



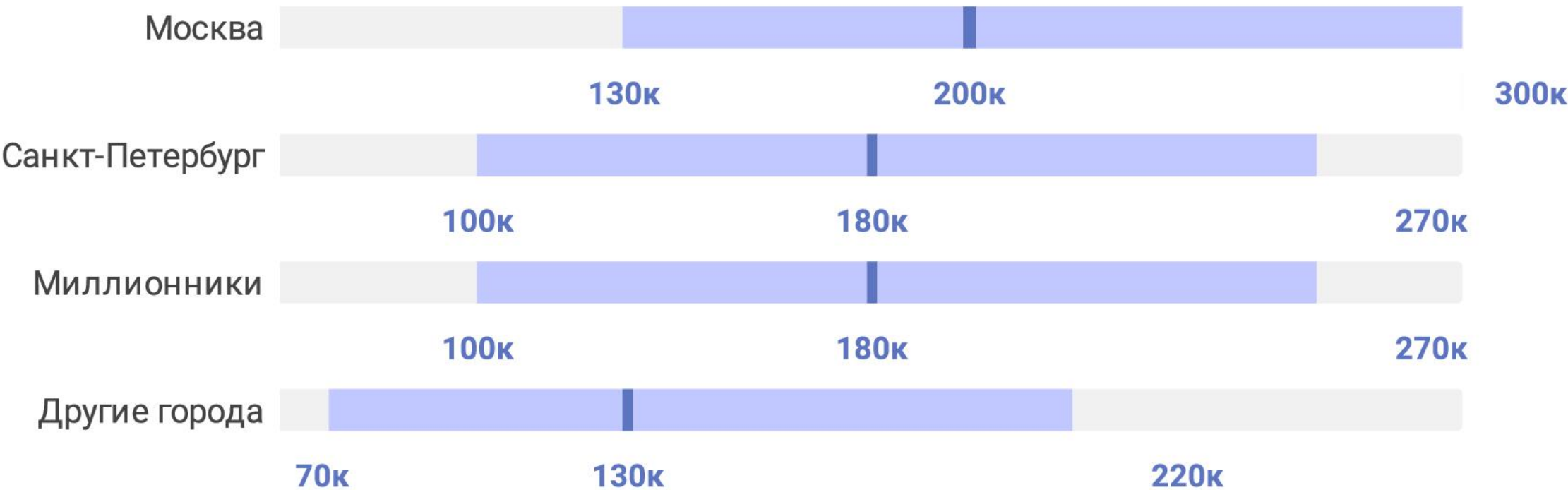
**Частное учреждение профессионального образования
«Высшая школа предпринимательства (колледж)»
(ЧУПО «ВШП»)**

Кафедра информационных технологий

**Инструментальные средства
разработки программного
обеспечения**

Зарплаты разработчиков в Москве, Санкт-Петербурге, городах-миллионниках и региональных городах России

за период 1 июля — 31 октября 2023 (на 10.11.2023), по данным калькулятора зарплат Хабр Карьеры



Как считаем?

- **Индекс стоимости жизни и аренды жилья**

Индексы остальных городов мы считали относительно Москвы с помощью сравнения городов на сервисе [Numbeo](#). Например, чтобы вычислить индекс Воронежа, мы взяли разницу в стоимости жизни+стоимости жилья и вычли ее из индекса Москвы ($100 - 35,1 = 64,9$).

- **Индекс зарплаты**

Вместе с этим мы добавили индекс зарплаты по принципу, по которому считали индексы из данных Numbeo. Мы вычли разницу зарплаты каждого города в процентах из Московской зарплаты, то есть медианная зарплата Москвы у нас 200 000 — берем ее за 100%, а зарплата Санкт-Петербурга 180 000 — $x\%$, вычислим, что 180 000 на 10% меньше 200 000, тогда $x=100-10=90\%$.

Индекс обеспеченности разработчиков в российских городах

	Индекс обеспеченности	Медианная зарплата разработчика*	Индекс стоимости жизни и жилья**	Индекс зарплаты**
Краснодар	123,1	168 000 ₽	68,2	84,0
Новосибирск	117,8	173 000 ₽	73,4	86,5
Пермь	117,0	150 000 ₽	64,1	75,0
Воронеж	115,5	150 000 ₽	64,9	75,0
Екатеринбург	114,1	170 000 ₽	74,5	85,0
Нижний Новгород	112,8	152 000 ₽	67,4	76,0
Калининград	111,9	151 500 ₽	67,7	75,8
Самара	111,7	150 000 ₽	67,1	75,0
Ростов-на-Дону	110,8	150 000 ₽	67,7	75,0
Казань	104,9	150 000 ₽	71,5	75,0
Санкт-Петербург	104,6	180 000 ₽	86,0	90,0
Москва	100,0	200 000 ₽	100,0	100,0
Уфа	90,9	120 000 ₽	66,0	60,0
Владивосток	66,5	125 000 ₽	94,0	62,5

Хард-скиллы	Софт-скиллы
<p data-bbox="135 254 952 304">Знание языков программирования</p> <p data-bbox="135 404 817 454">Умение писать понятный код</p> <p data-bbox="135 554 952 682">Понимание алгоритмов и структур данных</p> <p data-bbox="135 782 952 832">Умение работать с базами данных</p> <p data-bbox="135 932 927 982">Умение работать с библиотеками</p> <p data-bbox="135 1082 703 1132">Математические знания</p> <p data-bbox="135 1232 1003 1282">Умение эффективно отлаживать код</p>	<p data-bbox="1197 254 1768 304">Критическое мышление</p> <p data-bbox="1197 404 2079 532">Способность к обучению, особенно к самообучению</p> <p data-bbox="1197 632 1811 682">Умение вести переговоры</p> <p data-bbox="1197 782 1844 832">Эмоциональный интеллект</p> <p data-bbox="1197 932 1849 982">Навыки тайм-менеджмента</p> <p data-bbox="1197 1082 1926 1132">Самоанализ и саморефлексия</p> <p data-bbox="1197 1232 2061 1282">Работа в режиме неопределённости</p>

Повторим основные команды Git

git status

git add

git commit

git branch

git checkout

git clone

git push

git pull

Типичная работа с Гитхабом

- 1.Форк оригинального удаленного репозитория
- 2.Клонируем локально форк
- 3.Создаем свою ветку для работы
- 4.Работаем, коммитим
- 5.Пушим изменения в форк
- 6.Делаем пуллреквест из форка в оригинал
- 7.* Подтягиваем изменения из оригинального удаленного репозитория в локальный репозиторий

Как выложить сайт на GitHub Pages?

ДЗ-2

форков - 25

реквестов - 21

Вопросы по ДЗ-2?

Домашнее задание 2

1. Форк оригинального удаленного репозитория
2. Клонировать локально форк
3. Создаем свою ветку для работы
4. Создать файл `task2.md`
5. Заполнить его контентом (хобби, увлечение, информация):
 - Заголовки (разных уровней)
 - Выделение текста (жирный, курсив)
 - Списки (нумерованные и простые)
 - Ссылки
 - Картинки
6. Опубликовать в Github Pages
7. Сделать пулл-реквест с ссылкой на опубликованную страничку

Домашнее задание 3

1. Подтягиваем изменения из оригинального удаленного репозитория в локальный репозиторий
2. Создаем новую ветку для работы
3. Создать папку
4. Создаем в ней файл index.html
5. Заполнить его контентом (на основе предыдущего задания):
 - Заголовки (разных уровней)
 - Выделение текста (жирный, курсив)
 - Списки (нумерованные и простые)
 - Ссылки
 - Картинки
6. Опубликовать в Github Pages
7. Сделать пулл-реквест с ссылкой на опубликованную страничку

Дедлайн

08.12.2023 23:59:59

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <title>Заголовок страницы</title>
  </head>
  <body>
    <header class="main-header">
      <!-- Шапка сайта -->
    </header>
    <main>
      <!-- Основное содержимое страницы -->
    </main>
    <footer class="main-footer">
      <!-- Подвал сайта -->
    </footer>
  </body>
</html>
```

<h1> - <h6>

<i>

<p>

<a>

<article>

<section>

<nav>

<div>

<table>

Тег **<table>** используется для отображения данных в форме таблицы. Он позволяет создавать строки и столбцы, которые можно легко организовать и стилизовать.

Ограничения

таблицы сложно сделать отзывчивыми на небольших экранах, они могут вызывать горизонтальную прокрутку;

таблицы не следует использовать для вёрстки, так как они могут повлиять на доступность и структуру страницы.

<tr>

Тег **<tr>** создаёт в таблице строку.

<td>

Тег **<td>** используется для создания ячеек таблицы, он всегда вкладывается в элемент **<tr>**.

Ячейка может содержать текст, изображения, формы, список и другие HTML-элементы.

В одной строке таблицы должно быть одинаковое количество ячеек. Если содержимое в ячейке не помещается, браузер автоматически изменит её размеры, а если в ячейке нет содержимого, она всё равно появится, но с нулевой высотой и шириной.

```
<table>
  <tr>
    <td></td>
    <td>Продукт 1</td>
    <td>10.00</td>
  </tr>
  <tr>
    <td></td>
    <td>Продукт 2</td>
    <td>15.00</td>
  </tr>
</table>
```

<form>

Тег **<form>** используется для создания HTML-формы. В нём находится всё содержимое формы: поля для ввода, подписи к этим полям и кнопка отправки.

```
<form action='/login' method='POST'>  
  <label for='username'>Имя пользователя:</label>  
  <input type='text' id='username' name='username'><br>  
  <label for='password'>Пароль:</label>  
  <input type='password' id='password' name='password'><br>  
  <input type='submit' value='Войти'>  
</form>
```

Здесь **action** определяет URL, куда будут отправлены данные формы, а **method** указывает метод отправки данных: GET или POST. Если использовать метод GET, параметры формы будут отображаться в адресной строке браузера.

Это небезопасно, так как данные формы можно легко изменить.

<form>

Атрибуты тега **<form>**:

action — адрес, на который будут отправлены данные формы.

method — метод отправки данных: GET или POST.

target — окно или фрейм, в котором будет открыт результат отправки данных формы.

name — имя формы, которое используется для её идентификации при отправке данных на сервер.

autocomplete — может ли браузер запоминать введённые пользователем данные для автозаполнения.

enctype — способ кодирования данных формы при отправке на сервер.

novalidate — указывает, что данные формы не нужно проверять на корректность перед отправкой на сервер.

<input>

Когда пользователю нужно что-то ввести или выбрать на странице, скорее всего, вы будете использовать **<input>**. Тег **<input>** настраивается с помощью атрибута **type**.

text и **password**. Это, наверное, самые часто используемые типы полей ввода. Это простые текстовые поля. **password** отличается тем, что скрывает введенные символы.

```
<input type="text" name="username" placeholder="Имя пользователя">  
<input type="password" name="password" placeholder="Пароль">
```

<input>

radio и **checkbox**. С помощью **radio** пользователи выбирают один вариант из нескольких, а с помощью **checkbox** — ставят галочку, это может быть как одиночный выбор, так и множественный. Очень важно, чтобы у радиокнопок было одинаковое имя (`name`), чтобы группировать их.

`<input type="radio" name="animal" value="cat">` Кошка

`<input type="radio" name="animal" value="dog">` Собака

`<input type="checkbox" name="subscribe">` Подписаться на новости

submit — кнопка для отправки формы. Обычно ставится в конце формы. Хотя на практике часто используют `<button type="submit">`, так как кнопку проще стилизовать.

`<input type="submit" value="Отправить">`

<input>

Важные атрибуты

С помощью атрибута **name** сервер определяет, какие именно данные пришли.

placeholder полезен для коротких подсказок. Но это не замена для **<label>**. Всегда используйте подписи у полей ввода!

Если поле обязательно для заполнения, добавьте атрибут **required**.

Атрибут **value** устанавливает начальное значение поля.

<label>

Тег **<label>** связывает подпись элемента формы (**<input>**, **<textarea>**, **<select>**) с самим элементом. Тег также улучшает доступность формы для пользователей, использующих программы чтения с экрана.

Использовать тег можно двумя способами:

1. Добавить рядом с элементом формы. В таком случае **<label>** добавляется атрибут **for**, значением которого будет идентификатор соответствующего элемента формы.
2. Обернуть элемент формы в тег **<label>**.

<label>

Первый способ:

```
<label for="input-id">Метка</label>  
<input type="text" id="input-id">
```

Второй способ:

```
<label>Метка<input type="text"></label>
```

Атрибут **for** — указывает, к какому элементу формы привязана подпись.

<select>

Тег **<select>** используется для создания выпадающего списка. Этот тег содержит список опций, которые пользователь может выбрать. Когда пользователь выбирает опцию, выбранное значение отправляется на сервер или обрабатывается на клиентской стороне.

Все опции списка должны быть обёрнуты в тег **<option>**.

```
<select multiple>  
  <option value='apple'>Яблоко</option>  
  <option value='banana'>Банан</option>  
  <option value='orange'>Апельсин</option>  
</select>
```

<select>

Атрибуты тега **<select>**:

autocomplete — подсказка для функции автозаполнения формы;

disabled — делает элемент неактивным;

form — связывает список с формой;

multiple — позволяет выбрать несколько опций;

name — задает имя элемента, которое будет отправляться на сервер;

required — делает элемент обязательным для заполнения;

size — задает количество строк в списке.

Атрибуты **value** в тегах **<option>** могут быть использованы для отправки значений на сервер.

<button>

Тег **<button>** создаёт кликабельную кнопку, которая может запускать действия или события на веб-странице. Например, отправку или перезагрузку формы, запуск функции, открытие нового окна или добавление товара в корзину.

<button>Это кнопка</button>

<button onclick='alert("Кнопка нажми!")'>Нажми меня</button>

<button>

Атрибуты тега **<button>**:

name — имя кнопки.

type — тип кнопки, по умолчанию — submit. Атрибут type необходим, если указан атрибут value.

value — значение, которое будет отправлено на сервер при нажатии на кнопку.

disabled — указывает, что кнопка должна быть отключена.

form — одна или несколько форм, к которым принадлежит кнопка.

formaction — URL-адрес файла, который будет обрабатывать ввод при нажатии кнопки.

Напоминаю про ДЗ-2

17/38

не копите плохие долги



у меня ничего не получается

НЕ
напоминаю про ДЗ-1

не сдали 12/38

приходите на
отработку 02.12 в 9:00



- 1.Беляков Кирилл Александрович
- 2.Брекис Богдан Олегович
- 3.Васильев Матвей Николаевич
- 4.Елисеев Станислав Романович
- 5.Козлов Андрей Игоревич
- 6.Кузьмин Сергей Антонович
- 7.Минасян Леон Леонович
- 8.Морозова София Дмитриевна
- 9.Чистяков Георгий Витальевич
- 10.Шайтанов Максим Сергеевич
- 11.Шевцов Андрей Владимирович
- 12.Шорников Михаил Андреевич

02.12.2023

09:00

BCE!