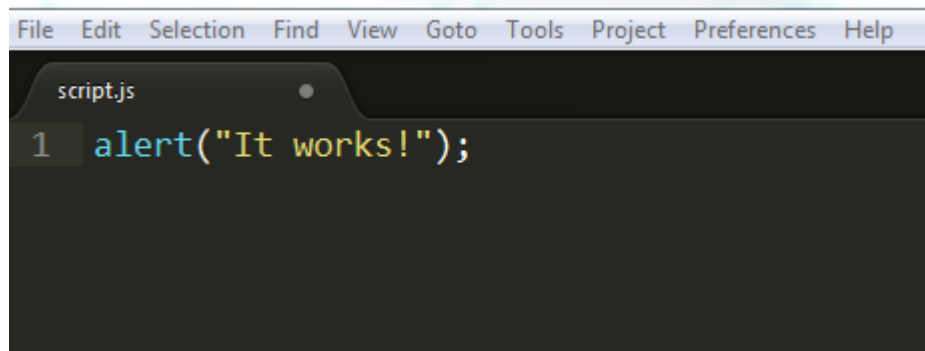


## JavaScript, базовый уровень.

Создадим пустую HTML страницу с подключенным скриптом:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3   <head>
4     <script src="script.js"></script>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6   </head>
7   <body>
8   </body>
9 </html>
```

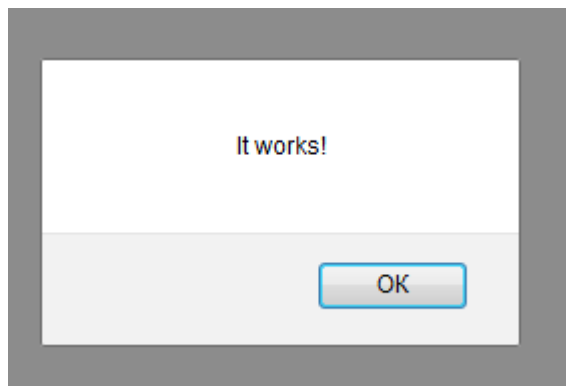
Создадим рядом со страницей файл script.js с одной строчкой:



The image shows a code editor window with a single file named 'script.js'. The editor has a menu bar with 'File', 'Edit', 'Selection', 'Find', 'View', 'Goto', 'Tools', 'Project', 'Preferences', and 'Help'. The code in the editor is:

```
1 alert("It works!");
```

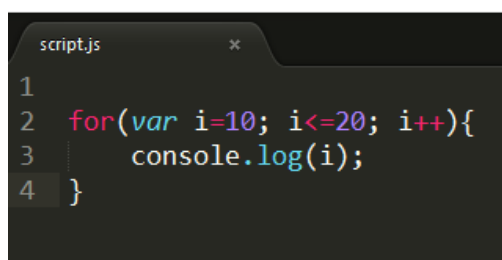
После чего откроем страницу в браузере и убедимся, что скрипт подключен:



Теперь откроем консоль браузера (клавиша F12, если вы пользуетесь Mozilla Firefox или Google Chrome). Мы будем выводить в лог результаты выполнения скриптов, после чего проверять правильность работы в этой консоли.

### Задание 1. Работа с консолью.

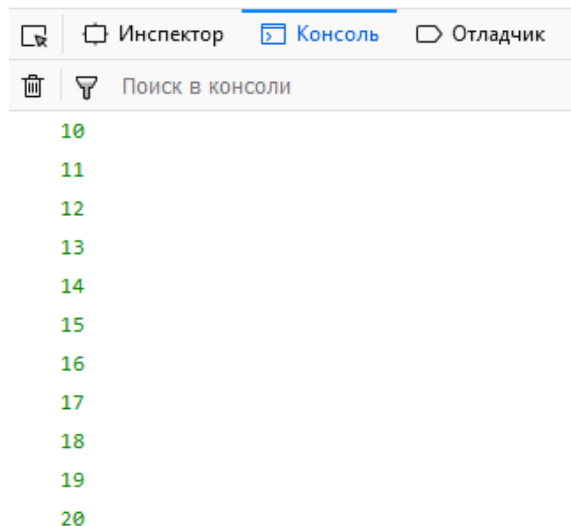
Вывести в консоль числа от 10 до 20 включительно.



The image shows a code editor window with a single file named 'script.js'. The code in the editor is:

```
1
2 for(var i=10; i<=20; i++){
3   console.log(i);
4 }
```

Смотрим, что в консоли браузера:



#### Задания для самостоятельного выполнения:

1. Доработайте скрипт таким образом, чтобы в консоль выводились квадраты чисел от 10 до 20.
2. Доработайте скрипт таким образом, чтобы в консоль выводились сумма всех чисел от 10 до 20.

#### Задание 2. Взаимодействие с HTML.

Добавим на страницу два поля ввода, кнопку и div для вывода результата:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3   <head>
4     <script src="script.js"></script>
5     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6   </head>
7   <body>
8     X1: <input type="text" id="x1"><br>
9     X2: <input type="text" id="x2"><br>
10    <input type="button" value="Run" onclick="buttonClick()"><br>
11    <div id="result"></div>
12  </body>
13 </html>
```

И добавим в скрипт функцию-обработчик клика по кнопке:

```
1 function buttonClick(){
2   var x1 = document.getElementById('x1').value;
3   var x2 = document.getElementById('x2').value;
4   var resultDiv = document.getElementById('result');
5   resultDiv.append("x1: "+x1 + ", x2: "+x2);
6 }
```

Здесь в первых двух строчках мы, используя id элементов, получаем текст, введенный в поля ввода. Далее получаем div-элемент, и в последней строке выводим туда значения.

Проверим, как это работает:

---

X1:

X2:

x1: 10, x2: 20

Теперь выведем сумму значений X1 и X2. Изменим скрипт следующим образом:

```
var resultDiv = document.getElementById('result');  
resultDiv.append("x1 + x2 = " + (x1+x2));  
}
```

Проверяем:

---

X1:

X2:

x1 + x2 = 1020

Совсем не тот результат, которого мы ожидали. Это произошло из-за того, что были сложены две строки «10» и «20». Нам надо получить из этих строк числа, прежде чем складывать. В JavaScript для этого существует метод `parseInt`:

```
1 function buttonClick(){  
2     var x1 = parseInt(document.getElementById('x1').value);  
3     var x2 = parseInt(document.getElementById('x2').value);  
4     var resultDiv = document.getElementById('result');  
5     resultDiv.append("x1 + x2 = " + (x1+x2));  
6 }
```

Проверяем:

---

X1:

X2:

x1 + x2 = 30

Выглядит неплохо, но у нас есть ещё проблемы в этом коде. Попробуйте ввести в поле не числовые значения:

---

X1:

X2:

x1 + x2 = NaN

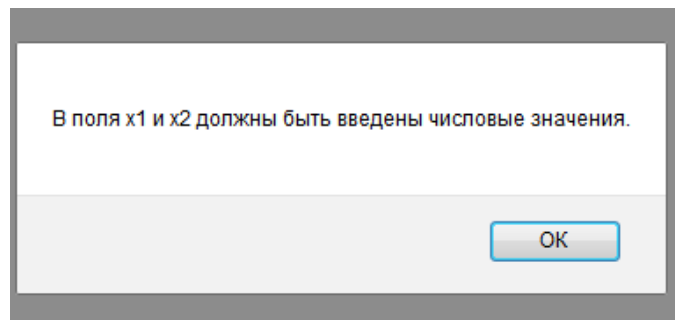
Метод `parseInt` не смог распарсить такую строку. Добавим проверку:

```

1  function buttonClick(){
2      var x1 = parseInt(document.getElementById('x1').value);
3      var x2 = parseInt(document.getElementById('x2').value);
4
5      if(Number.isNaN(x1) || Number.isNaN(x2)){
6          alert("В поля x1 и x2 должны быть введены числовые значения.");
7      } else {
8          var resultDiv = document.getElementById('result');
9          resultDiv.append("x1 + x2 = " + (x1 + x2));
10     }
11 }

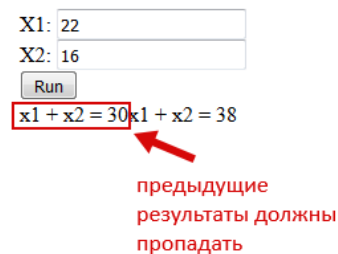
```

Теперь пользователь получает сообщение об ошибке:



#### Задания для самостоятельного выполнения.

1. Доработайте скрипт таким образом, чтобы после каждого нажатия кнопки Run поле вывода результата очищалось от предыдущих значений



2. Добавьте проверку на пустой ввод. Если пользователь не ввёл значения x1 или x2, выведите ошибку «Поля x1 и x2 должны быть заполнены».
3. Доработайте скрипт таким образом, чтобы после нажатия Run выводилась сумма всех чисел от x1 до x2.
4. Добавьте кнопку «Очистить», которая удаляет введённые в поля x1 и x2 значения:

X1:

X2:

5. Добавьте возможность подсчитывать не только сумму чисел от x1 до x2, но и их произведение. Для возможности выбора между этими двумя алгоритмами добавьте на форму два переключателя (<input type="radio">):

X1:

X2:

Summa всех чисел от x1 до x2

Произведение всех чисел от x1 до x2

Run

Очистить

6. **(повышенной сложности).** Реализуйте алгоритм нахождения простых чисел в промежутке от X1 до X2.

Натуральное число, большее 1, называется простым, если оно ни на что не делится, кроме себя и 1. Т.е. если  $x1=2$ , а  $x2=10$ , результат должен быть: 2, 3, 5, 7.

7. **(супер-повышенной сложности, специально для тех, кто уже знает JS, и кому было скучно при выполнении предыдущих заданий).** Напишите карточную игру “memory game”. В начале игры все карты выложены на стол рубашкой вверх. Игрок может открывать по 2 карты, если изображения на них одинаковые – карты исчезают, если разные – карты переворачиваются обратно через 1-2 секунды. Цель игры – последовательно открывая карты с одинаковыми изображениями, добиться исчезновения всей колоды.

Пример можно посмотреть здесь: <http://www.webgamesonline.com/memory/>