

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень,
звание

подпись, дата

Соловьева Н.А.

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4
«Клиентские языки сценариев. Javascript»
по дисциплине: Web-Технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

4134к

подпись, дата

Столяров Н.С.

инициалы,
фамилия

Санкт-Петербург
2024

Цель работы: получение опыта написания и применения функций на языке javascript.

Базовое задание: Подготовить несколько сценариев по заданиям ниже. Все сценарии встроить в сайт из лабораторной работы № 2. Варианты в таблице 1 определяют способы реализации сценариев (тип события, способ включения сценария, метод выбора элемента на странице, оформление функции). Вид метода (getElementById, getElementsByName, getElementsByTagName), заданный вариантом, ОБЯЗАТЕЛЬНО должен применяться для большей части всех случаев обращения к элементам на странице.

- 1) Вызов всплывающего окна с ФИО и группой автора сайта
- 2) Добавить сценарий по варианту из Таблицы 1 (столбец «сценарий»)
- 3) Добавить пустую строку в таблицу, подготовленную в работе 1
- 4) Изменить цвет элемента
- 5) Изменить размер элемента

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

19	mousemove	файл	getElementById	с именем	9
----	-----------	------	----------------	----------	---

Описание функций

Функция	Обработчик	Описание
addrow	onclick	Добавляет строку в таблицу
change_width	onclick	Изменение ширины блока
change_color	onclick	Изменение цвета текста
movePupil	AddEventListener mousemove	Расчёт движения зрачков глаз

Листинг JS

```
const eye_left = document.getElementById('eye_left');
const pupil_left = document.getElementById('pupil_left');

const eye_right = document.getElementById('eye_right');
const pupil_right = document.getElementById('pupil_right');

function movePupil(eye, pupil, mouseX, mouseY) {
    const eyeRect = eye.getBoundingClientRect();
    // console.log("eyeRect", eyeRect);
    const relativeX = mouseX - (eyeRect.left + eye.offsetWidth / 2);
    const relativeY = mouseY - (eyeRect.top + eye.offsetHeight / 2);
    const distance = Math.sqrt(relativeX ** 2 + relativeY ** 2);
    const maxRadius = Math.min(eye.offsetWidth / 2 - pupil.offsetWidth / 2,
eye.offsetHeight / 2 - pupil.offsetHeight / 2);
    const pupilRadius = Math.min(distance, maxRadius);
    pupil.style.transform = `translate(${(relativeX / distance) * pupilRadius}px, ${(relativeY /
distance) * pupilRadius}px)`;
}
```

```
function movePupils(event) {  
    const mouse = [event.clientX, event.clientY]  
  
    movePupil(eye_left, pupil_left, mouse[0], mouse[1]);  
    movePupil(eye_right, pupil_right, mouse[0], mouse[1]);  
}
```

```
document.addEventListener("mousemove", movePupils);
```

```
function get_month() {  
    const currentMonthNumber = new Date().getMonth();  
    const monthNames = [  
        'январь', 'февраль', 'март', 'апрель', 'май', 'июнь',  
        'июль', 'август', 'сентябрь', 'октябрь', 'ноябрь', 'декабрь'  
    ];  
    return monthNames[currentMonthNumber];  
}
```

```
document.addEventListener("DOMContentLoaded", () =>  
    document.getElementById('date').innerHTML = `Сегодня ${get_month().toUpperCase()}!`);
```

```
function change_color(element) {  
    element.style.color = "#FF0000";  
}
```

```
function change_width(element) {  
    element.style.width = "100px";  
    element.style.height = "100px";  
}
```

```

function addrow(){
    let table = document.getElementById("table")
    var row = table.insertRow(2);
    var cell1 = row.insertCell(0);
    cell1.innerHTML = document.getElementById('table_1').value;
    var cell2 = row.insertCell(1);
    cell2.innerHTML = document.getElementById('table_2').value;
    var cell3 = row.insertCell(2);
    cell3.innerHTML = document.getElementById('table_3').value;
}

```

Листинг 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;
    }

    div.eyes {
        position: fixed;
        right: 0%;
        bottom: 8px;
    }

```

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

```
div.eyes > div.eye {  
  position: relative;  
  display: inline-block;  
  /* width: 150px; */  
  width: 80px;  
  height: 80px;  
  background-color: #fff;  
  border-radius: 50%;  
  margin: 0 30px;  
}
```

```
div.eyes > div.eye > div.pupil {  
  position: absolute;  
  background-color: #000;  
  /* left: 40px;  
  top: 40px; */  
  left: 30px;  
  top: 30px;  
  width: 30px;  
  height: 30px;  
  border-radius: 50%;  
  margin-left: -5px;  
  margin-top: -5px;  
}
```

```
</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.</p>

<p>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.</p>

<p>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).</p>

</article>

<hr>

<article class="">

<p class="name">История</p>

<p>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.</p>

<p>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.</p>

<p>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.</p>

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

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

</article>

<hr>

<article class="">

<p class="name">Что можно писать на Kotlin</p>

<p>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.</p>

<p>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.</p>

</article>

<hr>

<article class="">

<p class="name">Безопасность</p>

<p>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.</p>

<p>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».</p>

</article>

<hr>

<article class="">

<p class="name">Корутины</p>

<p>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.</p>

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

<p>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.</p>

<p>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.</p>

</article>

<hr>

<article class="">

<p class="name">Что нужно новичку</p>

<p>Требования работодателей к новичкам сильно зависят от ниши. Если метите в бэкэнд, изучите фреймворки Kotlin для работы с сервером: например, Spring, Ktor и Vert.x. Плюс 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 хватает 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>

Интероперабельность с Java: Kotlin полностью совместим с Java, что упрощает интеграцию с существующими Java-проектами.

Безопасность типов: Предоставляет большую безопасность типов, уменьшая вероятность ошибок времени выполнения.

Краткость и выразительность: Сокращенный и читаемый

синтаксис делает код на Kotlin более понятным и легким в разработке.

Расширения функций (Extensions): Поддержка расширений функций позволяет добавлять методы к классам без наследования.

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

Минусы Kotlin:

Скорость компиляции: Компиляция Kotlin может быть медленнее по сравнению с Java из-за дополнительных шагов компиляции.

Размер библиотек: Использование Kotlin может требовать больших библиотек, увеличивая размер конечного приложения.

Меньшее сообщество: Kotlin имеет меньшее сообщество разработчиков по сравнению с Java, что может затруднить доступ к некоторым ресурсам или библиотекам.

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
<div>Интеграция с существующим Java-кодом</div> <div>Полностью совместим</div>
<div>Типы данных</div> <div>Безопасность типов (null safety)</div> <div>Стандартные типы данных</div>
<div>Синтаксис</div> <div>Краткий и выразительный</div> <div>Классический синтаксис Java</div>
<div>Работа с нулевыми значениями</div> <div>Требуется явное указание (nullable)</div> <div>Могут быть нулевыми (null)</div>
<div>Использование библиотек и фреймворков</div> <div>Можно использовать Java-библиотеки напрямую</div>
<div>Скорость компиляции</div> <div>Обычно медленнее Java</div> <div>Быстро</div>

```

</table>

<input type="text" id="table_1" placeholder="Характеристика">
<input type="text" id="table_2" placeholder="Kotlin">
<input type="text" id="table_3" placeholder="Java">
<button type="button" onclick="addrow()" name="button">Добавить</button>
</div>

<header class="head">



<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>

<h1 id="date" onclick="show_me()"></h1>

</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>

<div class="eyes">

  <div id="eye_left" class="eye">

    <div id="pupil_left" class="pupil"></div>

    <div class="color"></div>

  </div>

  <div id="eye_right" class="eye">

    <div id="pupil_right" class="pupil"></div>

    <div class="color"></div>

  </div>

</div>

```

```
</footer>

<script src="./static/js/main.js" charset="utf-8"></script>
<script src="./static/js/index.js" charset="utf-8"></script>
<script src="./static/js/date.js" charset="utf-8"></script>
<script src="./static/js/eye.js" charset="utf-8"></script>

</body>
</html>
```

```
<!DOCTYPE html>

<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Kotlin</title>
    <link rel="stylesheet" href="./static/css/main.css">
    <link rel="icon" type="image/x-icon" href="./static/img/favicon.svg">
  </head>
  <body>
    <div class="body">
      <div class="">
        <p class="name">Kotlin</p>
        <p class="info">Concise. Multiplatform. Fun.</p>
        <div class="button_start" onclick="openLink('https://kotlinlang.org/docs/getting-started.html')">
          <p>Get started</p>
        </div>
        <!-- <div class="news">
          <p>Последние новости</p>
          <ul id="news"></ul>
```

</div> -->

</div>

<div class="">

<p class="name">Why Kotlin?</p>

<div class="code">

<video src="/static/video/video.mp4" controls>

<source src="/static/video/video.mp4" type="video/mp4" />

</video>

</div>

<hr>

<div class="info">

<p class="name">Share code on your terms and for different platforms</p>

<p class="text">Simplify the development of cross-platform projects with Kotlin Multiplatform. It reduces time spent writing and maintaining the same code for different platforms while retaining the flexibility and benefits of native programming. Kotlin applications will work on different operating systems, such as iOS, Android, macOS, Windows, Linux, watchOS, and others.</p>

Learn about Kotlin Multiplatform

</div>

<hr>

<div class="info">

<p class="name">Big, friendly and helpful community</p>

<p class="text">Kotlin has great support and many contributors in its fast-growing global community. Enjoy the benefits of a rich ecosystem with a wide range of community libraries. Help is never far away — consult extensive community resources or ask the Kotlin team directly.</p>

Join the community

</div>

<div class="footer">

<p class="supported" onclick="change_color(this)">Supported and developed by JetBrains.</p>

<p class="license">Kotlin™ is protected under the Kotlin Foundation and licensed under the Apache 2 license.</p>

</div>

</div>

</div>

<div class="head">

<div class="menu">

Использованные источники

История

Документация

</div>

<h1 id="date" onclick="show_me()"></h1>

</div>

</body>

<script src="./static/js/main.js" charset="utf-8"></script>

<script src="./static/js/date.js" charset="utf-8"></script>

<script src="./static/js/index.js" charset="utf-8"></script>

</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">

<div class="menu">

Использованные источники

История

Документация

</div>

<h1 id="date" onclick="show_me()"></h1>

</header>

<div class="sources">

GitHub

Официальный сайт

Страница на wikipedia

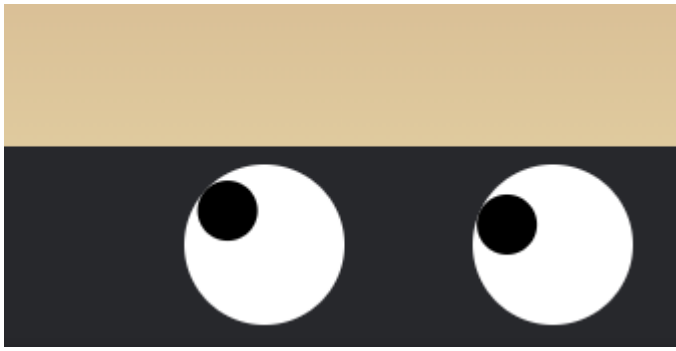
```


</a>
</li>
<li>
  <a href="https://en.wikipedia.org/wiki/Kotlin_(programming_language)">
    
    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>
<script src="./static/js/date.js" charset="utf-8"></script>
</body>
</html>

```

Скриншоты работы программы

Совместимость Kotlin с Java			
Характеристика	Язык		
	Kotlin	Java	
1	3	4	
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	
1	3	4	Добавить



Использованные источникиИсторияДокументация

Сегодня МАЙ!

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 бэкэнд, 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.

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 вшит 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».

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.

Supported and developed by JetBrains.
Kotlin™ is protected under the Kotlin Foundation and licensed under the Apache 2 license.