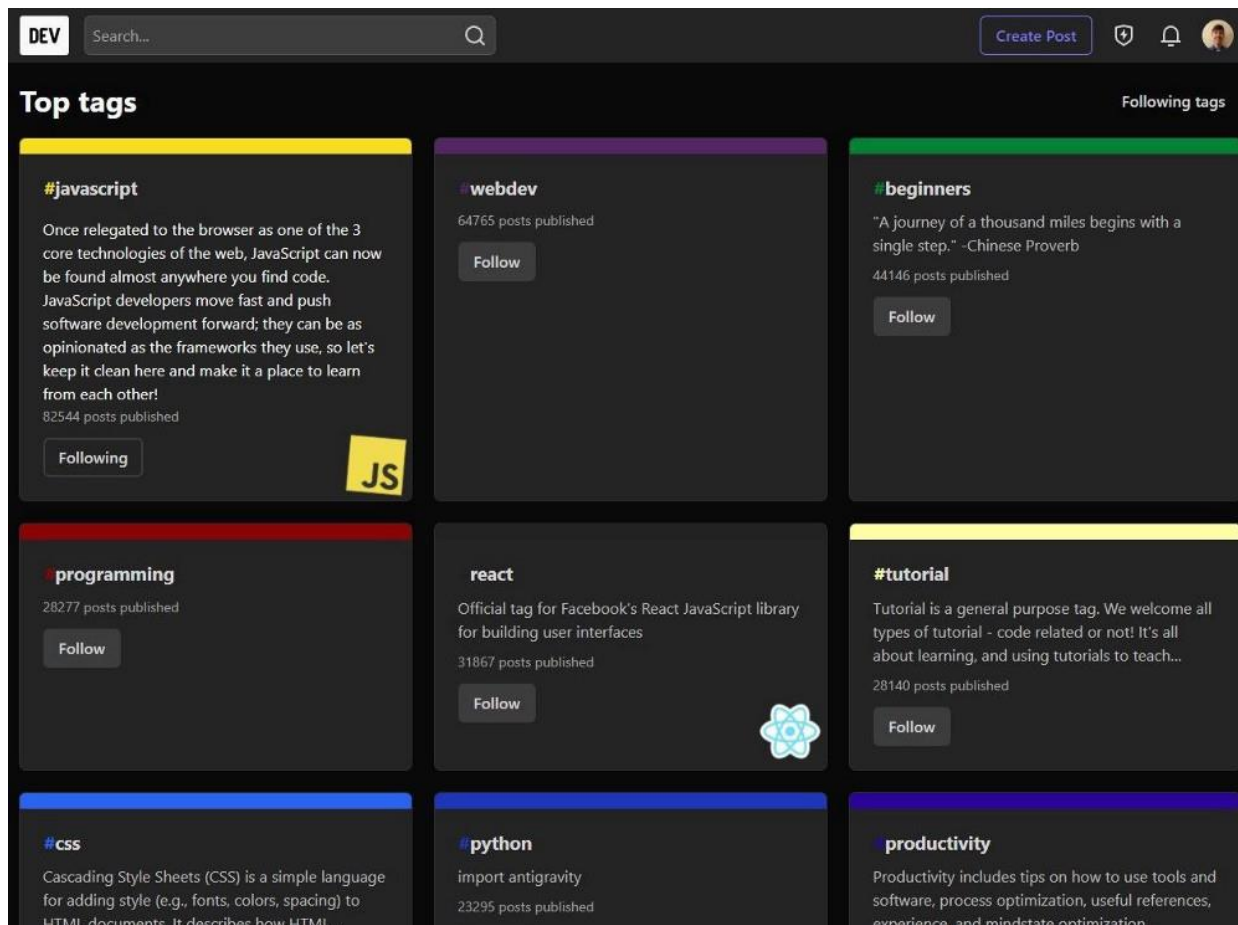


Как использовать JavaScript для расстановки элементов на веб-странице

В этом руководстве я расскажу поэтапно, как это сделать на примере DEV:



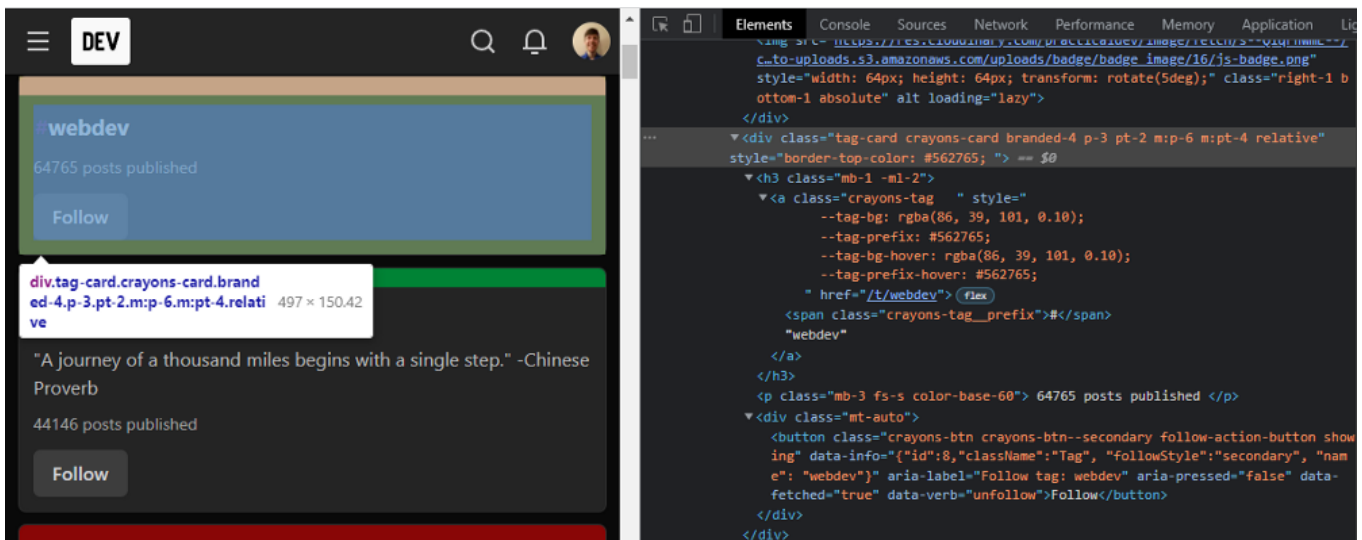
Скриншот страницы “Топ тегов” на DEV, где теги, похоже, расставлены без логики

Руководство

Перейдите на страницу, которую хотите упорядочить, откройте DevTools (F12) и приготовьтесь исследовать ее DOM структуру.

1 . Определите структуру элемента

Чтобы отсортировать все данные, нужно понимать, как устроен каждый интересующий вас элемент. Для этого выберите его с помощью кнопки Inspect (или Ctrl + Shift + C) и рекурсивно раскройте.



2. Определитесь, каким образом будут отсортированы элементы

Хотите упорядочить DIV по заголовку или, может быть, по числовому параметру?

В нашем случае будем сортировать их по количеству опубликованных постов. Если мы напишем в консоли `$0.querySelector('p.color-base-60').innerText`, то получим 64765 опубликованных статей.

3. Определите порядок

Нужен порядок по возрастанию или убыванию?

```
const ascendingOrder = false;
```

4. Выберите все элементы

Можете найти селектор, чтобы объединить все объекты для сортировки? В нашем случае, все элементы имеют tag-card класс.

```
const elements = [...document.querySelectorAll('.tag-card')];
```

5. Найти исходный узел всех элементов

Мы можем написать еще один селектор, чтобы найти DIV, который содержит все дочерние элементы, или же выбрать путь наименьшего сопротивления и уточнить родителя первого элемента.

```
const parentElement = document.querySelector('.grid');
```

```
// либо
```

```
const parentElement = elements[0].parentNode;
```

Мы используем исходный узел, чтобы переставить местами дочерние элементы с сохранением его структуры.

6. Сортировка элементов

Теперь самое время собрать все скрипты в один и запустить их для нашей страницы:

```
// 2 — Определить селектор

const selector = element => element.querySelector('p.color-base-60').innerText

// 3 — Выбрать необходимый порядок

const ascendingOrder = false;

const isNumeric = true;

// 4 — Выбрать все элементы

const elements = [...document.querySelectorAll('.tag-card')];

// 5 — Найти исходный узел

const parentElement = elements[0].parentNode;

// 6 — Рассортировать элементы

const collator = new Intl.Collator(undefined, {numeric: isNumeric, sensitivity: 'base'});

elements

.sort((elementA, elementB) => {

const [firstElement, secondElement] = ascendingOrder ? [elementA, elementB] : [elementB, elementA];

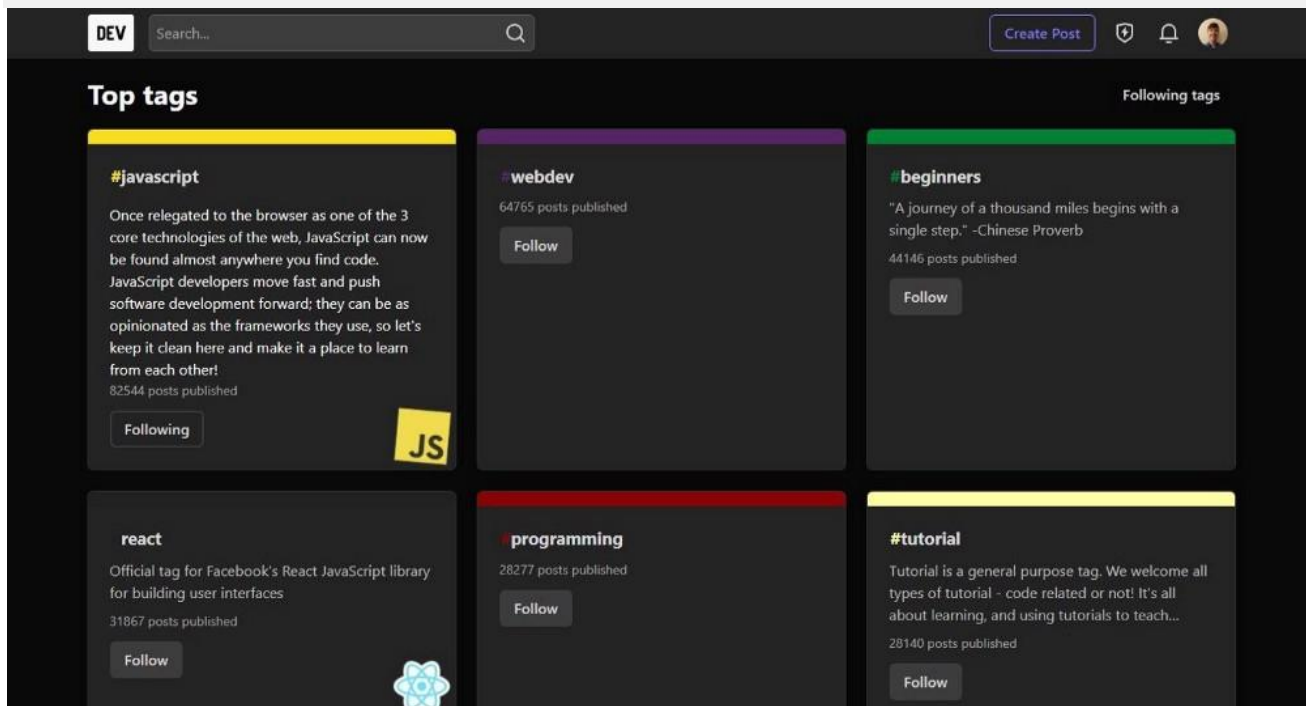
const textOfFirstElement = selector(firstElement);

const textOfSecondElement = selector(secondElement);

return collator.compare(textOfFirstElement , textOfSecondElement)
```

```
}}
```

```
.forEach(element => parentElement.appendChild(element));
```



Теперь теги упорядочены по количеству опубликованных постов!