ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Старший преподаватель		Соловьева Н.А.
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

«Применение каскадных таблиц стилей»

по дисциплине: Web-Технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4134к		Столяров Н.С.
		подпись, дата	инициалы, фамилия

Санкт-Петербург
2024

Цель работы: применение каскадных таблиц стилей (css) при оформлении HTML-страниц.

Базовое задание

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

1) Использовать три варианта подключения таблиц css:

- связные 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).

Расширенное задание

1.. Для одного и того же элемента применить правила, расположенные в блоках разного уровня (связанные, глобальные, локальные). Объяснить результат.

2.. использовать символ «+» для объединения селекторов

3.. использовать псевдоэлемент (:first-letter, :first-line и т.д.)

4.. в оформлении применить и показать разницу между margin, border, padding

5.. скруглить углы прямоугольного элемента (свойство border-radius)

6.. сделать фон с градиентом (свойство background-image: xxx-gradient)

7.. использовать свойство text-decoration

8.. применить абсолютное позиционирование

9.. для изображения использовать свойство filter

10.. использовать селектор атрибута

11.. использовать свойство transform

12.. реализовать деление страницы на два столбца

Вариант номер 4

7	Чередование цветов строк	Пункты оформить картинками

Ход выполнения

1) Использовать три варианта подключения таблиц css:

- связные таблицы стилей (отдельный внешний файл)

```
.body {
  display: grid;
  grid-template-columns: 1fr 1fr;
  width: 100%;
  height: 100vh;
  padding-top: 60px;
  box-sizing: border-box;
}

@media screen and (max-width: 1200px) {
  .body {
    grid-template-columns: 1fr;
```

```
 }

 .body > div:last-child {
   height: 100%;
 }

 .body > div:first-child {
   height: 400px;
 }

 div.footer {
   width: 100%;
   left: 0px;
 }
}
```

- Глобальные стили

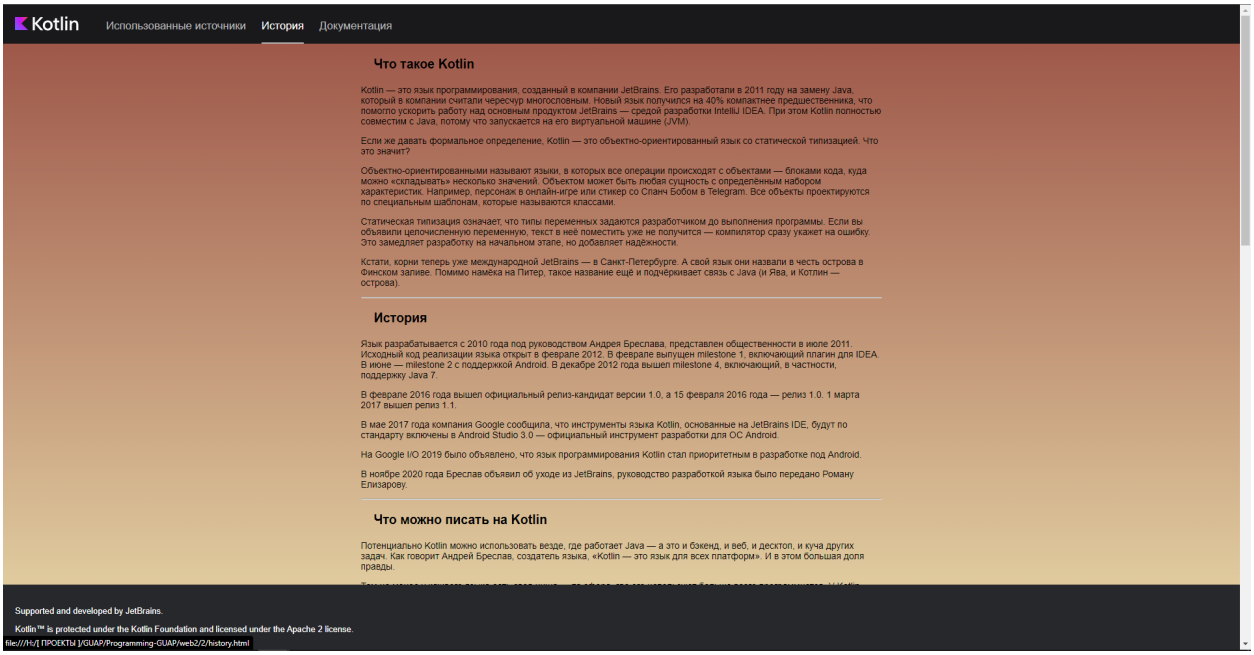
```
<style media="screen">
   .sources {
     padding-top: 60px;
   }

   .sources > ul {
     list-style-type: none;
   }

   .sources > ul > li {
     margin-bottom: 20px;
   }
```

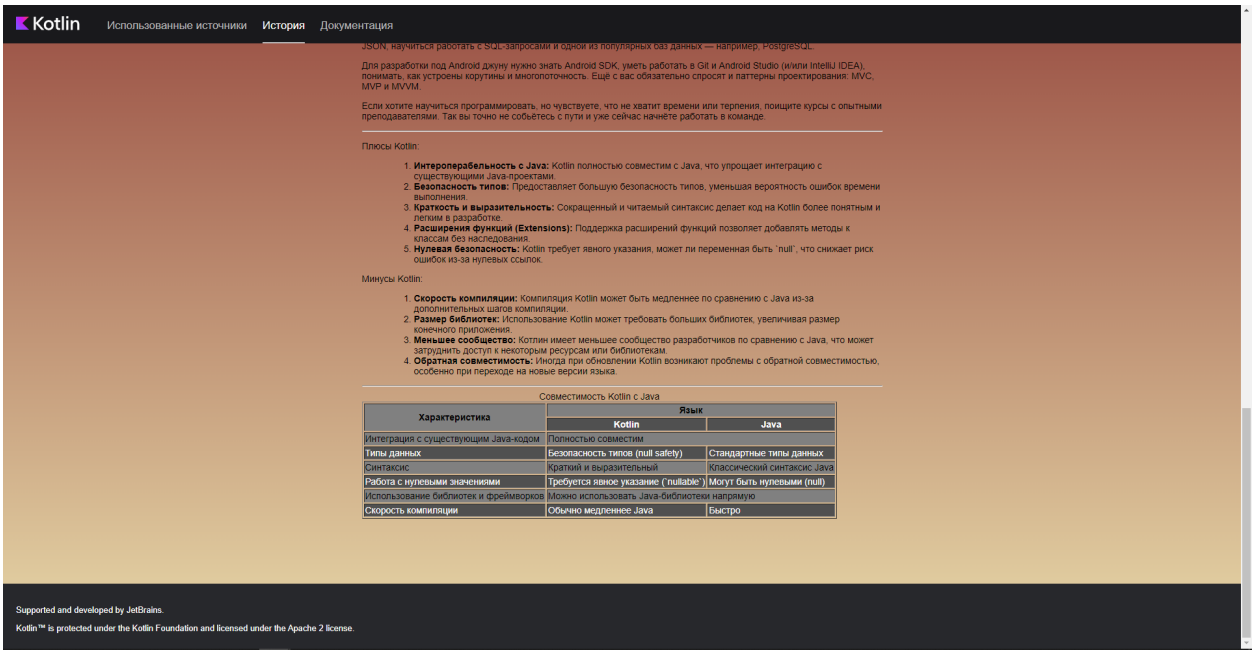
```
  .sources > ul > li > a {
    font-size: 1.2em;
  }

  .sources > ul > li > a:link {
    color: red;
  }

  .sources > ul > li > a:visited {
    color: forestgreen;
    text-decoration-color: hotpink;
  }

  .sources > ul > li > a > img {
    width: 40px;
  }
</style>
```

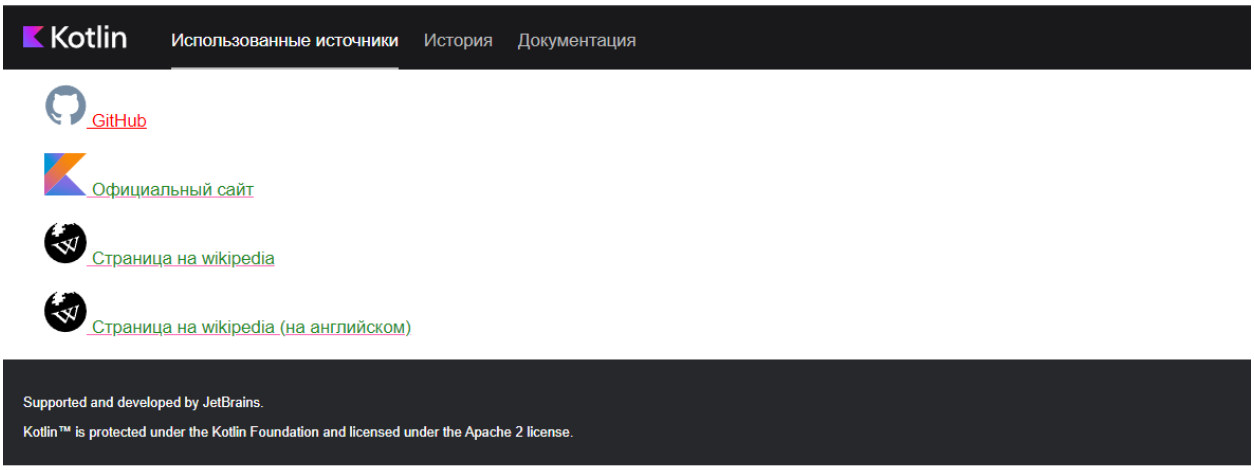
- локальные таблицы стилей (локально для одного тега (атрибут style)).

```
<a href="https://www.jetbrains.com/kotlin-multiplatform/" class="button" style="width: 250px;">Learn about Kotlin Multiplatform</a>
```

Скриншоты


Kotlin
Использованные источники
История
Документация
Что такое Kotlin
Kotlin — это язык программирования, созданный в компании JetBrains. Его разработали в 2011 году на замену Java, который в компании считали чересчур многословным. Новый язык получился на 40% компактнее предшественника, что помогло ускорить работу над основным продуктом JetBrains — средой разработки IntelliJ IDEA. При этом Kotlin полностью совместим с Java, потому что запускается на его виртуальной машине (JVM).
Если же давать формальное определение, Kotlin — это объектно-ориентированный язык со статической типизацией. Что это значит?
Объектно-ориентированными называют языки, в которых все операции происходят с объектами — блоками кода, куда можно «складывать» несколько значений. Объектом может быть любая сущность с определённым набором характеристик. Например, персонаж в онлайн-игре или стикер со Спанч Бобом в Telegram. Все объекты проектируются по специальным шаблонам, которые называются классами.
Статическая типизация означает, что типы переменных задаются разработчиком до выполнения программы. Если вы объявили целочисленную переменную, текст в неё поместить уже не получится — компилятор сразу укажет на ошибку. Это замедляет разработку на начальном этапе, но добавляет надёжности.
Кстати, корни теперь уже международной JetBrains — в Санкт-Петербурге. А свой язык они назвали в честь острова в Финском заливе. Помимо намёка на Питер, такое название ещё и подчеркивает связь с Java (и Ява, и Котлин — острова).
История
Язык разрабатывается с 2010 года под руководством Андрея Бреслава, представлен общественности в июле 2011. Исходный код реализации языка открыт в феврале 2012. В феврале выпущен milestone 1, включающий плагин для IDEA. В июне — milestone 2 с поддержкой Android. В декабре 2012 года вышел milestone 4, включающий, в частности, поддержку Java 7.
В феврале 2016 года вышел официальный релиз-кандидат версии 1.0, а 15 февраля 2016 года — релиз 1.0. 1 марта 2017 вышел релиз 1.1.
В мае 2017 года компания Google сообщила, что инструменты языка Kotlin, основанные на JetBrains IDE, будут по стандарту включены в Android Studio 3.0 — официальный инструмент разработки для ОС Android.
На Google I/O 2019 было объявлено, что язык программирования Kotlin стал приоритетным в разработке под Android.
В ноябре 2020 года Бреслав объявил об уходе из JetBrains, руководство разработкой языка было передано Роману Елизарову.
Что можно писать на Kotlin
Потенциально Kotlin можно использовать везде, где работает Java — а это и бэкенд, и веб, и десктоп, и куча других задач. Как говорит Андрей Бреслав, создатель языка, «Kotlin — это язык для всех платформ». И в этом большая доля правды.
Supported and developed by JetBrains.
Kotlin™ is protected under the Kotlin Foundation and licensed under the Apache 2 license.
file:///H:/[ПРОЕКТЫ]/GUAP/Programming-GUAP/web2/2/history.html


Kotlin
Использованные источники
История
Документация
JSON, научиться работать с SQL-запросами и одной из популярных баз данных — например, PostgreSQL.
Для разработки под Android джуну нужно знать Android SDK, уметь работать в Git и Android Studio (и/или IntelliJ IDEA), понимать, как устроены корутины и многопоточность. Еще с вас обязательно спросят и паттерны проектирования: MVC, MVP и MVVM.
Если хотите научиться программировать, но чувствуете, что не хватит времени или терпения, поищите курсы с опытными преподавателями. Так вы точно не собьётесь с пути и уже сейчас начнёте работать в команде.
Плюсы Kotlin:
1. Интероперабельность с Java: Kotlin полностью совместим с Java, что упрощает интеграцию с существующими Java-проектами.
2. Безопасность типов: Предоставляет большую безопасность типов, уменьшая вероятность ошибок времени выполнения.
3. Краткость и выразительность: Сокращенный и читаемый синтаксис делает код на Kotlin более понятным и легким в разработке.
4. Расширения функций (Extensions): Поддержка расширений функций позволяет добавлять методы к классам без наследования.
5. Нулевая безопасность: Kotlin требует явного указания, может ли переменная быть "null", что снижает риск ошибок из-за нулевых ссылок.
Минусы Kotlin:
1. Скорость компиляции: Компиляция Kotlin может быть медленнее по сравнению с Java из-за дополнительных шагов компиляции.
2. Размер библиотек: Использование Kotlin может требовать больших библиотек, увеличивая размер конечного приложения.
3. Меньшее сообщество: Котлин имеет меньшее сообщество разработчиков по сравнению с Java, что может затруднить доступ к некоторым ресурсам или библиотекам.
4. Обратная совместимость: Иногда при обновлении Kotlin возникают проблемы с обратной совместимостью, особенно при переходе на новые версии языка.
Совместимость Kotlin с Java
Характеристика
Язык
Kotlin
Java
Интеграция с существующим Java-кодом
Полностью совместим
Типы данных
Безопасность типов (null safety)
Стандартные типы данных
Синтаксис
Краткий и выразительный
Классический синтаксис Java
Работа с нулевыми значениями
Требуется явное указание (`nullable`)
Могут быть нулевыми (null)
Использование библиотек и фреймворков
Можно использовать Java-библиотеки напрямую
Скорость компиляции
Обычно медленнее Java
Быстро
Supported and developed by JetBrains.
Kotlin™ is protected under the Kotlin Foundation and licensed under the Apache 2 license.


Kotlin
Использованные источники
История
Документация
GitHub
Официальный сайт
Страница на wikipedia
Страница на wikipedia (на английском)
Supported and developed by JetBrains.
Kotlin™ is protected under the Kotlin Foundation and licensed under the Apache 2 license.

Листинг

sources.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Источники</title>
    <link rel="stylesheet" href="./static/css/main.css">
    <link rel="stylesheet" href="./static/css/history.css">
    <link rel="icon" type="image/x-icon" href="./static/img/favicon.svg">
  </head>
  <style media="screen">
    .sources {
      padding-top: 60px;
    }

    .sources > ul {
      list-style-type: none;
    }

    .sources > ul > li {
      margin-bottom: 20px;
    }

    .sources > ul > li > a {
      font-size: 1.2em;
    }
```

```
  .sources > ul > li > a:link {
    color: red;
  }

  .sources > ul > li > a:visited {
    color: forestgreen;
    text-decoration-color: hotpink;
  }

  .sources > ul > li > a > img {
    width: 40px;
  }
 </style>
 <body>

  <header class="head">
    <img class="logo" src="./static/img/logo.svg" onclick="openLink('./index.html')">
    <div class="menu">
      <a href="./sources.html" class="selected">Использованные источники</a>
      <a href="./history.html">История</a>
      <a href="https://kotlinlang.org/docs/getting-started.html#install-
kotlin">Документация</a>
    </div>
  </header>
  <div class="sources">
    <ul>
      <li>
        <a href="https://github.com/JetBrains/kotlin">
          <img src="./static/img/sources/github.svg">
          GitHub
```

```
        </a>
      </li>
      <li>
        <a href="https://kotlinlang.org/">
          <img src="./static/img/sources/kotlin.svg">
          Официальный сайт
        </a>
      </li>
      <li>
        <a href="https://ru.wikipedia.org/wiki/Kotlin">
          <img src="./static/img/sources/wiki.svg">
          Страница на wikipedia
        </a>
      </li>
      <li>
        <a href="https://en.wikipedia.org/wiki/Kotlin_(programming_language)">
          <img src="./static/img/sources/wiki.svg">
          Страница на wikipedia (на английском)
        </a>
      </li>
    </ul>
  </div>
  <footer class="footer">
    <p class="supported">Supported and developed by JetBrains.</p>
    <p class="license">Kotlin™ is protected under the Kotlin Foundation and licensed under the Apache 2 license.</p>
  </footer>
  <script src="./static/js/main.js" charset="utf-8"></script>
 </body>
</html>
```

hisotry.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
 <head>
  <meta charset="utf-8">
  <title>История</title>
  <link rel="stylesheet" href="./static/css/main.css">
  <link rel="stylesheet" href="./static/css/history.css">
  <link rel="icon" type="image/x-icon" href="./static/img/favicon.svg">
 </head>
 <style media="screen">
  body {
   background: linear-gradient(#9A5044, #E8D9A9) fixed;
  }
 </style>
 <body>
  <div class="history">
   <article class="">
    <p class="name">Что такое Kotlin</p>
    <p>Kotlin — это язык программирования, созданный в компании JetBrains. Его разработали в 2011 году на замену Java, который в компании считали чересчур многословным. Новый язык получился на 40% компактнее предшественника, что помогло ускорить работу над основным продуктом JetBrains — средой разработки IntelliJ IDEA. При этом Kotlin полностью совместим с Java, потому что запускается на его виртуальной машине (JVM).</p>
    <p>Если же давать формальное определение, Kotlin — это объектно-ориентированный язык со статической типизацией. Что это значит?</p>
    <p>Объектно-ориентированными называют языки, в которых все операции происходят с объектами — блоками кода, куда можно «складывать» несколько значений. Объектом может быть любая сущность с определённым набором характеристик. Например, персонаж в онлайн-игре или стикер со Спанч Бобом в Telegram. Все объекты проектируются по специальным шаблонам, которые
```

называются классами.

Статическая типизация означает, что типы переменных задаются разработчиком до выполнения программы. Если вы объявили целочисленную переменную, текст в неё поместить уже не получится — компилятор сразу укажет на ошибку. Это замедляет разработку на начальном этапе, но добавляет надёжности.

Кстати, корни теперь уже международной JetBrains — в Санкт-Петербурге. А свой язык они назвали в честь острова в Финском заливе. Помимо намёка на Питер, такое название ещё и подчёркивает связь с Java (и Ява, и Котлин — острова).

</article>

<hr>

<article class="">

История

Язык разрабатывается с 2010 года под руководством Андрея Бреслава, представлен общественности в июле 2011. Исходный код реализации языка открыт в феврале 2012. В феврале выпущен milestone 1, включающий плагин для IDEA. В июне — milestone 2 с поддержкой Android. В декабре 2012 года вышел milestone 4, включающий, в частности, поддержку Java 7.

В феврале 2016 года вышел официальный релиз-кандидат версии 1.0, а 15 февраля 2016 года — релиз 1.0. 1 марта 2017 вышел релиз 1.1.

В мае 2017 года компания Google сообщила, что инструменты языка Kotlin, основанные на JetBrains IDE, будут по стандарту включены в Android Studio 3.0 — официальный инструмент разработки для ОС Android.

На Google I/O 2019 было объявлено, что язык программирования Kotlin стал приоритетным в разработке под Android.

В ноябре 2020 года Бреслав объявил об уходе из JetBrains, руководство разработкой языка было передано Роману Елизарову.

</article>

<hr>

<article class="">

Что можно писать на Kotlin

Потенциально Kotlin можно использовать везде, где работает Java — а это и бэкенд, и веб, и десктоп, и куча других задач. Как говорит Андрей Бреслав, создатель языка, «Kotlin — это язык для всех платформ». И в этом большая доля правды.

Тем не менее у каждого языка есть своя ниша — та сфера, где его используют больше всего программистов. У Kotlin пока их две — это серверная и мобильная разработка. Хотя его всё чаще можно встретить и в других областях — например, в науке и Data Science.

</article>

<hr>

<article class="">

Безопасность

Безопасность — это то, как язык защищает программиста от его собственных ошибок. В языках со статической типизацией, таких как Java и Kotlin, компилятор следит, чтобы не смешивались несовместимые типы данных — например, строка и число.

В Kotlin вшито несколько функций, которые упрощают работу с типами — например, язык может сам привести переменные к единому типу, если того требует логика кода. Эта функция называется smart cast, или «умное преобразование».

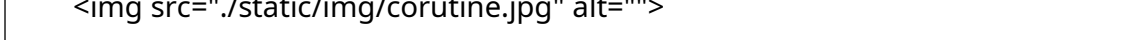
</article>

<hr>

<article class="">

Корутины

Coroutines (корутины) — это средство, которое обеспечивает параллелизм, чтобы программа могла выполнять несколько операций одновременно. Когда возникает необходимость, выполнение одной функции приостанавливается с сохранением данных, и начинает работать другая функция.



Допустим, нам нужно сделать приложение — электронный дневник, чтобы ученик мог зайти в него, узнать свои оценки и посмотреть, что задали.

Если загружать с сервера все данные последовательно, то приложение будет открываться очень долго. Вместо этого ненужные пока данные (например, новости), можно вынести в корутину и поставить их подгрузку на паузу, пока грузится основной интерфейс.

Корутины позволяют расставить приоритеты в работе программы. Этот механизм напоминает многопоточность, но тратит меньше ресурсов процессора — за что их и ценят, например, в мобильной разработке.

</article>

<hr>

```
<article class="">

  <p class="name">Что нужно новичку</p>

  <p>Требования работодателей к новичкам сильно зависят от ниши. Если метите в бэкенд, изучите фреймворки Kotlin для работы с сервером: например, Spring, Ktor и Vert.x. Плюсом к ним надо будет освоить протокол HTTP, форматы XML и JSON, научиться работать с SQL-запросами и одной из популярных баз данных — например, PostgreSQL.</p>

  <p>Для разработки под Android джуну нужно знать Android SDK, уметь работать в Git и Android Studio (и/или IntelliJ IDEA), понимать, как устроены корутины и многопоточность. Ещё с вас обязательно спросят и паттерны проектирования: MVC, MVP и MVVM.</p>

  <p>Если хотите научиться программировать, но чувствуете, что не хватит времени или терпения, поищите курсы с опытными преподавателями. Так вы точно не собьётесь с пути и уже сейчас начнёте работать в команде.</p>

</article>

<hr>

<dl>

  <dt>Плюсы Kotlin:</dt>

  <dd>

    <ol>

      <li><strong>Интероперабельность с Java:</strong> Kotlin полностью совместим с Java, что упрощает интеграцию с существующими Java-проектами.</li>

      <li><strong>Безопасность типов:</strong> Предоставляет большую безопасность типов, уменьшая вероятность ошибок времени выполнения.</li>

      <li><strong>Краткость и выразительность:</strong> Сокращенный и читаемый синтаксис делает код на Kotlin более понятным и легким в разработке.</li>

      <li><strong>Расширения функций (Extensions):</strong> Поддержка расширений функций позволяет добавлять методы к классам без наследования.</li>

      <li><strong>Нулевая безопасность:</strong> Kotlin требует явного указания, может ли переменная быть `null`, что снижает риск ошибок из-за нулевых ссылок.</li>

    </ol>

  </dd>
```

```
      <dt>Минусы Kotlin:</dt>
      <dd>
        <ol>
          <li><strong>Скорость компиляции:</strong> Компиляция Kotlin может быть медленнее по сравнению с Java из-за дополнительных шагов компиляции.</li>
          <li><strong>Размер библиотек:</strong> Использование Kotlin может требовать больших библиотек, увеличивая размер конечного приложения.</li>
          <li><strong>Меньшее сообщество:</strong> Котлин имеет меньшее сообщество разработчиков по сравнению с Java, что может затруднить доступ к некоторым ресурсам или библиотекам.</li>
          <li><strong>Обратная совместимость:</strong> Иногда при обновлении Kotlin возникают проблемы с обратной совместимостью, особенно при переходе на новые версии языка.</li>
        </ol>
      </dd>
    </dl>
    <hr>
    <table border="1">
      <caption>Совместимость Kotlin с Java</caption>
      <tr>
        <th rowspan="2">Характеристика</th>
        <th colspan="2">Язык</th>
      </tr>
      <tr>
        <th>Kotlin</th>
        <th>Java</th>
      </tr>
      <tr>
        <td>Интеграция с существующим Java-кодом</td>
        <td colspan="2">Полностью совместим</td>
      </tr>
```

```
    <tr>
      <td>Типы данных</td>
      <td>Безопасность типов (null safety)</td>
      <td>Стандартные типы данных</td>
    </tr>
    <tr>
      <td>Синтаксис</td>
      <td>Краткий и выразительный</td>
      <td>Классический синтаксис Java</td>
    </tr>
    <tr>
      <td>Работа с нулевыми значениями</td>
      <td>Требуется явное указание (`nullable`)</td>
      <td>Могут быть нулевыми (null)</td>
    </tr>
    <tr>
      <td>Использование библиотек и фреймворков</td>
      <td <td colspan="2">Можно использовать Java-библиотеки напрямую</td>
    </tr>
    <tr>
      <td>Скорость компиляции</td>
      <td>Обычно медленнее Java</td>
      <td>Быстро</td>
    </tr>
  </table>
</div>
<header class="head">
  <img class="logo" src="./static/img/logo.svg" onclick="openLink('./index.html')">
  <div class="menu">
```

```
      <a href="./sources.html">Использованные источники</a>
      <a href="./history.html" class="selected">История</a>
      <a href="https://kotlinlang.org/docs/getting-started.html#install-kotlin">Документация</a>
    </div>
  </header>
  <footer class="footer">
    <p class="supported">Supported and developed by JetBrains.</p>
    <p class="license">Kotlin™ is protected under the Kotlin Foundation and licensed under the Apache 2 license.</p>
  </footer>
  <script src="./static/js/main.js" charset="utf-8"></script>
 </body>
</html>
```

history.css

```
div.history {
  margin-top: 80px;
}

.footer {
  position: sticky;
  width: 100%;
}

div.history {
  box-sizing: border-box;
  padding-top: 0px;
  width: 800px;
  margin-left: 50%;
```

```
  transform: translate(-50%, 0);
  padding-left: 0px;
  padding-right: 0px;
  padding-bottom: 100px;
}

div.history > article > p.name {
  font-size: 1.4em;
  font-weight: bold;
  padding-left: 20px;
}

div.history > article > img {
  max-width: 400px;
}

@media screen and (max-width: 900px) {
  div.history {
    box-sizing: border-box;
    padding-top: 0px;
    width: 100%;
    margin-left: 0;
    transform: none;
    padding-left: 50px;
    padding-right: 50px;
  }
}
```

main.css

```
:root {
  --dark-bg: #19191C;
  --light-bg: #fff;
}

html, body {
  height: 100%;
}

body {
  font-family: -apple-system,BlinkMacSystemFont,Helvetica,Arial,sans-serif,"Apple Color Emoji";
  font-size: 14px;
  line-height: 1.5
  padding: 0px;
  margin: 0px;
}

/* ------------------------------------------------------------------------ */

.body {
  display: grid;
  grid-template-columns: 1fr 1fr;
  width: 100%;
  height: 100vh;
  padding-top: 60px;
  box-sizing: border-box;
}

@media screen and (max-width: 1200px) {
```

```
  .body {
    grid-template-columns: 1fr;
  }

  .body > div:last-child {
    height: 100%;
  }

  .body > div:first-child {
    height: 400px;
  }

  div.footer {
    width: 100%;
    left: 0px;
  }
}

.body > div {
  position: relative;
  width: 100%;
  height: 100%;
  background-color: var(--light-bg);
}

.body > div:first-child {
  background-color: var(--dark-bg);
}
```

```
.body > div:first-child > p.name {
  margin: 0px;
  margin-top: 100px;
  margin-left: 100px;
  font-size: 72px;
  color: #fff;
  font-weight: 500;
}

.body > div:first-child > p.info {
  color: #fff;
  opacity: .5;
  font-size: 35px;
  margin: 0px;
  margin-left: 100px;
}

.body > div:first-child > div.button_start {
  position: absolute;
  margin-top: 40px;
  right: 0px;
  background-color: #7F52FF;
  transition: .1s all;
  border-radius: 20px 0 0 20px;
  cursor: pointer;
  width: 200px;
}

.body > div:first-child > div.button_start > p {
```

```
  padding: 10px 40px;
  margin: 0px;
  color: #fff;
  font-weight: 500;
  font-size: 1.5em;
}

.body > div:first-child > div.button_start:hover {
  background-color: #6B47D2;
}

.body > div:first-child > div.news {
  margin-top: 100px;
}

.body > div:first-child > div.news > ul {
  list-style-type: none;
}

.body > div:first-child > div.news > ul {
  margin-left: 0;
  padding-left: 0;
}

.body > div:first-child > div.news > ul > li {
  display: inline-block;
  background-color: #303033;
  width: 280px;
  height: 380px;
```

```
  border-radius: 8px;
  margin-left: 20px;
  margin-bottom: 20px;
  padding: 10px;
  box-sizing: border-box;
}

.body > div:first-child > div.news > ul > li > img {
  background-color: #303033;
  width: 280px;
  border-radius: 8px 8px 0 0;
  margin: -10px;
}

.body > div:first-child > div.news > ul > li > p {
  color: rgb(255, 255, 255);
}

.body > div:first-child > div.news > ul > li > p.name {
  font-size: 16px;
  font-weight: 530;
}

.body > div:first-child > div.news > ul > li > p.date {
  opacity: .6;
}

.body > div:first-child > div.news > ul > li > p.body {
  opacity: .6;
```

```
  padding: 0px;
}

/* ------------------------------------------------------------------------ */

.body > div:last-child {
  padding-bottom: 50px;
}

.body > div:last-child > p.name {
  color: #19191C;
  font-size: 2em;
  margin-left: 40px;
  font-weight: 600;
}

.body > div:last-child > div.code {
  margin-left: 40px;
  margin-top: 0px;
  width: calc(100% - 80px);
  border-radius: 12px;
  background-color: #0C0C0E;
  margin-bottom: 20px;
}

.body > div:last-child > div.code > video {
  border-radius: 8px;
  margin: 5px;
  width: calc(100% - 10px);
```

```
}

.body > div:last-child > div.info {
  position: relative;
  width: 100%;
  margin: 0px;
  padding: 40px;
  box-sizing: border-box;
}

.body > div:last-child > div.info > p.name {
  font-size: 2em;
  font-weight: bold;
  margin-top: 0px;
  margin-bottom: 0px;
  width: calc(100% - 340px);
}

.body > div:last-child > div.info > p.text {
  font-size: 1.2em;
  padding: 0px;
  width: calc(100% - 340px);
}

.body > div:last-child > div.info > img {
  position: absolute;
  top: 50%;
  right: 35px;
  width: 300px;
```

```
  transform: translate(0, -50%);
}

.body > div:last-child > div.info > a.button {
  margin: 0px;
  padding: 0px;
  color: #fff;
  font-size: 1.2em;
  background-color: #19191C;
  border-radius: 25px;
  padding: 10px 50px;
  white-space: nowrap;
  cursor: pointer;
  text-decoration: none;
}

/* ------------------------------------------------------------------------ */

.head {
  position: fixed;
  top: 0px;
  left: 0px;
  width: 100%;
  height: 60px;
  background-color: var(--dark-bg);
}

.head > img.logo {
  position: absolute;
```

```
  left: 20px;
  top: 18px;
  height: 22px;
  cursor: pointer;
}

.head > div.menu {
  margin-left: 160px;
  margin-top: 23px;
}

.head > div.menu > a {
  color: #BABABB;
  font-size: 1.2em;
  text-decoration: none;
  margin-right: 20px;
  padding-bottom: 16px;
}

.head > div.menu > a:hover {
  color: #FFFFFF;
  border-bottom: 2px solid #BABABB;
}

.head > div.menu > a.selected {
  color: #FFFFFF;
  border-bottom: 2px solid #BABABB;
}
```

```
/* ----------------------------------------------------------------------- */

.footer {
  position: absolute;
  bottom: 0px;
  left: -100%;
  width: calc(100% * 2);
  height: 100px;
  background-color: #27282C;
  box-sizing: border-box;
  overflow: hidden;
}

.footer > p {
  color: #fff;
  font-size: 13px;
}

.footer > p.supported {
  padding: 20px;
  padding-bottom: 0px;
}

.footer > p.license {
  left: 20px;
  bottom: 30px;
  padding-left: 20px;
}
```

```
/* ---------------------------------------------------------------------- */


tr:nth-child(odd) {
  background-color: gray;
}


tr:nth-child(even) {
  background-color: rgb(80, 80, 80);
  color: #ffffff;
}
```