

Техническое задание

1. Мы бы хотели взглянуть на то, как вы верстаете - приведите, пожалуйста, две-три ссылки на код, возможно даже на работающие сайты, которые используют ваши шаблоны.
2. Если вы знакомы с любым JS-фреймворком или CSS-препроцессором или слышали о БЭМ, нам интересно об этом узнать. Перечислите все, что вспомните, с небольшими комментариями.
3. Если знаете любой язык программирования, кроме JavaScript, расскажите об этом вкратце.
4. Расскажите, что делает нижеприведенная функция. Для чего она может быть использована? Напишите пример использования.

```
function bind(method, context) {  
  var args = Array.prototype.slice.call(arguments, 2);  
  return function() {  
    var a = args.concat(Array.prototype.slice.call(arguments, 0));  
    return method.apply(context, a);  
  }  
}
```

5. Реализуйте функцию `inArray`, примеры вызова которой приведены ниже.

```
inArray(15, [1, 10, 145, 8]) === false;  
[23, 674, 4, 12].inArray(4) === true;
```

6. Реализуйте компонент “Фильтр” по нижеприведенному описанию. Ваше решение мы ожидаем увидеть в виде ссылки на публичный репозиторий на GitHub/GitLab/Bitbucket.

Спецификация

Компонент “фильтр” состоит из инпутов для ввода данных, кнопки **Add condition**, и кнопок **Apply** и **Clear filter**

The mockup shows a filter component with three input fields in a row: a dropdown menu with 'Text field' selected, another dropdown menu with 'Containing' selected, and an empty text input field. Below these fields is a blue button with a plus icon and the text '+ Add condition'. At the bottom of the component are two buttons: a blue 'Apply' button and a grey 'Clear filter' button.

При клике на первый инпут появляется выпадашка с возможностью выбора поля фильтрации

- Text field
- Number field

This mockup shows the first dropdown menu open, displaying two options: 'Text field' and 'Number field'. The 'Text field' option is highlighted with a blue background. The rest of the UI, including the 'Containing' dropdown and the empty text input field, remains the same as in the previous mockup.

При выборе второго инпута появятся выпадашка с возможностью выбора операции фильтрации. Операция зависит от выбранного поля фильтрации.

Для **Text field** фильтра

- Containing
- Exactly matching
- Begins with
- Ends with

Для **Number field** фильтра

- Equal
- Greater than
- Less than

The image shows two examples of a filter configuration interface. The top example is for a 'Text field' with the operation 'Containing'. A dropdown menu is open, showing options: 'Containing' (selected), 'Exactly matching', 'Begins with', and 'Ends with'. Below the dropdown are 'Apply' and 'Clear filter' buttons. The bottom example is for a 'Number field' with the operation 'Equal'. A dropdown menu is open, showing options: 'Equal' (selected), 'Greater than', and 'Less than'. Below the dropdown are 'Apply' and 'Clear filter' buttons. In both examples, there is a '+ Add condition' link and an input field for the filter value.

В последний инпут вводится значение фильтра. Для **Text field** фильтра это произвольная строка, для **Number field** это число.

The image shows the filter configuration interface with the filter value input field populated. The top example is for a 'Text field' with the operation 'Containing', and the input field contains 'some text'. The bottom example is for a 'Number field' with the operation 'Equal', and the input field contains '100'. Both examples have 'Apply' and 'Clear filter' buttons and a '+ Add condition' link.

При изменении поля фильтрации с **Text field** на **Number field** или наоборот, остальные инпуты (операция и значение фильтрации) сбрасываются на дефолтные.

При клике на **Add condition** появляется новая строка с инпутами. Дефолтное состояние новой строки фильтра - тип фильтра **Text**, операция **Containing**.

Максимальное количество строк - **10**.

Если строк больше одной, в конце каждой из них появляется кнопка **X** удаляющая данную строку.

The screenshot shows a filter configuration interface. It contains two rows of conditions. The first row has a dropdown menu set to 'Text field', a second dropdown menu set to 'Containing', an empty text input field, and a small 'X' button to its right. The second row has a dropdown menu set to 'Number field', a second dropdown menu set to 'Equal', an empty text input field, and a small 'X' button to its right. Below these rows is a blue link with a plus icon and the text '+ Add condition'. At the bottom of the interface are two buttons: a blue 'Apply' button and a light gray 'Clear filter' button.

При нажатии кнопки **Clear filter** весь компонент скидывается в первоначальное состояние.

This screenshot shows the filter configuration interface in its initial state. It features a single row with a dropdown menu set to 'Text field', a second dropdown menu set to 'Containing', and an empty text input field. Below this row is a blue link with a plus icon and the text '+ Add condition'. At the bottom are two buttons: a blue 'Apply' button and a light gray 'Clear filter' button.

При нажатии кнопки **Apply** на экран или в консоль браузера выводится информация о текущем состоянии фильтра (примеры ниже).

Примеры:

Для фильтра

Text field ▼	Containing ▼	some text	×
Number field ▼	Greater than ▼	1000	×
Text field ▼	Begins with ▼	word	×

[+ Add condition](#)

Apply

Clear filter

Вывод:

```
{
  text: [
    { operation: 'containing', value: 'some text' },
    { operation: 'begins with', value: 'word' },
  ],
  number: [
    { operation: 'greater than', value: 1000 },
  ],
}
```

Для фильтра

Number field ▼	Less than ▼	500
----------------	-------------	-----

[+ Add condition](#)

Apply

Clear filter

Вывод:

```
{
  text: [],
  number: [
    { operation: 'less than', value: 500 },
  ],
}
```