

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»
ЛР 09.03.01.14.002

Студент группы 1ИСП-21 _____ А.А. Кайль
группа и.о., фамилия

Преподаватель ассистент, к. т. н. _____ С. В. Умбетов
должность, ученая степень и.о., фамилия

БАРНАУЛ 2025

Лабораторная работа №1

Цели и задачи работы: необходимо создать html страницу и js код, страница должна быть валидной стандарту HTML5. В коде реализовать шифрование ROT13 для четного варианта и расшифровку для нечетного.

Задание принял: _____



Подпись

ФИО

Кайль А.А.

Ход работы

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.

```
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;
```

Это цикл, который проходит по каждому символу введенного 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] || 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 —> a

```
            if (char === abc[j]) {  
                output += abc[newPos];  
            } else {  
                output += ABC[newPos];  
            }  
            found = true;  
            break;
```

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

Тестирование

Ввод

hello1234%

Вывод

uryyb1234%

Шифровать/Дешифровать

А В С D E F G H I J K L M
N O P Q R S T U V W X Y Z
H E L L O
U R Y Y B

Рисунок 1 — Тестирование шифровка

Ввод

uryyb1234%

Вывод

hello1234%

Шифровать/Дешифровать

А В С D E F G H I J K L M
N O P Q R S T U V W X Y Z
H E L L O
U R Y Y B

Рисунок 2 — Тестирование дешифровка

Проверка

Таблица 1: Проверка шифровки

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
l	строчная	11	$(11+13)\%26 = 24$	y (abc[24])	ury
l	строчная	11	$(11+13)\%26 = 24$	y (abc[24])	uryy
o	строчная	14	$(14+13)\%26 = 1$	b (abc[1])	uryyb
1	цифра	-	-	1 (без изменений)	uryyb1
%	символ	-	-	% (без изменений)	uryyb1234%

Таблица 2: Проверка дешифровки

Символ	Тип	Позиция (j)	newPos = $(j+13)\%26$	Изначальный символ	Результат
u	строчная	20	$(20+13)\%26 = 7$	h (abc[7])	h
r	строчная	17	$(17+13)\%26 = 4$	e (abc[4])	he
y	строчная	24	$(24+13)\%26 = 11$	l (abc[11])	hel
y	строчная	24	$(24+13)\%26 = 11$	l (abc[11])	hell
b	строчная	1	$(1+13)\%26 = 14$	o (abc[14])	hello
1	цифра	-	-	1 (без изменений)	hello1
%	символ	-	-	% (без изменений)	hello1234%

Код

html

```
<!DOCTYPE html>
<html lang="ru">
  <header>
    <title>rot13</title>
    <link rel="stylesheet" href="./style/style.css">
    <script src="./script/script.js"></script>
  </header>
  <body>
    <header>
    </header>
    <main>
      <div class="central block">
        <div class="inputPanel">
          <p>Ввод</p>
          <textarea id="inputArea"></textarea>
          <button
onclick="encode()">Шифровать/Дешифровать</button>
        </div>
        <div class="outputPanel">
          <p>Вывод</p>
          <textarea
id="outputArea"></textarea>
        </div>
      </div>
    </main>
    <div class="rot13-demo">
      <p>
        <span class="original">A B C D E F G H I
J K L M</span><br>
        <span class="rot13">N O P Q R S T U V W
X Y Z</span>
      </p>
      <p>
        <span class="original">H E L L
O</span><br>
        <span class="rot13">U R Y Y B</span>
      </p>
    </div>
    <footer>
    </footer>
  </body>
</html>
```

CSS

```
body {
  font-family: 'Courier New', monospace;
  background-color: #f5f5f5;
  margin: 0;
  padding: 20px;
  color: #333;
}
header, footer {
  text-align: center;
  padding: 10px;
}
.central.block {
  display: flex;
  justify-content: center;
  gap: 30px;
  max-width: 800px;
  margin: 0 auto;
  background-color: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}
.inputPanel, .outputPanel {
  flex: 1;
  display: flex;
  flex-direction: column;
}
.inputPanel p, .outputPanel p {
  font-weight: bold;
  margin-bottom: 5px;
  color: #444;
}
textarea {
  height: 150px;
  padding: 10px;
  border: 2px solid #ddd;
  border-radius: 4px;
  resize: none;
  font-family: 'Courier New', monospace;
  font-size: 16px;
}
button {
  background-color: #4a6fa5;
  color: white;
  border: none;
```

```
padding: 8px 15px;
margin-top: 10px;
border-radius: 4px;
cursor: pointer;
font-family: 'Courier New', monospace;
transition: background-color 0.3s;
}

button:hover {
    background-color: #3a5a8f;
}

.rot13-demo {
    text-align: center;
    margin: 30px auto;
    font-size: 18px;
    line-height: 1.6;
}

.rot13-demo .original {
    color: #4a6fa5;
    font-weight: bold;
}

.rot13-demo .rot13 {
    color: #d35400;
    font-weight: bold;
}
```


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.