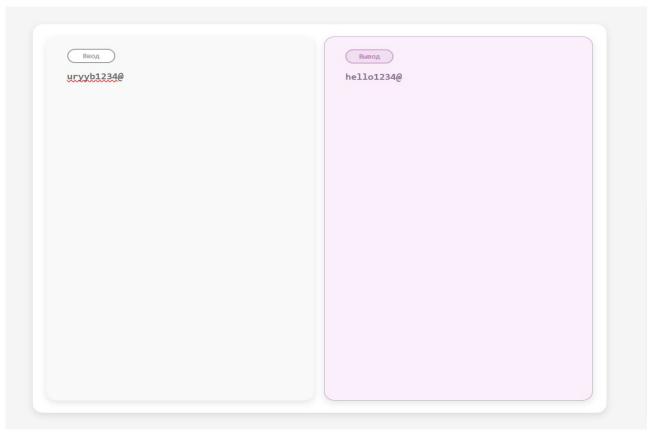


Рисунок 4 — Тестирование дешифровка

# Проверка

Таблица 1: Проверка шифровки

тионици т. проверки шперовки					
Символ	Тип	Позиция (j)	newPos = (j+13)%26	Измененный символ	Результат
h	строчная	7	(7+13)%26 = 20	u (abc[20])	u
e	строчная	4	(4+13)%26 = 17	r (abc[17])	ur
1	строчная	11	(11+13)%26 = 24	y (abc[24])	ury
1	строчная	11	(11+13)%26 = 24	y (abc[24])	uryy
0	строчная	14	(14+13)%26 = 1	b (abc[1])	uryyb
1	цифра	-	-	1 (без изменений)	uryyb1
%	символ	-	-	% (без изменений)	uryyb1234%

Таблица 2: Проверка дешифровки

	імвол  Тип	<u> </u>	newPos =	Изначальный	Результат
Символ		(j)	(j+13)%26	символ	
u	строчная	20	(20+13)%26 = 7	h (abc[7])	h
r	строчная	17	(17+13)%26 = 4	e (abc[4])	he
У	строчная	24	(24+13)%26 = 11	1 (abc[11])	hel
у	строчная	24	(24+13)%26 = 11	1 (abc[11])	hell
b	строчная	1	(1+13)%26 = 14	o (abc[14])	hello
1	цифра	-	-	1 (без изменений)	hello1
%	символ	-	-	% (без изменений)	hello1234%

#### **HTML**

```
index.html M X
                  # style.css M
                                  JS script.js M
practice > ♦ index.html > ♦ html > ♦ body > ♦ main > ♦ div.centralBlock
      <!DOCTYPE html>
      <html lang="ru">
           <meta charset="UTF-8">
           <title>rot13</title>
           <link rel="stylesheet" href="./style/style.css">
      <body>
           <header>
           </header>
 11
 12
           <main>
               <div class="centralBlock">
 13
                   <div class="inputPanel">
                       Ввод
                     <textarea id="inputArea"></textarea>
                   </div>
 19
                   <div class="outputPanel">
                       Вывод
 21
                       <textarea id="outputArea" readonly></textarea>
               </div>
           </main>
           <footer>
           </footer>
           <script src="./script/script.js"></script>
 30
       </body>
      </html>
```

Рисунок 5 — Код html

#### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for contents of text-input area

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

Document checking completed. No errors or warnings to show.

Рисунок 6 — Проверка html на валидность

```
index.html M
                # style.css M X JS script.js M
practice > style > # style.css > 6 header
      body {
          font-family: 'Consolas', monospace;
background-color: ■#f5f5f5;
          margin: 0;
          padding: 0;
          color: □#333;
          display: flex;
          align-items: center;
          justify-content: center;
  10
           min-height: 100vh;
      header, footer {
           text-align: center;
           padding: 10px;
           font-family: 'Consolas', monospace;
 16
       .centralBlock {
         display: flex;
           justify-content: center;
          height: 80vh;
          width: 60vw;
          max-width: 800px;
          min-width: 300px;
          background-color: white;
          padding: 20px;
           border-radius: 15px;
          box-shadow: 0 4px 10px □rgba(0, 0, 0, 0.1);
           font-family: 'Consolas', monospace;
       .inputPanel, .outputPanel {
         flex: 1;
          display: flex;
          flex-direction: column;
          border-radius: 15px;
          padding: 15px;
          position: relative;
          min-width: 120px;
           font-family: 'Consolas', monospace;
       .inputPanel {
           background-color: ■#f9f9f9;
           box-shadow: 0 4px 10px ☐ rgba(0, 0, 0, 0.1);
           margin-right: 15px;
       .outputPanel {
          border: 1px solid □#8c729589;
           background-color: #fbeffb;
          box-shadow: 0 4px 10px □rgba(0, 0, 0, 0.1);
       .inputPanel p, .outputPanel p {
          display: flex;
          justify-content: center;
          align-items: center;
          width: 60px;
           font-family: 'Consolas', monospace;
           padding: 5px 10px;
           border-radius: 15px;
           font-weight: 750;
           font-size: 11px;
           margin-top: 5px;
           margin-left: 15px;
```

```
practice > style > # style.css > ...

77 .outputPanel p {

80 color: \( \pm \)#bd78bd;
                 padding: 1%;
                 textarea {
  height: 90%;
  padding: 0 15px;
  border-radius: 10px;
  resize: none;
  font-family: 'Consolas', monospace;
  font-size: 15px;
  line-height: 1.5;
  font-weight: 550;
  border: none;
                           font-weight: 558;
border: none;
outline: none;
color: □#6b6b6b;
background-color: transparent;
                    .outputPanel textarea {
    color: □#8c7295;
    caret-color: transparent;
                    .outputPanel textarea[readonly] {
   cursor: default;
                  @media (max-width: 768px) {
    .central.block {
        flex-direction: column;
        height: auto;
        width: 85vw;
        padding: 15px;
}
                        .inputPanel, .outputPanel {
   width: 100%;
   margin-right: 0;
   margin-bottom: 15px;
                            textarea {
   height: 200px;
                  @media (max-width: 480px) {
    .central.block {
        width: 95vw;
        padding: 10px;
                            .inputPanel p, .outputPanel p {
  width: 50px;
  font-size: 10px;
  padding: 4px 8px;
  margin-left: 10px;
                             textarea {
    font-size: 14px;
    padding: 0 10px;
```

Рисунок 8 — Стили css @media



Перейти к: Проверенный CSS

W3C результаты проверки CSS для TextArea (CSS3 + SVG)

Поздравляем! Ошибок не обнаружено.

Этот документ проходит проверку по стандарту  $\underline{\text{CSS3} + \text{SVG}}$  !

Рисунок 8 — Стили css проверка на валидность

## Вывод

В ходе данной лабораторной работы была создана функция, которая реализует алгоритм rot13 на JavaScript, который шифрует и дешифрует введенные данные. Проблемы: обработка регистра букв(раздельные алфавиты строчные и заглавные), использование % 26 для обратного сдвига.

https://github.com/him1k0ta/practice.git