КАК СТАТЬ АВТОРОМ

marus_space 1 июня в 18:09

История одного макета: способы сделать сайт послушным

Блог компании АО «ГНИВЦ», Разработка веб-сайтов*, CSS*, HTML*

Вёрстка помогает нам выстраивать содержимое веб-страниц по определённым правилам: например, строго в соответствии с согласованным макетом или в зависимости от пользовательского устройства. Сегодня сайты неплохо умеют подстраивать свой контент и под различные размеры экранов, но так было не всегда.

Существует несколько основных подходов к вёрстке, особенности которых мы рассмотрим в этой статье на простом наглядном примере:

```
<body>
  <header>
    <!-- Содержимое заголовка -->
  </header>
  <main>
    <div class="container">
      <div class="card">
        <!-- Содержимое карточки -->
      </div>
```

```
      <!-- Ещё 5 таких же карточек -->
    </div>
  </main>
</body>
```

Структура представленной разметки страницы очень простая: сверху располагается шапка, ниже — основное содержимое страницы, которое включает в себя контейнер для карточек и сами карточки.

Для демонстрации подходов к вёрстке мы будем вносить изменения только в CSS, в то время как представленная выше HTML-разметка будет оставаться неизменной.

А теперь давайте вернёмся к обсуждению самих подходов.

Фиксированная вёрстка

Самым простым и не требующим больших усилий решением можно считать *фиксированную вёрстку*. Фиксированной она называется потому, что содержимое страницы, свёрстанной таким образом, никак не подстраивается под размер экрана и отображается на всех устройствах одинаково.

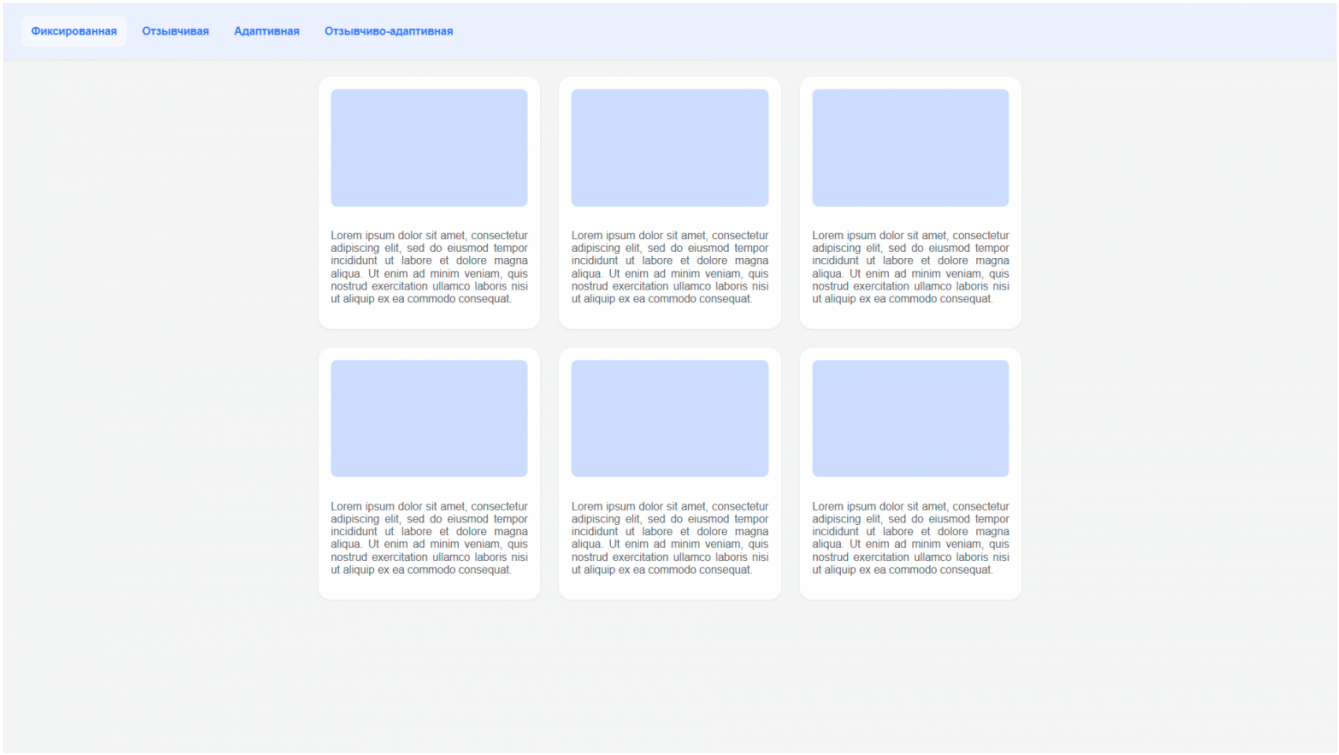
Для фиксированной вёрстки ширина контейнера, в котором находится весь контент, указывается в абсолютных величинах (например, `px`). Помимо ширины контейнера не лишним будет указать внешние отступы, чтобы контент отображался всегда по центру экрана.

Стили, указанные после отступа, нужны для формирования сетки контента, которая будет иметь три колонки одинаковой ширины и отступы в `24px` между колонками и строками. Сетка задаётся с помощью CSS Grid Layout, и если вы ещё не знакомы с возможностями этого способа раскладки, рекомендую ознакомиться с данной статьёй.

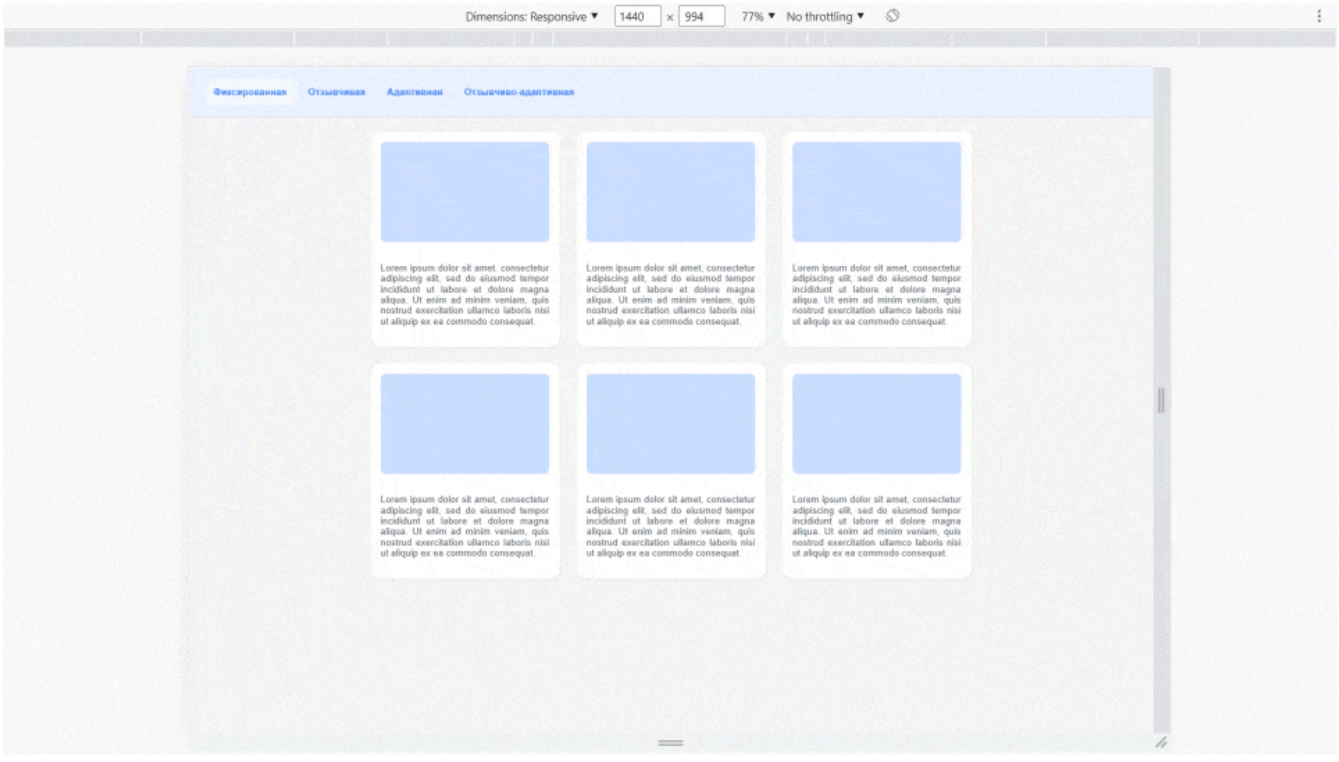
```
.container {
  width: 900px;
  margin: 24px auto;

  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 24px;
}
```

Этого уже будет достаточно, чтобы добиться ожидаемого отображения веб-сайта на обычном экране компьютера.



Однако, такая вёрстка совершенно не подходит для отображения на мобильных устройствах: из-за фиксированной ширины контейнера пользователю придётся приближать страницу и скроллить её влево-вправо, чтобы иметь возможность прочитать имеющийся на ней текст.



Отзывчивая вёрстка

В таком случае мы могли бы сделать наш интерфейс чуть более дружелюбным, если бы стали использовать *отзывчивую* (или *резиновую*) *вёрстку*. Её особенность состоит в том, что для задания параметров используются не абсолютные, а относительные величины (например, `%`), благодаря чему содержимое страницы может подстраиваться под ширину экрана.

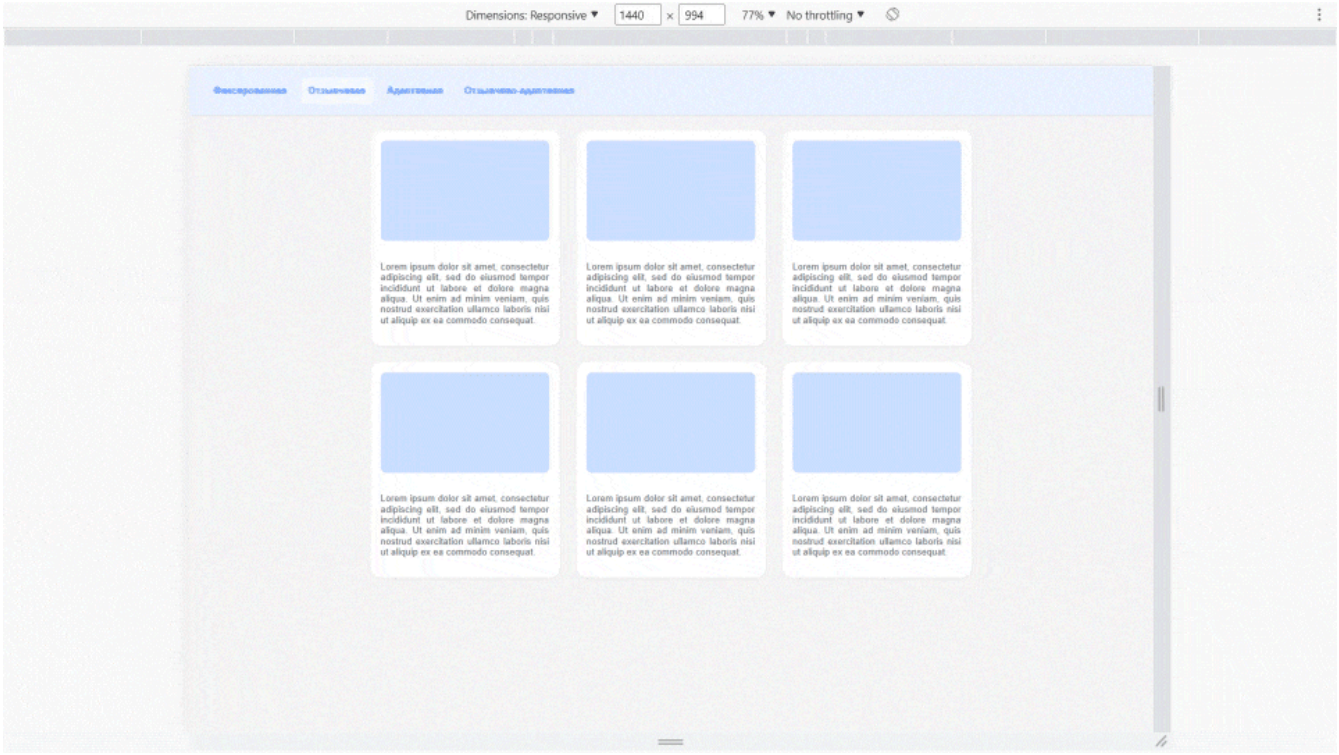
Для нашего примера также необходимо указать максимальную ширину контейнера, чтобы на больших экранах контент отображался точно так же, как и в случае с фиксированной вёрсткой.

Ещё мы добавим внутренний отступ, чтобы на маленьком экране карточки не прилипали к его краям. Именно из-за этих внутренних отступов в `16px` слева и справа ограничения по максимальной ширине увеличиваются с `900px` до `932px`.

```
.container {
  width: 100%;
  max-width: 932px;
  padding: 0 16px;
  margin: 24px auto;

  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 24px;
}
```

Вот как поведёт себя резиновая вёрстка на средних и маленьких экранах. На планшете она отображается так же хорошо, как и на экране компьютера, но на мобильных устройствах ситуация только ухудшилась: карточки стали настолько узкими, что прочитать текст на них практически невозможно.



Адаптивная вёрстка

С ростом популярности смартфонов появилась и стала распространённой *адаптивная вёрстка*, которая позволила изменять стили контейнера в зависимости от ширины экрана. Этот подход может помочь нам избежать проблем, с которыми мы столкнулись выше, когда использовали резиновую вёрстку на маленьком экране.

С помощью медиа-запросов мы можем изменять сетку, по которой будет выстраиваться контент. Изначально стили указываются для минимальных размеров экрана (такой подход называется mobile-first), и если ширина экрана достигает какого-либо брейкпоинта, первоначальные стили заменяются новыми. Существует множество систем величин брейкпоинтов, но мы воспользуемся значениями, представленными здесь.

В нашем примере мы будем изменять максимальную ширину контейнера и сетку для расположения карточек: на маленьких экранах (шириной меньше `480px`) колонка будет одна, на экранах средней величины (от `480px` до `768px`) — две, а на больших (шириной больше `768px`) — три. Максимально возможная ширина контейнера по-прежнему `932px`, она будет установлена на экранах, ширина которых равна этому значению или больше него.

```
.container {
  max-width: 300px;
  padding: 0 16px;
```

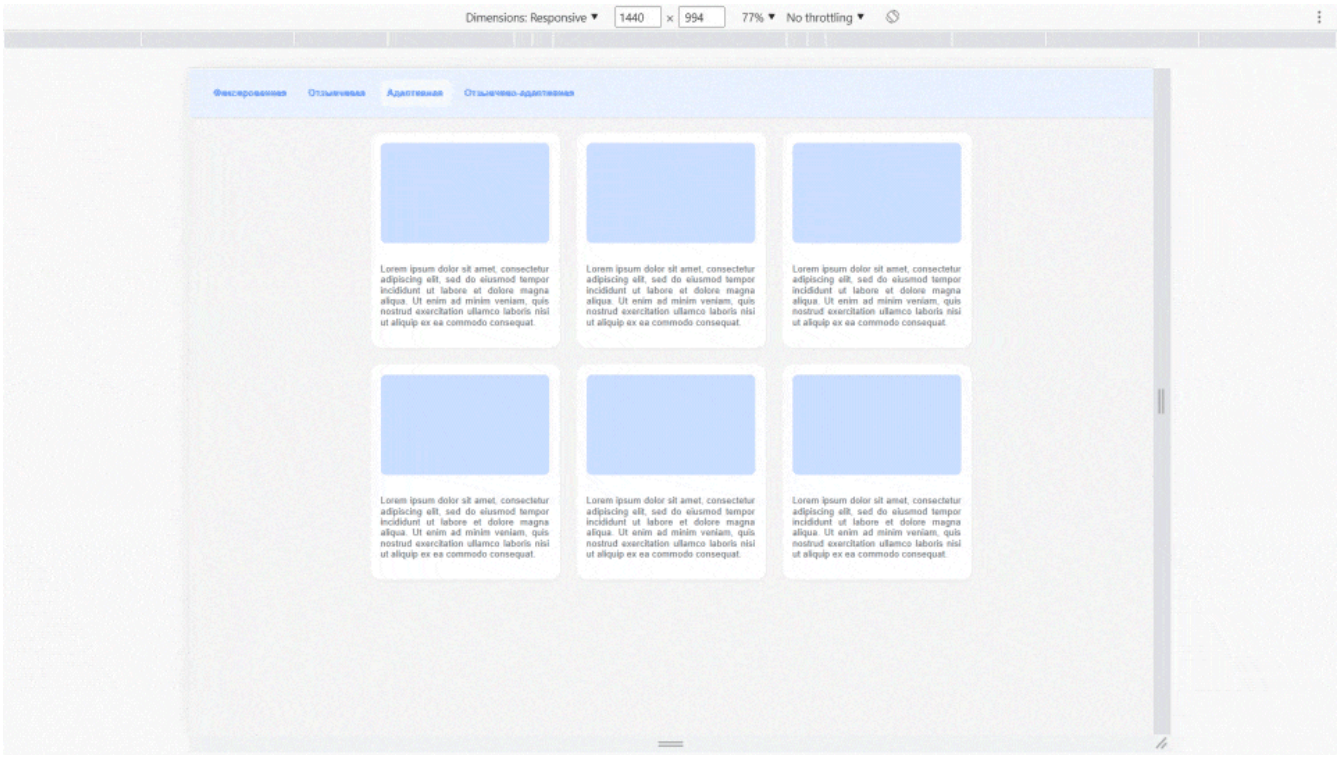
```
  margin: 24px auto;

  display: grid;
  grid-template-columns: repeat(1, 1fr);
  gap: 24px;
}

@media screen and (min-width: 480px) {
  .container {
    max-width: 480px;
    grid-template-columns: repeat(2, 1fr);
  }
}

@media screen and (min-width: 768px) {
  .container {
    max-width: 768px;
    grid-template-columns: repeat(3, 1fr);
  }
}

@media screen and (min-width: 932px) {
  .container {
    max-width: 932px;
  }
}
```



Отзывчиво-адаптивная вёрстка

Последним и самым современным подходом к вёрстке веб-сайтов является *отзывчиво-адаптивная вёрстка*, которая сочетает в себе положительные свойства двух рассмотренных выше подходов. Такая вёрстка подойдёт в том случае, когда приоритетная задача состоит в том, чтобы занять максимум от имеющегося пространства: например, если контент представлен не карточками, а изображениями, и нам не нужны поля слева и справа от контейнера.

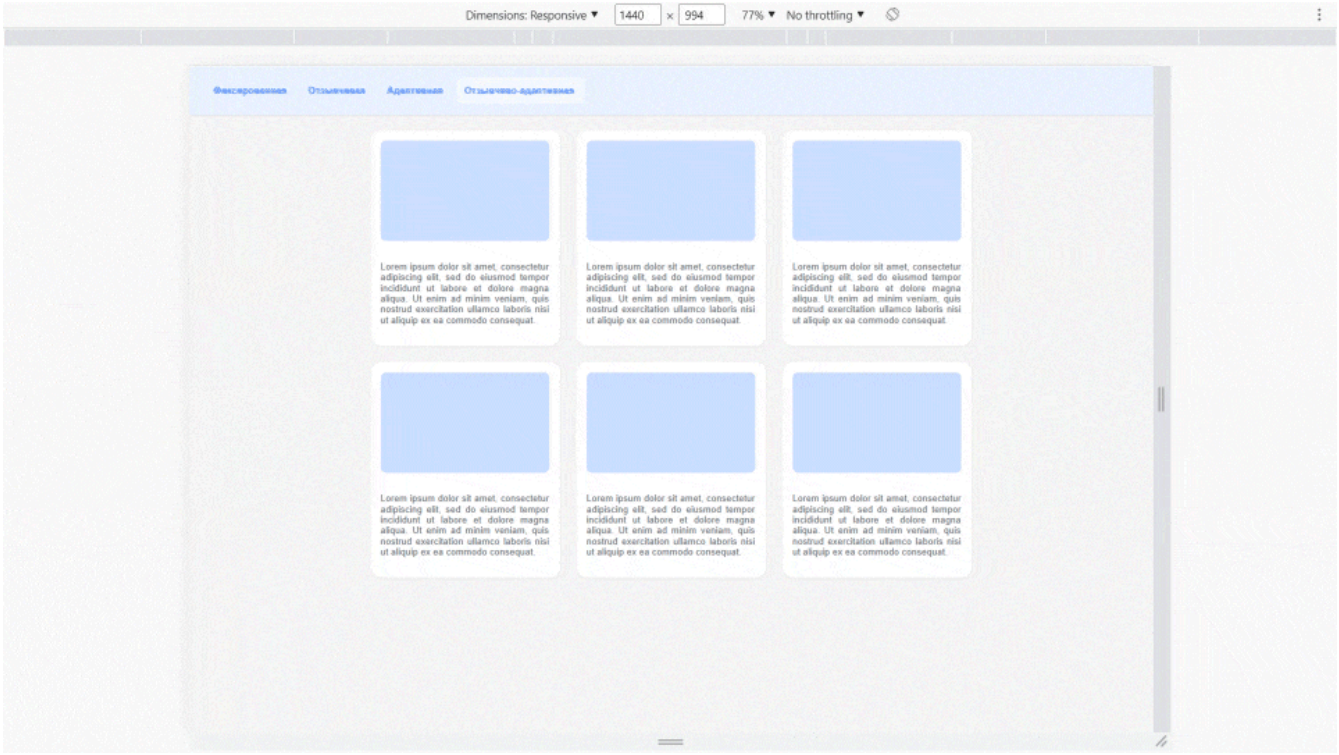
При таком подходе вёрстка между брейкпоинтами ведёт себя абсолютно «резиново», но на брейкпоинтах могут меняться какие-то важные свойства отображения (в нашем случае — свойства сетки). Используя данный подход, вы можете быть уверены, что на экране абсолютно любого размера ваше приложение будет отображаться как нужно и им будет удобно пользоваться.

```
.container {
  width: 100%;
  max-width: 932px;
  padding: 0 16px;
  margin: 24px auto;

  display: grid;
  grid-template-columns: repeat(1, 1fr);
  gap: 24px;
}

@media screen and (min-width: 480px) {
  .container {
    grid-template-columns: repeat(2, 1fr);
  }
}

@media screen and (min-width: 768px) {
  .container {
    grid-template-columns: repeat(3, 1fr);
  }
}
```



Последний рассмотренный подход к вёрстке считается самым надёжным и безопасным решением, однако, выбор типа вёрстки должен производиться индивидуально в зависимости от задач и структуры конкретного веб-приложения.

Код всех рассмотренных примеров вы можете посмотреть в песочнице.

В этой статье мы рассмотрели типы вёрстки, отличающиеся поведением на экранах разных размеров, но это лишь один взгляд на вёрстку. Существуют также иные подходы к выделению её типов, например, в зависимости от используемых HTML-тегов, но об этом мы поговорим уже в следующих статьях.

Теги: вёрстка, верстка, адаптивность, адаптивная вёрстка, резиновая вёрстка, фиксированная вёрстка

Хабы: Блог компании АО «ГНИВЦ», Разработка веб-сайтов, CSS, HTML

+4 3.3K 48

Редакторский дайджест
Присылаем лучшие статьи раз в месяц

Электропочта

АО «ГНИВЦ»

Главный научный инновационный внедренческий центр

Сайт

2 Карма -0.5 Рейтинг

Мария Гопкало @marus_space

React Frontend Developer

Комментарии 3

antoscenco-vladimir
01.06.2022 в 20:23

Или юзай bootstrap)

0 Ответить

dom1n1k
02.06.2022 в 02:27

Это какая-то самовыдуманная терминология.

На самом деле:
1. Фиксированная.
2. Резиновая — лейаут механически тянется в ширину, не меняя порядка/конструкции блоков. На западе это называлось Liquid/Fluid/Elastic. Строго говоря, эти три слова не синонимы, между ними были минорные отличия, но сейчас уже всем плевать.
3. Адаптивная (Adaptive) — лейаут тянется и перестаивается, но дискретно, под несколько фиксированных размеров.
4. Отзывчивая (Responsive) — лейаут тянется и перестаивается непрерывно для любой ширины экрана.

Это как правильно. На практике на сегодняшний день всех интересует только отзывчивая верстка (все прочие морально устарели), но в обиходе её часто называют адаптивной. Наверное, потому что слово проще. Такие дела.

0 Ответить

khegay
02.06.2022 в 14:24

Тот же бутстрап называет себя Responsive, но по факту – работает на брейкпоинтах и является адаптивным

0 Ответить

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

ВАКАНСИИ КОМПАНИИ «АО «ГНИВЦ»»

Разработчик
от 180 000 ₽ · АО «ГНИВЦ» · Нижний Новгород · Можно удаленно

Разработчик Hadoop
АО «ГНИВЦ» · Можно удаленно

Больше вакансий на Хабр Карьере

ЛУЧШИЕ ПУБЛИКАЦИИ ЗА СУТКИ

вчера в 20:54
Самого быстрого GIF не существует
+75 8.6K 28 26 +26

вчера в 16:00
В Data Science не нужна математика (Почти)
+63 14K 157 42 +42

вчера в 19:01
Возим «ложкой» по ковшу жидкого чугуна и снимаем «шлакопенку»
+56 4.6K 10 19 +19

вчера в 14:03

Настройка языка

Техническая поддержка

Полная версия

Вернуться на старую версию

© 2006–2022, Habr