

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

старший преподаватель  
должность, уч. степень, звание

подпись, дата

Н.А. Соловьева  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Применение каскадных таблиц стилей

по курсу: Web-технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4132

подпись, дата

Р.В. Шенин  
инициалы, фамилия

Санкт-Петербург 2024

## СОДЕРЖАНИЕ

1. Задание.....	<b>Error! Bookmark not defined.</b>
2. Используемые средства.....	3
3. Базовое задание.....	3
4. Расширенное задание .....	3
5. Список использованных тегов HTML .....	<b>Error! Bookmark not defined.</b>
6. Скриншоты web-страниц с подписью и номерами рисунков .....	4
7. Тексты документов HTML с названиями файлов .....	6

## 1. Задание

HTML-страницы, разработанные в рамках лабораторной работы № 1, оформить с применением каскадных таблиц стилей. Выполнить задания:

1 ) Использовать три варианта подключения таблиц css:

- связанные таблицы стилей (отдельный внешний файл)
- глобальные таблицы стилей (блок css в файле html (тег style))
- локальные таблицы стилей (локально для одного тега (атрибут style)).

2 ) В таблицах

- оформить границы;
- в одну из ячеек вставить картинку, сохранив при этом выравнивание в таблице.

3 ) Использовать следующие технические средства:

- селекторы: тегов, классов, идентификаторов, составной;
- псевдоклассы (:hover, :visited, :link);
- указание размера: в пикселях, в миллиметрах, через процент;
- указание цвета: слово, шестнадцатеричный формат, десятичный формат.

4 ) Выполнить задание по индивидуальному варианту (Таблица1)

Вариант 21(9)

№	Оформление таблицы	Оформление списка
9	Чередование цветов строк	Сложная нумерация

## 2. Используемые средства

Браузер – Microsoft Edge

Редактор – Visual Studio Code

## 3. Базовое задание

Лабораторная работа состоит из трех HTML-файлов:

index1.html – Основная информация по теме

index2.html – Дополнительная информация по теме

index3.html – источники информации

style.css – внешний файл таблиц стилей

Страница index1 сверстана средствами HTML 4 в ней использованы локальные таблицы стилей, другие - средствами HTML 5. Отдельный внешний файл style.css подключен к странице index2.html

В таблице на странице index2.html оформлены границы и добавлено изображение, также реализовано чередование цветов строк и сложная нумерация. В лабораторной работе использованы все указанные в задании технические средства.


## 4. Расширенное задание

- 1) Выполнено скругление углов прямоугольных элементов шапки и подвала на странице index2.html.
- 2) На странице index3.html сделан фон с градиентом
- 3) Для изображения на странице index1.html использовано свойство filter для увеличения контрастности
- 4) Использован селектор атрибута <a> в файле styles.css
- 5) На первой странице реализовано деление страницы на два столбца

## 5. Скриншоты web-страниц с подписью и номерами рисунков

Параллельные вычисления

Главная страница



- Главная
- Доп. информация
- Вычисления

Введение в параллельные вычисления


Обычно данные анализируются, приходится получать данные из различных источников, а затем отсылать их в компьютер. Часто эти процессы необходимо проводить на больших объемах данных.

Система для распределения задач по нескольким вычислительным узлам и в частности для реализации — параллельные вычисления. С их помощью серверные приложения могут обрабатывать огромные объемы данных и оперативно возвращать результаты запросов.

Общая характеристика параллельных вычислений в целом, а также их реализации. Почему вычисления программистов зачастую в функциях, использующих возможности современных вычислительных систем и кластерных компьютеров для распределения и параллелизации рабочих узлов.

Определение параллельного программирования

Параллельные вычисления имеют в основе большое количество серверных инструментов обработки данных. Эти инструменты используют вычислительную мощность и память. Современная задача представляет из себя набор, что основная задача может быть разбита на подзадачи, которые могут выполняться параллельно на нескольких компьютерах.



Множественные задачи (подзадачи) в исполнении параллельных вычислений

Преимущества параллельного программирования

Параллельные вычисления имеют ряд преимуществ перед традиционными вычислениями. Они позволяют решать задачи за меньшее время. В свою очередь, большие данные могут быть обработаны быстрее, а результаты — точнее. Параллельные вычисления также позволяют использовать ресурсы серверов более эффективно.

Параллельное замедление

Параллельные вычисления имеют ряд преимуществ, но также имеют и недостатки. Распределение задач на кластеры компьютеров сопровождается дополнительными затратами, связанными с тем, как узлы взаимодействуют друг с другом. Поэтому в некоторых случаях распределение задачи может не только не ускорить выполнение, но и замедлить его, снизив производительность параллельных вычислений.

Другим способом решить проблему является использование более мощных серверов, что также не всегда возможно, так как требует дополнительных затрат на оборудование и обслуживание. Поэтому при распределении задач, — это возможность их распределения по кластеру вычислительных узлов.

Параллельные вычисления. Автор: Шенг Ю.К., группа 1122

Напишите название: [Получите следующий файл для более подробной информации: \[index1.html\]\(#\)](#)

Сайт был опубликован в понедельник 20 февраля 2024.

Применение параллельных вычислений

Параллельные вычисления являются одной из ключевых областей в современной информатике, ставшей перед собой задачу увеличения производительности вычислительных систем путем одновременного выполнения нескольких задач. В мире, где объем данных постоянно растет, параллельные вычисления становятся все более важными, обеспечивая высокую эффективность работы и обработку данных. Они используются в различных областях, включая науку, финансы, медицину и другие. Одним из распространенных способов реализации параллельных вычислений является использование многопроцессорных систем, где задачи могут выполняться одновременно на разных процессорах. Также существуют специализированные аппаратные средства, такие как графические процессоры (GPU), которые широко используются для параллельной обработки больших объемов данных. Параллельные вычисления позволяют сократить время выполнения сложных задач, таких как обработка изображений, анализ больших объемов данных, симуляция физических процессов и другие. Они также играют важную роль в разработке высокопроизводительных вычислительных систем и серверов. Параллельные вычисления также представляют собой сложную проблему, требующую правильного управления ресурсами, учета задержек между узлами и предотвращения взаимных блокировок при доступе к общим данным. Таким образом, успешное применение параллельных вычислений требует не только знания технологий, но и опыта в разработке эффективных параллельных программ.

Видео: что такое параллельные вычисления?





Рисунок 1. index1.html(1)

Доп. информация

- Главная
- Доп. информация
- Вычисления

Плюсы и минусы параллельных вычислений

Аспект	Преимущества	Недостатки
Производительность	Увеличение скорости выполнения задач	Возможность замедления из-за накладных расходов на координацию
Масштабируемость	Возможность обработки больших объемов данных	Сложность управления распределенными вычислениями
Надежность	Распределение риска сбоя на нескольких узлах	Возможность сбоя в критических параллельных вычислениях



Параллельные вычисления предоставляют широкий спектр возможностей для увеличения производительности и эффективности, однако требуют внимательного управления и адаптации для достижения желаемых результатов.

Ссылки на ресурсы

I. [Многопоточность](#)

Механизм, позволяющий одновременно выполнять несколько потоков исполнения в пределах одного процесса.

II. [Распределенные вычисления](#)

Метод организации вычислений, при котором задачи разбиваются на подзадачи, которые выполняются на разных вычислительных узлах, часто в сети.

III. [CUDA](#)

Архитектура параллельных вычислений, разработанная компанией NVIDIA, позволяющая использовать графические процессоры для обработки вычислений.

I. [OpenMP](#)

Стандарт для параллельного программирования, позволяющий добавлять параллелизм в программы на C, C++ и Fortran.

II. [MPI](#)

Стандарт для передачи сообщений между процессами в распределенных вычислениях, часто используемый в суперкомпьютерах.

III. [MapReduce](#)

Модель программирования, используемая для обработки больших объемов данных параллельно на распределенном кластере компьютеров.

Параллельное программирование на Python




Рисунок 2. Index2.html(1)

4

Где-то очень глубоко, может даже можно использовать, чтобы распространять вычисления функции по вычислениям кода, но это совсем другая история, которую мы рассмотрим в будущем.

Предположим, что у нас есть датасет `data`, содержащий персональную информацию о студентах, в том числе их возраст и год обучения.

```
import pandas as pd

df = pd.DataFrame({
    'id': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],
    'name': ['John', 'Jane', 'Mike', 'Emily', 'David', 'Sophia', 'Daniel', 'Olivia', 'Liam', 'Ava', 'Noah', 'Mia', 'Ethan', 'Isabella', 'Jacob', 'Charlotte', 'Benjamin', 'Amelia', 'Lucas', 'Harper'],
    'age': [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37],
    'year': [2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039]
})
```

```
print(df)

# Output:
#   id  name  age  year
#  0   1  John  18  2020
#  1   2  Jane  19  2021
#  2   3  Mike  20  2022
#  3   4  Emily  21  2023
#  4   5  David  22  2024
#  5   6  Sophia  23  2025
#  6   7  Daniel  24  2026
#  7   8  Olivia  25  2027
#  8   9  Liam  26  2028
#  9  10  Ava  27  2029
# 10  11  Noah  28  2030
# 11  12  Mia  29  2031
# 12  13  Ethan  30  2032
# 13  14  Isabella  31  2033
# 14  15  Jacob  32  2034
# 15  16  Charlotte  33  2035
# 16  17  Benjamin  34  2036
# 17  18  Amelia  35  2037
# 18  19  Lucas  36  2038
# 19  20  Harper  37  2039
```

Теперь представим, что нам нужно вычислить средний возраст студентов в каждом учебном году. В параллельных вычислениях мы используем библиотеку `numba`, чтобы распространять это вычисление по четырем ядрам процессора.

```
from numba import njit

@njit
def compute_avg_age(groupby_year):
    year, group = groupby_year
    avg_age = group['age'].mean()
    return avg_age

# Apply the function to each year group
df['avg_age'] = df.groupby('year').apply(compute_avg_age)

# Print the result
print(df['avg_age'])

# Output:
#   year  avg_age
#  0  2020    18.0
#  1  2021    19.0
#  2  2022    20.0
#  3  2023    21.0
#  4  2024    22.0
#  5  2025    23.0
#  6  2026    24.0
#  7  2027    25.0
#  8  2028    26.0
#  9  2029    27.0
# 10  2030    28.0
# 11  2031    29.0
# 12  2032    30.0
# 13  2033    31.0
# 14  2034    32.0
# 15  2035    33.0
# 16  2036    34.0
# 17  2037    35.0
# 18  2038    36.0
# 19  2039    37.0
```

Обратите внимание, что в этом примере используется очень маленький набор данных, который при массовых вычислениях может вызвать эффект параллельного замедления, о котором говорилось ранее. Приведенный код подходит только для работы с большими объемами данных.

Для выполнения данных можно применить следующий способ:

Пример вывода кода `df.groupby('year')['age'].mean()`:

```
df.groupby('year')['age'].mean()

# Output:
# year
# 2020    18.0
# 2021    19.0
# 2022    20.0
# 2023    21.0
# 2024    22.0
# 2025    23.0
# 2026    24.0
# 2027    25.0
# 2028    26.0
# 2029    27.0
# 2030    28.0
# 2031    29.0
# 2032    30.0
# 2033    31.0
# 2034    32.0
# 2035    33.0
# 2036    34.0
# 2037    35.0
# 2038    36.0
# 2039    37.0
```

Параллельные вычисления. Автор: Шенни Р.В., группа 4132

Написано [webmaster](#). Посетите следующий сайт для более подробной информации: [wikipedia.org](#)

Сайт был спроектирован и создан 20 февраля 2024.

Рисунок 3. Index2.html(2)

## Использованные источники

Главная  
Доп. информация  
Источники

Параллельные вычисления. Введение

Статья на Хабре

YouTube видео

Параллельные вычисления. Автор: Шенни Р.В., группа 4132

Написано [webmaster](#). Посетите следующий сайт для более подробной информации: [wikipedia.org](#)

Сайт был спроектирован и создан 20 февраля 2024.

## Использованные источники

- Главная
- Доп. информация
- Источники

Параллельные вычисления. Введение

Статья на Хабре

YouTube видео

Параллельные вычисления. Автор: Шенни Р.В., группа 4132

Написано [webmaster](#). Посетите следующий сайт для более подробной информации: [wikipedia.org](#)

Сайт был спроектирован и создан 20 февраля 2024.

Рисунок 4. Index3.html

## 6. Тексты документов HTML с названиями файлов

### Index1.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Параллельные вычисления</title>
</head>
<body>
  <div class="header" style = "margin: 0 auto; text-align: center; width: 80%;">
    <h1 class="title" style="color: #336699;">Параллельные вычисления</h1>
    <h1 class="title" style="color: #336699;">Главная страница</h1>
    <h1 class="logo"></h1>
    <ul class="nav-links" style = "padding: 0; list-style-position: inside;">
      <li style="text-align: center;"><a href="index.html" style="color: #333;">Главная</a></li>
      <li style="text-align: center;"><a href="index2.html" style="color: #333;">Доп.
информация</a></li>
      <li style="text-align: center;"><a href="index3.html" style="color:
#333;">Источники</a></li>
    </ul>
  </div>

  <hr>

  <div class="content" style="display: flex;">
    <div style="width: 50%; padding: 20px;">
      <div><h2 style="color: #336699;">Введение в параллельные вычисления</h2></div>
      <div style="color: #333;">Обычно дата-инженерам приходится получать данные из
нескольких источников, а затем очищать их и агрегировать. Часто эти процессы необходимо
применять на больших объемах данных.</div>
      <div style="color: #333;">Сегодня мы рассмотрим одно из самых фундаментальных
понятий в области вычислительных технологий и в частности дата-инженерии —
параллельные вычисления. С их помощью современные приложения могут обрабатывать
огромные объемы данных за относительно небольшие промежутки времени.</div>
      <div style="color: #333;">Обсудим преимущества параллельных вычислений в целом,
а также их недостатки. Изучим несколько программных пакетов и фреймворков,
использующих возможности современных многоядерных систем и кластеров компьютеров
для распределения и параллелизации рабочих нагрузок.</div>

      <div><h2 style="color: #336699;">Определение параллельного
программирования</h2></div>
      <div>Параллельные вычисления лежат в основе большого количества современных
инструментов обработки данных. Эти фреймворки используют вычислительную мощность и
память. Современные машины предлагают их таким образом, что основная задача может
быть разбита на подзадачи, которые могут выполняться параллельно на нескольких
компьютерах.</div>

    <div><h1></h1>
```

<div style="color: #333;">Множественные задачи (подзадачи) в исполнении параллельного вычисления</div>

<div><h2 style="color: #336699;">Преимущества параллельного программирования</h2></div>

<div style="color: #333;">Первая отличительная черта параллельного программирования связана с применением многоядерных систем, которые позволяют решать задачи за меньшее время. В эпоху больших данных датасеты могут достигать невероятных размеров, и “поместить” их в одну машину — порой невыполнимая задача. Благодаря параллельным вычислениям такие наборы данных могут загружаться распределенным методом с помощью нескольких вычислительных машин.</div>

<div><h2 style="color: #336699;">Параллельное замедление</h2></div>

<div style="color: #333;">Помимо очевидных преимуществ, концепция параллельных вычислений имеет недостатки. Распределение задач по кластеру компьютеров сопровождается некоторыми издержками, связанными с тем, как узлы взаимодействуют друг с другом. Поэтому в некоторых случаях распределение довольно простых задач может не только не ускорить выполнение, но и замедлить его, спровоцировав параллельное замедление. Другими словами, решать небольшие и легкие задачи эффективнее (и, возможно, легче) на одной машине, чем распределять их по кластеру узлов. Еще один фактор, который нужно принимать во внимание при распараллеливании задач, — это возможность их распределения по кластеру вычислительных узлов.</div>

</div>

<div style="width: 50%; padding: 20px;">

<div><h2 style="color: #336699;">Применение параллельных вычислений</h2></div>

<div style="color: #333;">Параллельные вычисления являются одной из ключевых областей в современной информатике, ставящей перед собой задачу увеличения производительности вычислительных систем путем одновременного выполнения нескольких задач. В мире, где объем данных постоянно растет, параллельные вычисления становятся все более важными, обеспечивая возможность эффективного анализа и обработки данных. Они используются в различных областях, включая науку, технику, финансы, медицину и другие.

Одним из распространенных способов реализации параллельных вычислений является использование многоядерных процессоров, где задачи могут выполняться параллельно на разных ядрах. Также существуют специализированные аппаратные ускорители, такие как графические процессоры (GPU), которые широко используются для параллельной обработки больших объемов данных.

Параллельные вычисления позволяют ускорить выполнение сложных задач, таких как обработка изображений, анализ больших наборов генетических данных, симуляции физических процессов и других. Они также играют важную роль в разработке высокопроизводительных вычислительных систем и суперкомпьютеров.

Однако параллельные вычисления также представляют собой сложную проблему, требующую правильного управления ресурсами, учета зависимостей между задачами и предотвращения возможных конфликтов при доступе к общим данным. Таким образом, успешное применение параллельных вычислений требует не только хорошего понимания алгоритмов и структур данных, но и опыта в разработке эффективных параллельных программ.

</div>

```

        <div><h2 style="color: #336699;">Видео: что такое параллельные
вычисления?</h2></div>
        <video width="640" height="360" controls>
            <source src="video.mp4" type="video/mp4">
            Ваш браузер не поддерживает видео элемент.
        </video>
    </div>

</div>

<hr>

<div class="footer">
    <div style="color: #333;">Параллельные вычисления. Автор: Шенин Р.В., группа
4132</div>
    <address>
        Написано <a href="mailto:webmaster@example.com" style="color:
#333;">webmaster</a>.
        Посетите следующий сайт для более подробной информации: <a
href="https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%
D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B
8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F">wikipedia.org</a>.
    </address>
    <div style="color: #333;">Сайт был спроектирован и создан <time datetime="2024-02-
20T08:00" style="color: #333;">20 февраля 2024</time>.</div>
</div>

</body>
</html>

```

## Index2.html

```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Параллельные вычисления</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

    <header class = "myclass1" style = "margin: 0 auto; text-align: center;">
        <h1>Доп. информация</h1>
        <nav>
            <ul>
                <li><a href="index.html">Главная</a></li>
                <li><a href="index2.html">Доп. информация</a></li>
                <li><a href="index3.html">Источники</a></li>
            </ul>
        </nav>
    </header>

```



```

<hr>

<main>
  <section>
    <table border="1" style="margin: 0 auto;">
      <caption><h3>Плюсы и минусы параллельного вычисления</h3></caption>
      <colgroup>
        <col span="1">
        <col span="2">
      </colgroup>
      <thead>
        <tr>
          <th rowspan="2">Аспект</th>
          <th colspan="2">Показатели</th>
        </tr>
        <tr>
          <th>Преимущества</th>
          <th>Недостатки</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Производительность</td>
          <td>Увеличение скорости выполнения задач</td>
          <td>Возможность замедления из-за накладных расходов на координацию</td>
        </tr>
        <tr>
          <td>Масштабируемость</td>
          <td>Возможность обработки больших объемов данных</td>
          <td>Сложность управления распределенными вычислениями</td>
        </tr>
        <tr>
          <td>Надежность</td>
          <td>Распределение риска сбоя на несколько узлов</td>
          <td>Возможность ошибок в алгоритмах параллельных вычислений</td>
        </tr>
      </tbody>
      <tfoot>
        <tr>
          <td></td>
          <td colspan="2">Параллельные вычисления предоставляют широкие возможности для увеличения производительности и эффективности, однако требуют внимательного управления и адаптации для минимизации рисков и недостатков.</td>
        </tr>
      </tfoot>
    </table>
    <hr>
    <h3>Список определений</h3>

    <dl>
      <dt><mark>Многопоточность</mark></dt>

```

<dd>Механизм, позволяющий одновременно выполнять несколько потоков исполнения в пределах одного процесса.</dd>

<dt><mark>Распределенные вычисления</mark></dt>

<dd>Метод организации вычислений, при котором задача разбивается на подзадачи, которые выполняются на разных вычислительных узлах, часто в сети.</dd>

<dt><mark>CUDA</mark></dt>

<dd>Архитектура параллельных вычислений, разработанная компанией NVIDIA, позволяющая использовать графические процессоры для обработки вычислений.</dd>

</dl>

<br>

<dl>

<dt><mark>OpenMP</mark></dt>

<dd>Стандарт для параллельного программирования, позволяющий добавлять параллелизм в программы на C, C++ и Fortran.</dd>

<dt><mark>MPI</mark></dt>

<dd>Стандарт для передачи сообщений между процессами в распределенных вычислениях, часто используемый в суперкомпьютерах.</dd>

<dt><mark>MapReduce</mark></dt>

<dd>Модель программирования, используемая для обработки больших объемов данных параллельно на распределенном кластере компьютеров.</dd>

</dl>

</section>

<hr>

<section>

<h3>Параллельное программирование на Python</h3>

<picture>

<source srcset="phyton.jpg" media="(min-width: 700px)">



</picture>

<p>Пакет multiprocessing входит в стандартную библиотеку Python. Он предлагает интуитивно простой API, позволяющий запускать несколько процессов как для локального, так и для удаленного параллелизма, что является хорошим способом обходить глобальную блокировку интерпретатора (Global Interpreter Lock).</p>

<p>Говоря более конкретно, объект Pool можно использовать, чтобы распараллеливать выполнение функции по нескольким входным значениям таким образом, что входные данные будут распределяться по нескольким процессам, которые могут быть запущены на разных ядрах CPU даже на одной машине.</p>

<p>Предположим, что у нас есть датафрейм Pandas, содержащий персональную информацию о студентах, в том числе их возраст и год обучения:</p>

<code>

<pre>

import pandas as pd

df = pd.DataFrame(

[

(1, 2021, 15),

(2, 2020, 14),

(3, 2021, 17),

(4, 2019, 13),

(5, 2019, 14),

(6, 2020, 15),

(7, 2020, 14),

```

(8, 2021, 14),
(9, 2021, 13),
(10, 2020, 14),
(11, 2019, 12),
(12, 2018, 10),
(13, 2019, 15),
(14, 2019, 16),
],
columns=['student_id', 'class_year', 'age']
)

print(df)
student_id class_year age
0 1 2021 15
1 2 2020 14
2 3 2021 17
3 4 2019 13
4 5 2019 14
5 6 2020 15
6 7 2020 14
7 8 2021 14
8 9 2021 13
9 10 2020 14
10 11 2019 12
11 12 2018 10
12 13 2019 15
13 14 2019 16
</pre>
</code>

```

Теперь представьте, что нам нужно вычислить средний возраст студентов в каждом учебном году. В нижеприведенном примере мы используем библиотеку **multiprocessing**, чтобы распараллелить это вычисление по четырем ядрам локальной машины.

```

<code>
<pre>
from multiprocessing import Pool

def compute_mean_age(groupby_year):
    year, group = groupby_year
    return pd.DataFrame({'age': group['age'].mean()}, index=[year])

with Pool(4) as p:
    mean_age = p.map(compute_mean_age, df.groupby('class_year'))
    df_mean_age_per_year = pd.concat(mean_age)

#И вот результат:

print(df_mean_age_per_year)
age
2018 10.00
2019 14.00

```

2020 14.25

2021 14.75

</pre>

</code>

<p><em>Обратите внимание</em>, что в этом примере используется очень маленький набор данных, который при иных обстоятельствах вызвал бы эффект параллельного замедления, о котором говорилось ранее. Приведенный выше подход пригодится только при работе с большим объемом данных.

</p><p>Для маленьких датасетов можно применять следующий способ:</p>

<p>Пример вывода команды

<samp>df.groupby('class\_year')['age'].mean()</samp>:</p>

<code>

<pre>

class\_year

2018 10.00

2019 14.00

2020 14.25

2021 14.75

Name: age, dtype: float64

</pre>

</code>

</section>

</main>

<hr>

<footer>

<p>Параллельные вычисления. Автор: Шенин Р.В., группа 4132</p>

<address>

Написано <a href="mailto:webmaster@example.com">webmaster</a>.

Посетите следующий сайт для более подробной информации: <a href="https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5\_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F">wikipedia.org</a>.

</address>

<p>Сайт был спроектирован и создан <time datetime="2024-02-20T08:00">20 февраля 2024</time>.</p>

</footer>

</body>

</html>

**Index3.html**

<!DOCTYPE html>

<html lang="ru">

```

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
  <title>Параллельные вычисления</title>
  <style>

    body {
      background-image: radial-gradient(circle, #ffffff, #c462cb);
    }

    /* Стили для заголовка страницы */
    header {
      text-align: center;
      /* background-color: rgb(68, 255, 105); */
    }

    header h1 {
      color: #336699; /* Цвет текста */
    }

    /*Стили для списка ссылок в навигации */
    nav ul li a {
      color: #333;
      font-weight: bold;
    }

    nav ul {
      list-style: none; /* Убираем маркеры списка */
      padding: 0;
    }

    /* Стили для ссылок */
    a:link, a:visited {
      color: blue;
      text-decoration: none;
      transition: color 0.3s;
    }

    a:hover {
      color: red; /* Цвет ссылки в исходном состоянии */
    }

    a[href="https://dzen.ru/a/YtcRTMjIPkkPez4v"] {
      font-size: 2em;
    }

    a[href="https://habr.com/ru/articles/126930/"] {
      font-size: 2em;
    }

    a[href="https://www.youtube.com/watch?v=TdfU7-Cw6sM"] {
      font-size: 2em;
    }
  </style>

```

```

/* Стили для футера */
footer {
    text-align: center; /* Выравнивание текста по центру */
    padding: 20px 0; /* Поля вокруг текста */
    /* background-color: rgb(68, 255, 105); */
    color: #333; /* Цвет текста */
}

/* Стили для адреса в футере */
footer address {
    font-style: normal; /* Отменяем курсив */
    color: #666; /* Цвет текста адреса */
}
</style>
</head>
<body>

<header>
<h1>Использованные источники</h1>
<nav>
<ul>
<li><a href="index.html">Главная</a></li>
<li><a href="index2.html">Доп. информация</a></li>
<li><a href="index3.html">Источники</a></li>
</ul>
</nav>
</header>

<hr>

<main>
<p><a href="https://dzen.ru/a/YtcRTMjlpkPez4v">Параллельные вычисления.
Введение</a></p>
<p><a href="https://habr.com/ru/articles/126930/">Статья на Хабре</a></p>
<p><a href="https://www.youtube.com/watch?v=TdfU7-Cw6sM">YouTube видео</a></p>
</main>

<hr>

<footer>
<p>Параллельные вычисления. Автор: Шенин Р.В., группа 4132</p>
<address>
    Написано <a href="mailto:webmaster@example.com">webmaster</a>.
    Посетите следующий сайт для более подробной информации: <a
href="https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%
D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B
8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F">wikipedia.org</a>.
</address>
<p>Сайт был спроектирован и создан <time datetime="2024-02-20T08:00">20 февраля
2024</time>.</p>
</footer>

```

```
</body>
</html>
```

## styles.css

```
/* Стили для заголовков */
h1, h2, h3 {
    color: #336699; /* Цвет текста */
}

/* Стили для списка ссылок в навигации */
nav ul li a {
    color: #333;
    font-weight: bold;
}

.myclass1 {
    background-color: rgb(254, 115, 249);
    border-radius: 50px;
}

/* Стили для таблиц */
table {
    border-collapse: collapse;
    width: 45%;
    margin: 0 auto;
}

table th, table td {
    border: 10px solid #676767; /* Увеличиваем толщину границ */
    padding: 10px;
}

table tbody tr:nth-child(odd) {
    background-color: #acacac; /* Цвет фона для четных строк */
}

table th {
    background-color: #f0f0f0;
}

table colgroup col:nth-child(odd) {
    background-color: #f0f0f0;
}

#result {
    width: 300px;
}
```

```
/* Стили для изображений */
img {
    max-width: 100%;
    height: auto;
    display: block;
    margin: 0 auto;
}
```

```
/* Стили для текста кода */
code {
    display: block;
    background-color: #f0f0f0;
    padding: 2.64583mm;
    margin: 10px 0;
    overflow-x: auto;
}
```

```
/* Стили для выделенного текста */
mark {
    background-color: #ff0;
    padding: 2px;
}
```

```
/* Стили для ссылок */
a:link, a:visited {
    color: blue;
    text-decoration: none;
    transition: color 0.3s;
}
```

```
a:hover {
    color: #ff0000;
    /* Цвет ссылки при наведении */
}
```

```
dl {
    counter-reset: my-counter; /* Сброс счетчика */
}
```

```
dt {
    margin-top: 40px; /* Отступ сверху для разделов */
}
```

```
dt::before {
    counter-increment: my-counter; /* Увеличение счетчика */
    content: counter(my-counter, upper-roman) ". "; /* Использование римских цифр и точки */
    font-weight: bold; /* Жирный шрифт для маркировки */
}
```



```

dd {
    margin-left: 0; /* Отступ слева для определений */
}

nav ul {
    list-style: none; /* Убираем маркеры списка */
    padding: 0;
}

footer {
    text-align: center; /* Выравнивание текста по центру */
    padding: 20px 0; /* Поля вокруг текста */
    background-color: rgb(254, 115, 249); /* Цвет фона */
    color: #333; /* Цвет текста */
    border-radius: 50px;
}

/* Стили для адреса в футере */
footer address {
    font-style: normal; /* Отменяем курсив */
    color: #666; /* Цвет текста адреса */
}

```