

Task1

```
7 /head>
8 <body>
9   <script>
10     const firstName = prompt("Enter your first name:");
11     const lastName = prompt("Enter your last name:");
12     const dateOfBirth = prompt("Enter your date of birth (mm/dd/yyyy)");
13     function getAge(dateString) {
14       var today = new Date();
15       var birthDate = new Date(dateString);
16       var age = today.getFullYear() - birthDate.getFullYear();
17       var m = today.getMonth() - birthDate.getMonth();
18       if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {
19         age--;
20       }
21       return age;
22     }
23     console.log('Hello, my name is ${firstName} ${lastName}. I am ${getAge(dateOfBirth)} years old.');
```

Консоль

Уровни по умолчанию 1 проблема: 1 56 скрыто

Регистрировать запросы XMLHttpRequests

Энергичное вычисление

Автозаполнение из истории

Рассматривать оценку кода как пользовательское действие

Hello, my name is ilyas kenbay. I am 19 years old. task1.html:23

Task2

```
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 </head>
7 <body>
8   <script>
9     const age = prompt("Enter your age: ");
10    const DriverLic = prompt("Do you have driver license? yes/no");
11    const canDrive = age>=18 && DriverLic === "yes";
12    console.log('You are eglible to drive: ${canDrive ? "yes" : "no"}');
13  </script>
14  <!-- Code injected by live-server -->
```

Консоль

Уровни по умолчанию 1 проблема: 1 5 скрыто

Регистрировать запросы XMLHttpRequests

Энергичное вычисление

Автозаполнение из истории

Рассматривать оценку кода как пользовательское действие

You are eglible to drive: yes task2.html:12

Task3

Screenshot of a web browser's developer console showing the execution of a JavaScript script. The script prompts the user to enter a number and then checks several conditions: if the number is even, odd, a multiple of 5, and greater than 10. The console output shows the user entered 12, and the script correctly identifies it as even, not odd, not a multiple of 5, and greater than 10, concluding it is special.

```
8 <script>
9   const number = prompt("Enter a number:");
10
11   const isEven = number % 2 === 0;
12   const isOdd = number % 2 !== 0;
13   const isMultipleOf5 = number % 5 === 0;
14   const isGreaterThan10 = number > 10;
15
16   console.log(`You entered: ${number}`);
17   console.log(`Is the number even? ${isEven}`);
18   console.log(`Is the number odd? ${isOdd}`);
19   console.log(`Is the number a multiple of 5? ${isMultipleOf5}`);
20   console.log(`Is the number greater than 10? ${isGreaterThan10}`);
21
22   if (isMultipleOf5 || isGreaterThan10) {
23     console.log("The number is special!");
24   }
25 </script>
```

Console output:

```
You entered: 12
Is the number even? true
Is the number odd? false
Is the number a multiple of 5? false
Is the number greater than 10? true
The number is special!
```

Task4

Screenshot of a web browser's developer console showing the execution of a JavaScript script. The script prompts the user to enter a number, a string, and a boolean value, then logs the input and its type. The console output shows the user entered 11, 'HI', and true, and the script correctly identifies their types as number, string, and boolean respectively.

```
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 </head>
7 <body>
8   <script>
9     const num = prompt("Enter a number:");
10    const str = prompt("Enter a string:");
11    const bool = prompt("Enter a boolean value (true/false):");
12
13    console.log(`Number : ${num}, Type: ${typeof parseInt(num)}`);
14    console.log(`String : ${str}, Type: ${typeof str}`);
15    console.log(`Boolean : ${bool}, Type: ${typeof (bool === 'true')}`);
16  </script>
17  <!-- Code injected by live-server -->
18  <script>
19    // ... (DATA) ... For SVG support
20  </script>
```

Console output:

```
Number : 11, Type: number
String : HI, Type: string
Boolean : true, Type: boolean
```

Task5

top

127.0.0.1:5500

homeworks/ilyas

task5.html

MetaMask

```
7 <body>
8
9 <script>
10   let length = prompt("Enter the initial length:");
11   let width = prompt("Enter the initial width:");
12
13   function calculateArea(len, wid) {
14     return len * wid;
15   }
16
17   console.log('Initial Area: ${calculateArea(length, width)}');
18
19   const updateChoice = prompt("Do you want to update the length or width? (type 'length' or 'width')");
20
21   if (updateChoice === 'length') {
22     length = prompt("Enter the new length:");
23   } else if (updateChoice === 'width') {
24     width = prompt("Enter the new width:");
25   }
26
27   console.log('New Area: ${calculateArea(length, width)}');
28 </script>
29 <!-- Code injected by live-server -->
```

Выражения контрольного значения

Точки останова

Приостанавливать на необнаруженном

Приостанавливать на обнаруженном

Область действия

Не приостановлено

Стек вызовов

Не приостановлено

Точки останова для XMLHttpRequest/fetch

Точки останова DOM

Глобальные прослушатели

Точки останова прослушателя

Точки останова при нарушении CORS

Консоль

Что нового

Проблемы

Автозаполнение

Фильтр

Уровни по умолчанию

1 проблема

1 скрыто

Скрыть сеть

Сохранять журнал

Только выбранный контекст

Группировать в консоли похожие сообщения

Показывать ошибки CORS в консоли

Регистрировать запросы XMLHttpRequests

Энергичное вычисление

Автозаполнение из истории

Рассматривать оценку кода как пользовательское действие

Initial Area: 200

task5.html:16

New Area: 300

task5.html:26

Task6

127.0.0.1:5500

homeworks/ilyas

task6.html

MetaMask

```
6 <title>Color input</title>
7 </head>
8 <body>
9 <script>
10   const colorCount = prompt("How many colors?");
11   const colors = [];
12
13   let i = 0;
14   while(i < colorCount){
15     const color = prompt('Enter color ${i+1}');
16     colors.push(color);
17     i++;
18   }
19
20   console.log('First Color: ${colors[0]}');
21   console.log('Last Color: ${colors[colors.length - 1]}');
22 </script>
23 <!-- Code injected by live-server -->
24 <script>
25   // <![CDATA[ <-- For SVG support
26   if ('WebSocket' in window) {
```

Точки останова

Приостанавливать на необнаруженном

Приостанавливать на обнаруженном

Область действия

Не приостановлено

Стек вызовов

Не приостановлено

Точки останова для XMLHttpRequest/fetch

Точки останова DOM

Глобальные прослушатели

Точки останова прослушателя

Точки останова при нарушении CORS

Консоль

Что нового

Проблемы

Автозаполнение

Фильтр

Уровни по умолчанию

1 проблема

1 скрыто

Скрыть сеть

Сохранять журнал

Только выбранный контекст

Группировать в консоли похожие сообщения

Показывать ошибки CORS в консоли

Регистрировать запросы XMLHttpRequests

Энергичное вычисление

Автозаполнение из истории

Рассматривать оценку кода как пользовательское действие

First Color: red

task6.html:20

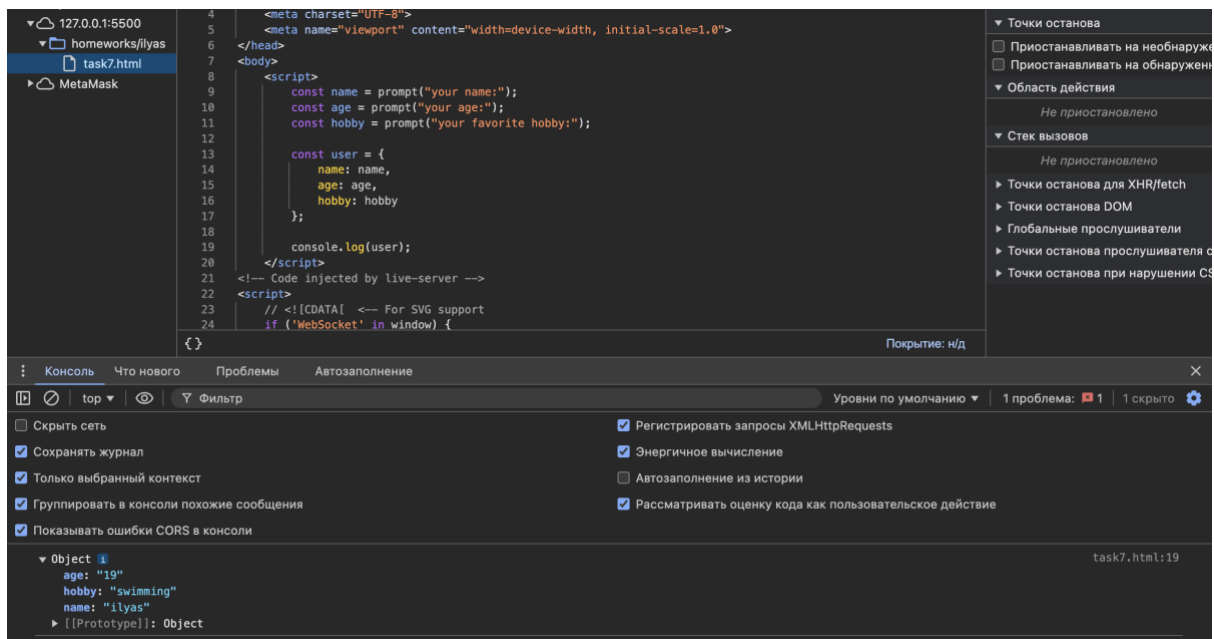
Last Color: yellow

task6.html:21

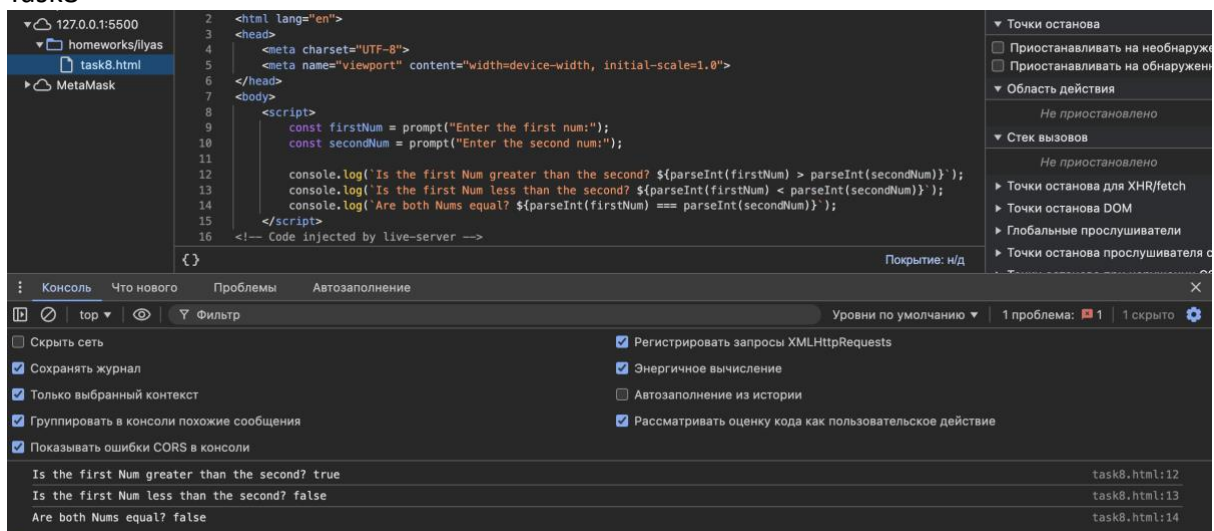
Live reload enabled.

task6.html:51

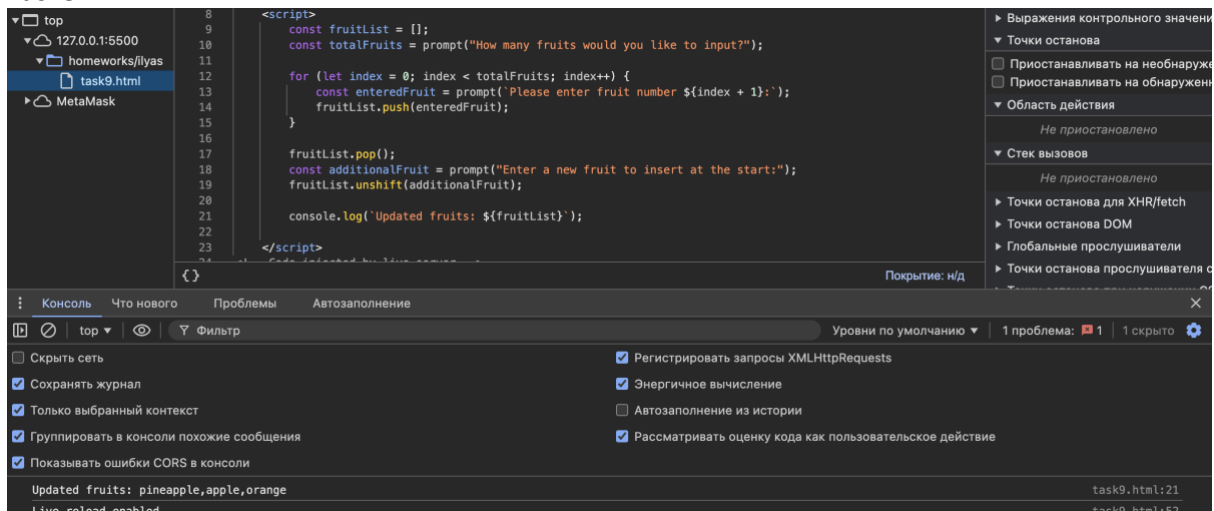
Task7



Task8



Task9



Task10

The screenshot displays a web browser's developer console with a JavaScript script for a simple calculator. The script is as follows:

```
<script>
8   const num1 = parseFloat(prompt("first number:"));
9   const num2 = parseFloat(prompt("second number:"));
10  const operation = prompt("Enter an operation (+, -, *, /):");
11
12
13
14
15  let result;
16
17  switch(operation) {
18      case "+":
19          result = num1 + num2;
20          break;
21      case "-":
22          result = num1 - num2;
23          break;
24      case "*":
25          result = num1 * num2;
26          break;
27      case "/":
28          if (num2 === 0) {
29              result = "Error: Division by zero";
30          } else {
31              result = num1 / num2;
32          }
33          break;
34      default:
35          result = "Invalid operation";
36  }
37
38  console.log('Result: ${result}');
39
40  </script>
```

The console output shows the result of the calculation: `Result: 30`. The right sidebar of the developer console is open, showing various debugging tools like 'Expressions', 'Breakpoints', and 'Call Stack'. The bottom status bar indicates the file `task10.html` is open at line 36.