# Лабораторные работы по дисциплине «Основы Web-технологий»

## Лабораторная работа 1

**Разработка простых HTML-документов**

В рамках работы необходимо создать HTML-документ, используя набор базовых тегов для описания структуры документа и форматирования его составных блоков: изменение начертания текста, выделение цветом, выделение жирным или курсивом, подчеркивание, и так далее. Также необходимо воссоздать текстовую структуру случайной статьи из открытой интернет-энциклопедии Wikipedia, сохранив простое форматирование текста.

**Задание 1.** Записать код на языке HTML, устанавливающий следующие характеристики шрифтов текста:

*Пример 1.*

Установить базовый размер шрифта, равный 2,3,4 и распечатать текст:

базовый размер шрифта равен 2

базовый размер шрифта равен 3

базовый размер шрифта равен 4

Выровнять данный текст по правому краю, по левому краю, по центру.

*Пример 2.*

Установить размер шрифта в абсолютных и относительных единицах и напечатать следующие заголовки:

Пример 6

Пример 5

|  |
| --- |
| Пример 4 |
| Пример 3 |
| Пример 2 |
| Пример 1 |

*Пример 3.*

Установить один из следующих типов шрифта Verdana, Times New Roman, Arial Unicode MS, Calibri и распечатать следующий текст:

Текст, написанный шрифтом Verdana

Текст, написанный шрифтом Times New Roman

Текст, написанный шрифтом Arial Unicode MS

Текст, написанный шрифтом Calibri

*Пример 4.*

Установить цвет текста документа как самим названием, например, red, blue и т.д, так и в системе RGB шестнадцатеричным кодом и распечатать следующий текст:

цвет текста документа красный

цвет текста документа оранжевый

цвет текста документа зеленый

цвет текста документа синий

**Задание 2.** Записать код на языке HTML, выделяющий соответствующий текст, следующим образом:

*Пример 1.*

*Между какими тегами заключен этот текст*

*Пример 2*.

Какими тегами выделен этот текст

*Пример 3*.

~~Текст, выделенный тегами~~

*Пример 4*.

**Какие теги выделяют этот текст**

*Пример 5*.

Проверить, как будет выглядеть текст, выделенный тегами <tt> и </tt>

*Пример 6*.

А где в этом тексте фрагмент выделенный тегами

*Пример 7*.

В этом тексте есть фрагмент текста выделенный тегами

*Пример 8*.

Текст, заключенный между какими тегами, будет увеличен, относительно текущего шрифта

*Пример 9*.

Текст, заключенный между какими тегами будет уменьшен, относительно текущего шрифта

**Задание 3.** Записать код на языке HTML, форматирующий соответствующий текст следующим образом:

*Пример 1*.

Следующий текст записать с нового абзаца. Проверить все атрибуты соответствующего тега.

*Пример 2.*

Этот тег используется для переходов на новую строку без образования абзацев.

*Пример 3*.

Этот тег запрещает перевод строки. Текст, заключенный между тегами данного элемента, отображается в одну строку.

*Пример 4*.

Оформить следующий текст как цитату:

Оформляет находящийся между начальным и конечным тэгами текст как цитату. Используется для длинных цитат. Цитируемый текст отображается отдельным абзацем с увеличенным отступом.

**Задание 4.** Записать код на языке HTML, создающий следующие заголовки

|  |
| --- |
| **Заголовок первого уровня** |
| **Заголовок второго уровня** |
| **Заголовок третьего уровня** |
| **Заголовок четвертого уровня** |
| **Заголовок пятого уровня** |
| **Заголовок шестого уровня** |

## Лабораторная работа 2

**Верстка структурных блоков HTML-документов**

В рамках работы необходимо осуществить верстку ряда структурных блоков HTML-документов, используя соответствующие теги: таблицы, нумерованные и маркированные списки, картинки и гиперссылки. Также необходимо воссоздать контент статьи из открытой интернет-энциклопедии Wikipedia, сохранив форматирование текста, структурные блоки (таблицы, списки, гиперссылки, сноски) и изображения.

Из заданий ниже студенты выполняют задания, соответствующие последней цифре номера их студенческого билета.

Л А Б О Р А Т О Р Н А Я Р А Б О Т А **(1)**

**Задание 1.** Записать код на языке HTML, создающий следующие таблицы:

Пример 1.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 2.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 3.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 4.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 5.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 6.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки 2 ячейка 1 строки 2 ячейка 1 строки 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 7.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки 2 ячейка 1 строки 2 ячейка 1 строки 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 8.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки 2 ячейка 1 строки 2 ячейка 1 строки 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

Пример 9 / 0.

|  |  |
| --- | --- |
| 1 ячейка 1 строки | 2 ячейка 1 строки |
| 1 ячейка 2 строки | 2 ячейка 2 строки |

**Задание 2.** Записать код на языке HTML, объединяющий ячейки таблиц.

Пример 1 / 6.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 и 2 ячейки вместе | | 3 ячейка | 4 ячейка |
| 1 ячейка | 2 ячейка | 3 ячейка | 4 ячейка |
| 1 ячейка | 2 ячейка | 3 ячейка | 4 ячейка |

Пример 2 / 7.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 и 2 ячейки вместе | | 3 ячейка | 4 ячейка |
| 1 ячейка | 2 ячейка3 ячейка | | 4 ячейка |
| 1 ячейка | 2 ячейка3 ячейка4 ячейка | | |

Пример 3 / 8.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1-е ячейки двух строк |  | 2 ячейка | 3 ячейка | 4 ячейка |
|  | 2 ячейка | 3 ячейка | 4 ячейка |
| 1 ячейка |  | 2 ячейка | 3 ячейка | 4 ячейка |

Пример 4 / 9.

|  |  |  |  |
| --- | --- | --- | --- |
| 1-е ячейки 3х строк | 2 ячейка | 3 ячейка | 4-е ячейки 3х строк |
| 2 ячейка | 3 ячейка |
| 2 ячейка | 3 ячейка |

Пример 5 / 0.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 ячейка | 2 ячейка | 3 ячейка | 4 ячейка |
| 1 ячейка | 2 и 3- я ячейка 2х строк | | 4 ячейка |

Л А Б О Р А Т О Р Н А Я Р А Б О Т А **(2)**

**Задание 1.** Записать код на языке HTML, создающий нумерованные списки

Пример 1 / 6.

1. Текст списка 1
2. Текст списка 2
3. Текст списка 3
4. Текст списка 4

Пример 2 / 7.

1. Текст списка 1
2. Текст списка 2
3. Текст списка 3
4. Текст списка 4

Пример 3 / 8.

1. Текст списка 1
2. Текст списка 2
3. Текст списка 3
4. Текст списка 4

Пример 4 / 9.

1. Текст списка 1
2. Текст списка 2
3. Текст списка 3
4. Текст списка 4

Пример 5 / 0.

1. Текст списка 1
2. Текст списка 2
3. Текст списка 3
4. Текст списка 4

**Задание 2.** Записать код на языке HTML, создающий нумерованный список (для всех)

1. Первый пункт

1. Первый пункт

4. Первый пункт

5. Первый пункт

1. Второй пункт
   * второй пункт
     1. Первый пункт
     2. Первый пункт
   * второй пункт
2. Третий пункт

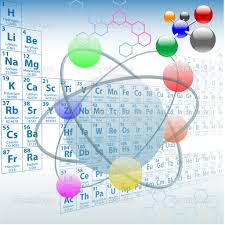
Л А Б О Р А Т О Р Н А Я Р А Б О Т А **(3)**

**Задание 1.** Записать код на языке HTML, располагающий изображение следующим образом относительно текста.

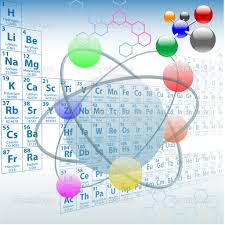
Пример 1 / 6.

Загрузить изображение на страницу, задав значения параметрам его размера и ширину рамки.

Пример 2 / 7.

Выравнивает изображение по левому краю документа. Прилегающий текст обтекает изображение справа. выравнивает изображение по левому краю документа. Выравнивает изображение по левому краю документа. Прилегающий текст обтекает изображение справа. выравнивает изображение по левому краю документа. Выравнивает изображение по левому краю документа. Прилегающий текст обтекает изображение справа. выравнивает изображение по левому краю документа.

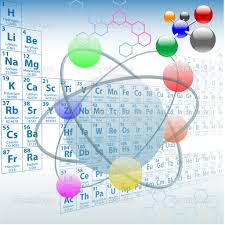
Пример 3 / 8.

Выравнивает изображение по правому краю документа. Текст обтекает изображение слева. Выравнивает изображение по правому краю документа. Текст обтекает изображение слева. Выравнивает изображение по правому краю документа. Текст обтекает изображение слева. Выравнивает изображение по правому краю документа. Текст обтекает изображение слева.

Пример 4 / 9.

Выровнять базовую линию текущей текстовой строки с центром изображения.

Пример 5 / 0.

Выравнивает нижнюю кромку изображения с нижней линией текущей текстовой строки. Выравнивает нижнюю кромку изображения с нижней линией текущей текстовой строки. Выравнивает нижнюю кромку изображения с нижней линией текущей текстовой строки.

Л А Б О Р А Т О Р Н А Я Р А Б О Т А **(4)**

**Задание 1.** Перенести в формат HTML содержимое страницы: <https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B1%D0%B8%D1%80%D1%8C_(%D0%A3%D1%81%D0%BE%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D1%80%D0%B0%D0%B9%D0%BE%D0%BD)>

Название статьи: **Сибирь (Усольский район)**

Сохранить все пункты статьи, заголовки, внешние и внутренние ссылки.

## Лабораторная работа 3

**Создание HTML-форм и стилизация при помощи CSS**

В рамках работы необходимо разработать и настроить HTML-форму, содержащую различные управляющие элементы: поля для ввода текста, группы радио-кнопок, группы чек-боксов, и так далее. Осуществить стилизацию формы при помощи правил CSS: задать шрифты, задать цвет текста и фона, цвет, тип и размер рамок, ширину отступов, и так далее.

Л А Б О Р А Т О Р Н А Я Р А Б О Т А

**Задание 1.** Создать следующую форму в HTML

Начало формы

|  |  |
| --- | --- |
| Поле ввода однострочного текста (text) |  |
| Поле ввода пароля (password) |  |
| Поле ввода многострочного текста (textarea) |  |
| Независимые переключатели (checkbox) | 1  2  3 |
| Зависимые переключатели (radio) | Первый  Второй  Третий |
| Выбор из списка (select) |  |

**Задание 2.** Создать следующие элементы формы в HTML. Выполнить пример, соответствующий последней цифре номера студенческого билета.

*Пример 1 / 6*.



*Пример 2 / 7*.

 Самолет  
 Поезд  
 Автомобиль  
 Автобус  
 

*Пример 3 / 8*.

Пол:  
 М  Ж

*Пример 4 / 9*.

Коротко о себе:

 Молодой человек

 Девушка

 Семейное положение  
 Социальный статус  


*Пример 5 / 0*.

Ваше имя: 

Мужчина 

Женщина 

Детей 

Страна 

* Украина 
* Россия 
* Другая 

Электронный адрес: 

Для подтверждения ввода нажмите на кнопку SUBMIT.

Для сброса введенных данных нажмите на кнопку RESET.



**Задание 3.** Осуществить стилизацию страницы и форм при помощи CSS: изменить шрифты, их размер, цвет, цвет фона, границы, и так далее.

## Лабораторная работа 4

**Основы программирования на Javascript**

В рамках работы необходимо создать HTML-страницу, содержащую ряд скриптов – фукнций, написанных на языке javascript. Скрипты активируются при нажатии на соответствующие кнопки. Каждая функция решает задачи, связанные с обработкой структуры html-документа (изменение, дополнение), демонстрацией модальных окон или реализацией простых алгоритмов обработки данных, например, решение квадратных уравнений или генерация случайных чисел.

***Упражнения 1***

Создать **7 кнопок**, запускающих функции, решающие приведенные ниже 7 заданий. Функции объявлены и реализованы внутри тега <script> в файле html.

**Задание 1.**

Проверить, какое значение возвращает функция window.confirm() при нажатии кнопки «ОК» и какое при нажатии «Отмена».

**Задание 2.**

В поле ввода окна, создаваемого window.prompt (сообщение), записать свою фамилию и вывести приветствие, обращенное к себе, в окне с кнопкой «ОК».

**Задание 3.**

С помощью метода window.prompt(сообщение) узнать имя, возраст и настроение пользователя, а затем с помощью метода document.write() выдать следующий монолог:

Здравствуй, имя

|  |  |
| --- | --- |
| Тебе возраст лет  И у тебя введенное настроение! | |
| Всё верно,не так-ли?! |

**Задание 4.**

Допиcать фрагмент кода для конвертации строкового значения в числовое. Для вывода результата использовать метод [document.write](http://labs.org.ru/javascript-1/#_Javascript_HTML)():

var x = prompt("Введите значение x ", "0");

var y = prompt("Введите значение y ", "0.0");

var integerX = ...

var floatY = ...

var result = integerX + floatY;

...;

**Задание 5.**

Найти ошибки в следующем фрагменте кода:

var firstName = prompt("Введите имя"; "сюда");

var lastName = promt("Введите фамилию");

int age = prompt("Введите возраст");

alert( “В следующем году вам будет”, Age);

**Задание 6.**

Создать несколько абзацев с различными ID, например ID1, ID2, ID3. . . С помощью метода  prompt() ввести ID и, используя innerHTML, методом alert() вывести текст соответствующего абзаца в окне.

**Задание 7.**

Создать несколько абзацев с различными ID, например ID1, ID2, ID3. . . С помощью метода prompt() ввести ID и, используя innerHTML заменить текст соответствующего абзаца на текст, который вводится с помощью метода prompt(). Методом alert() вывести новый текст в окне.

***Упражнения 2***

Выполнить 3 задания, реализовав их во встроенном javascript внутрь html. Каждое задание – **запускается отдельной кнопкой** (всего должно быть **3 кнопки**, с текстом, соответствующим номеру задания)

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.

ПЕРВЫЙ БЛОК ЗАДАНИЙ

**Задание 1.**

В переменной min лежит число от 0 до 59. Определить, в какую четверть часа попадает это число (в первую, вторую, третью или четвертую).

**Задание 2.**

Переменная num может принимать 4 значения: 1, 2, 3 или 4. Если она имеет значение '1', то в переменную res записать 'зима', если имеет значение '2' – 'весна' и т.д. Решить задачу, используя оператор switch.

**Задание 3.**

В переменной day лежит какое-то число из интервала от 1 до 31. Определить, в какую декаду месяца попадает это число (в первую, вторую или третью).

**Задание 4.**

 В переменной month лежит какое-то число из интервала от 1 до 12. Определить, в какую пору года попадает этот месяц (зима, лето, весна, осень).

**Задание 5.**

В переменной year хранится год. Определить, является ли он  високосным (в таком году есть 29 февраля). Год будет високосным в двух случаях: либо он делится на 4, но при этом не делится на 100, либо делится на 400. Так, годы 1700, 1800 и 1900 не являются високосными, так как они делятся на 100 и не делятся на 400. Годы 1600 и 2000 – високосные, так как они делятся на 400.

**Задание 6.**

Создать страницу, которая предлагает пользователю ввести число, а затем выводит кубы всех чисел от 1 до введенного числа при условии, что они не превосходят 1000. Воспользоваться ·

* циклом for;
* циклом while;
* циклом do while.

**Задание 7.**

Создать страницу, которая предлагает пользователю ввести число, а затем выводит информацию о четности числа на страницу документа. Для выяснения четности воспользоваться· оператором

* if … else;
* оператором ? :
* оператором switch.

**Задание 8.**

 Задать некоторое число, например, 12345. Проверить, что первой цифрой в этом числе являются 1, 2 или 3. Если это так – вывести 'да', в противном случае вывести 'нет'.

**Задание 9.**

 Дана строка из 6-ти цифр. Проверьте, что сумма первых трех цифр равняется сумме вторых трех цифр. Если это так – выведите 'да', в противном случае выведите 'нет'.

**Задание 10.**

Дано число n=1000. Делить его на 2 столько раз, пока результат деления не станет меньше 50. Какое число получится? Посчитать количество итераций, необходимых для этого, и записать его в переменную num.

ВТОРОЙ БЛОК ЗАДАНИЙ

**Задание 11.**

 Найти квадратный корень из 379. Результат округлить до целых, до десятых, до сотых.

**Задание 12.**

 Найти квадратный корень из 587. Округлить результат в большую и меньшую стороны, записать результаты округления в объект с ключами 'floor' и 'ceil'.

**Задание 13.**

Написать скрипт, который генерирует случайное число между min и max.

**Задание 14.**

Написать скрипт, который генерирует случайное целое между min и max.

**Задание 15.**

Написать скрипт, который выводит на страницу 10 случайных целых чисел.

**Задание 16.**

 Вывести на экран таблицу умножения (как в школьной тетради).

**Задание 17.**

Даны три числа. Определить, можно ли из отрезков с такими длинами составить треугольник. Определить вид треугольника (прямоугольный, тупоугольный, остроугольный), если он существует.

**Задание 18.**

Даны числа min и max. Найти все треугольники с целочисленными длинами сторон от min до max включительно.

**Задание 19.**

Треугольники, у которых длины сторон и площадь являются натуральными числами, называются треугольниками Герона. Например, площадь треугольника со сторонами 13, 14 и 15 равна 84.

A. Найти n треугольников Герона.

B. Найти n треугольников Герона, длины сторон каждого из которых являются последовательными числами (например, 13, 14, 15).

C. Найти n треугольников Герона, у которых площадь равна периметру.

**Задание 20.**

Написать программу, которая определяет, можно ли из четырех отрезков с данными длинами a, b, c и d составить прямоугольник.

ТРЕТИЙ БЛОК ЗАДАНИЙ

**Задание 21.**

Дан прямоугольник, длины сторон которого выражаются целыми числами. Найти количество квадратов (длины сторон каждого квадрата целые), на которые можно разрезать данный прямоугольник при условии, что при разрезании каждый раз отрезается квадрат наибольшей площади со стороной, общей стороне текущего прямоугольника.

**Задание 22.**

Найти все прямоугольники с целыми длинами сторон данной целочисленной площади. Например, для площади, равной 12, подходят три прямоугольника 1 х 12, 2 х 6, 3 х 4.

**Задание 23.**

Написать программу для нахождения всех различных прямоугольных параллелепипедов, объем которых равен V, а ребра выражены натуральными числами. Параллелепипеды, получающиеся один из другого, если поменять ребра местами, считаются одинаковыми.

**Задание 24.**

Нарисовать пирамиду, как показано на рисунке, у которой должно быть 20 рядов.

x  
xx  
xxx  
xxxx  
`

**Задание 25.**

Нарисовать пирамиду, как показано на рисунке, воспользовавшись циклом for.

xx  
xxxx  
xxxxxx  
xxxxxxxx  
xxxxxxxxxx

**Задание 26.**

С помощью двух вложенных циклов нарисовать следующую пирамидку:

1  
22  
333  
4444  
55555  
666666  
7777777  
88888888  
999999999

**Задание 27 / 29.**

Три натуральных числа a, b и c образуют пифагорову тройку, если . Пифагорова тройка называется основной, если наибольший общий делитель ее чисел равен единице. Например, 3, 4, 5 – основная тройка, 6, 8, 10 – производная тройка. Найти все основные пифагоровы тройки, числа в которых не превышают данное число max.

**Задание 28 / 30.**

Написать программу нахождения всех решений на отрезке [2;100] уравнения n – известное натуральное число.

Придумать решение, которое работает за O(n2), а лучше за O(n) операций.

## Лабораторная работа 5

**Работа с функциями и массивами в javascript**

В рамках работы необходимо реализовать ряд функций javascript, реализующих простые алгоритмы по обработке данных и работе с массивами: заполнение массивов указанными значениями, сортировка массивов, поиск наибольших общих делителей, наименьших общих кратных элементов, и так далее. Каждый алгоритм реализовать в виде отдельной функции, и привязать активацию к кнопке в HTML-документе.

***Упражнения 1***

Выполнить 4 задания, реализовав их во встроенном javascript внутрь html. Каждое задание – **запускается отдельной кнопкой** (всего должно быть **4 кнопки**, с текстом, соответствующим номеру задания). Результат выполнения задания выводится в «консоль» (блок в html документе).

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.

ПЕРВЫЙ БЛОК ЗАДАНИЙ

**Задание 1.**

Написать скрипт, который заполняет массив следующим образом: в первый элемент запишите '1', во второй '22', в третий '333' и так далее.

**Задание** **2.**

Написать скрипт, который заполняет массив следующим образом: в первый элемент записывает 'x', во второй 'xx', в третий 'xxx' и так далее.

**Задание** **3.**

Дан массив с числами. Определить, сколько элементов с начала массива надо сложить, чтобы в сумме получилось больше 10-ти.

**Задание** **4.**

Дан массив с числами. Проверьте, есть ли в нем два одинаковых числа подряд. Если есть - выведите 'да', а если нет – выведите 'нет'.

**Задание** **5.**

Дан массив чисeл arr. Создать массивы, в которых элементы массива arr расположены в обратном порядке: arr1, используя метод reverse, и arr2 без использования метода.

**Задание** **6.**

Переменная lang может принимать 2 значения: 'ru' и 'en'. Если она имеет значение 'ru', то в переменную arr записать массив дней недели на русском языке, а если имеет значение 'en' – то на английском. Решить задачу, используя два оператора if.

**Задание** **7.**

Переменная lang может принимать 2 значения: 'ru' и 'en'. Если она имеет значение 'ru', то в переменную arr записать массив дней недели на русском языке, а если имеет значение 'en' – то на английском. Решить задачу, используя оператор switch-case.

**Задание** **8.**

Дан двумерный массив чисел, например [[1, 2], [3, 4, 5], [6]]. Найти сумму элементов этого массива.

**Задание** **9.**

Дан двумерный массив символов. Выписать предложение, записанное в этом массиве, если каждая его строка это есть отдельное слово предложения.

**Задание** **10.**

Заполнить квадратную матрицу n x n так, чтобы все числа первого столбца и первой строки были равны 1, а каждое из оставшихся чисел  равно сумме верхнего и левого соседей. Вывести на экран заполненную матрицу.

ВТОРОЙ БЛОК ЗАДАНИЙ

**Задание** **11.**

Переменная lang может принимать 2 значения: 'ru' и 'en'. Если она имеет значение 'ru', то в переменную arr записать массив дней недели на русском языке, а если имеет значение 'en' – то на английском. Решить задачу через многомерный массив.

**Задание** **12.**

Создать двухмерный массив. Первые два ключа – это 'ru' и 'en'. Пусть первый ключ содержит элемент, являющийся массивом названий дней недели по-русски, а второй – по-английски. Вывести с помощью этого массива понедельник по-русски и среду по-английски.

**Задание** **13.**

Пусть многомерный массив arr содержит названия цветов по-русски и по-английски. Ввести название цвета по по-русски и получить его английский эквивалент и наоборот.

**Задание 14.**

Дан трехмерный массив с числами, например [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]. Найти сумму элементов этого массива.

**Задание** **15.**

Переменная lang может принимать 2 значения: 'ru' и 'en'. Если она имеет значение 'ru', то в переменную arr записать массив дней недели на русском языке, а если имеет значение 'en' – то на английском. Решить задачу через многомерный массив.

**Задание** **16.**

Создать двухмерный массив. Первые два ключа – это 'ru' и 'en'. Пусть первый ключ содержит элемент, являющийся массивом названий дней недели по-русски, а второй – по-английски. Вывести с помощью этого массива понедельник по-русски и среду по-английски.

**Задание** **17.**

Пусть многомерный массив arr содержит названия цветов по-русски и по-английски. Ввести название цвета по по-русски и получить его английский эквивалент и наоборот.

**Задание 18.**

Дан трехмерный массив с числами, например [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]. Найти сумму элементов этого массива.

**Задание** **19.**

Переменная lang может принимать 2 значения: 'ru' и 'en'. Если она имеет значение 'ru', то в переменную arr записать массив дней недели на русском языке, а если имеет значение 'en' – то на английском. Решить задачу через многомерный массив.

**Задание** **20.**

Создать двухмерный массив. Первые два ключа – это 'ru' и 'en'. Пусть первый ключ содержит элемент, являющийся массивом названий дней недели по-русски, а второй – по-английски. Вывести с помощью этого массива понедельник по-русски и среду по-английски.

ТРЕТИЙ БЛОК ЗАДАНИЙ

**Задания на массивы с использованием пользовательских функций**

**Задание 21.**

Написать скрипт, который сортирует массив чисел по убыванию. Сортировку оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 22.**

Написать скрипт, который сортирует массив чисел следующим образом: сначала располагаются положительные числа, потом отрицательные. Сортировку оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 23.**

Написать скрипт, который сортирует массив чисел следующим образом: сначала располагаются четные числа, потом нечетные. Сортировку оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 24.**

Написать скрипт, который сортирует массив чисел следующим образом: сначала располагаются нечетные числа, потом четные. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 25.**

Написать скрипт, который сортирует массив чисел следующим образом: сначала располагаются простые числа, потом составные. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание** **26.**

Написать скрипт, который сортирует массив чисел следующим образом: сначала располагаются числа, которые делятся на 2,потом на 3 и т.д. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание** 2**7.**

Написать скрипт, который располагает элементы массива в обратном порядке от исходного. Оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание** 2**8.**

Написать скрипт, который сортирует массив строк по возрастанию их длин. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание** 2**9.**

Написать скрипт, который сортирует массив строк по убыванию их длин. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание** **30.**

Написать скрипт, который сортирует массив строк по возрастанию количества гласных в строке. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

ЧЕТВЕРТЫЙ БЛОК ЗАДАНИЙ

**Задание** **31.**

Написать скрипт, который сортирует массив строк по возрастанию без учета регистра. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 32.**

Написать скрипт, который сортирует массив строк по убыванию их длин. Сортировку оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 33.**

Написать скрипт, который удаляет из массива четные числа. Операцию оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и выполнить над ним соответствующие действия, используя функцию этого класса.

**Задание** **34.**

Написать скрипт, который удаляет из массива i чисел, начиная с j-ого. Операцию оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и выполнить над ним соответствующие действия, используя функцию этого класса.

**Задание 35.**

Написать скрипт, который заменяет i-ое число в массиве, последовательностью из j чисел. Операцию оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и выполнить над ним соответствующие действия, используя функцию этого класса.

**Задание** **36.**

Написать скрипт, который вставляет после i-го числа в массиве, последовательность из j чисел. Операцию оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и выполнить над ним соответствующие действия, используя функцию этого класса.

**Задание** **37.**

Дан массив чисел. Найти непрерывный подмассив исходного массива, сумма элементов которого максимальна. Написать функцию getMaxSubSum (массив), которая возвращает эту сумму. Если все элементы отрицательные, то сумма равна нулю. Например, getMaxSubSum([-1, 2, 3, -9, 11]); //11

**Задание** **38.**

Написать скрипт, который сортирует массив строк по возрастанию без учета регистра. Сортировку оформить в виде функции, поместив ее в части head html-файла. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 39.**

Написать скрипт, который сортирует массив строк по убыванию их длин. Сортировку оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и отсортировать его соответствующим образом, используя функцию этого класса.

**Задание 40.**

Написать скрипт, который удаляет из массива четные числа. Операцию оформить в виде функции, поместив ее в js-файл. Cоздать объект типа Array и выполнить над ним соответствующие действия, используя функцию этого класса.

***Упражнения 2***

Выполнить 2 задания, реализовав их во встроенном javascript внутрь html. Каждое задание – **запускается отдельной кнопкой** (всего должно быть **2 кнопки**, с текстом, соответствующим номеру задания). Результат выполнения задания выводится в «консоль» (блок в html документе).

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.

ПЕРВЫЙ БЛОК ЗАДАНИЙ

**Задание 1.**

Дано простое число. Составить функцию, которая будет находить следующее за ним простое число (то есть ближайшее простое число, большее данного, – параметр функции). Например, если дано число 11, то значение функции должно быть 13, если дано 23, то значение 29. Если входное данное не является простым числом, то есть не выполняется условие, то значение функции считать равным нулю.

**Задание 2.**

Составить функцию, значением которой было бы наименьшее общее кратное всех натуральных чисел от 1 до n включительно (n – параметр функции).

**Задание 3.**

Составить функцию, значением которой было бы наименьшее натуральное число, которое можно представить в виде суммы кубов двух натуральных чисел не единственным способом.

**Задание 4.**

Создать функцию, которая подсчитывает среднюю разницу в возрасте между матерями и их детьми. Подумать, как оптимально задать исходные данные.

**Задание 5.**

Целые числа *a1, a2, …, ak*, *k>2* называются взаимно простыми, если наибольший общий делитель этих чисел равен единице.

А. Составить функцию, которая определяет, являются ли данные два натуральных числа взаимно простыми.

В. Составьте функцию, которая проверяла бы, являются ли три числа взаимно простыми.

**Задание 6.**

Составить функцию, значением которой было бы наименьшее общее кратное всех нечетных натуральных чисел от 1 до n включительно (n – параметр функции).

**Задание 7.**

Числа Мерсенна – числа вида *Mn* = *2n - 1* {\displaystyle M\_{n}=2^{n}-1}, где {\displaystyle n}n – [натуральное число](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%82%D1%83%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE). Составить функцию, которая находит первые n чисел Мерсенна.

**Задание 8.**

Запись вида: , где – простые числа, – натуральные числа, *i=1;2;3;...;m* – называется каноническим разложением натурального числа «n».

Составить функцию, которая находит каноническое разложение данного натурального числа.

**Задание 9.**

Составить функцию, которая проверяет, можно ли заданное натуральное число представить в виде произведения двух простых чисел.

**Задание 10.**

Составить функцию, которая проверяет, можно ли заданное натуральное число представить в виде произведения трех простых чисел.

ВТОРОЙ БЛОК ЗАДАНИЙ

**Задание 11.**

Составить функцию, которая проверяет, можно ли заданное натуральное число представить в виде квадрата (куба) какого-либо простого числа.

**Задание 12.**

Создать функцию, которая принимает параметром число от 1 до 7, а возвращает день недели на русском языке.

**Задание 13.**

Дан массив чисел. Создать из него новый массив, где останутся только положительные числа. Для этого создать вспомогательную функцию isPositive(), которая параметром будет принимать число и возвращать true, если число положительное, и false – если отрицательное.

**Задание 14.**

Создать функцию arrayFill(), которая будет заполнять массив заданными значениями. Первым параметром функция принимает значение, которым заполнять массив, а вторым - сколько элементов должно быть в массиве. Например, arrayFill('x', 5) сделает массив ['x', 'x', 'x', 'x', 'x'].

**Задание 15.**

Написать две функции, reverseArray() и reverseArrayInPlace(). Первая получает массив как аргумент и выдаёт новый массив, с обратным порядком элементов. Вторая меняет порядок элементов на обратный в том массиве, который был ей передан в качестве аргумента.

**Задание 16.**

Составить функцию, которая проверяет, можно ли заданное натуральное число представить в виде квадрата (куба) какого-либо простого числа.

**Задание 17.**

Создать функцию, которая принимает параметром число от 1 до 7, а возвращает день недели на русском языке.

**Задание 18.**

Дан массив чисел. Создать из него новый массив, где останутся только положительные числа. Для этого создать вспомогательную функцию isPositive(), которая параметром будет принимать число и возвращать true, если число положительное, и false – если отрицательное.

**Задание 19.**

Создать функцию arrayFill(), которая будет заполнять массив заданными значениями. Первым параметром функция принимает значение, которым заполнять массив, а вторым - сколько элементов должно быть в массиве. Например, arrayFill('x', 5) сделает массив ['x', 'x', 'x', 'x', 'x'].

**Задание 20.**

Написать две функции, reverseArray() и reverseArrayInPlace(). Первая получает массив как аргумент и выдаёт новый массив, с обратным порядком элементов. Вторая меняет порядок элементов на обратный в том массиве, который был ей передан в качестве аргумента.

## Лабораторная работа 6

**Обработка строк в javascript**

В рамках работы необходимо реализовать ряд функций, генерирующих и обрабатывающих строки с использованием стандартных функций javascript. Кроме того, необходимо реализовать процедуры поиска по строковым константам с использованием механизма регулярных выражений. Необходимо разработать ряд шаблонов регулярных выражений для поиска соответствующих подстрок. Алгоритм решения каждой задачи выделяется в отдельную функцию, которая привязывается к соответствующей кнопке в HTML-документе.

***Упражнения 1***

Выполнить 2 задания, реализовав их во встроенном javascript внутрь html. Каждое задание – **запускается отдельной кнопкой** (всего должно быть **2 кнопки**, с текстом, соответствующим номеру задания). Результат выполнения задания выводится в «консоль» (блок в html документе).

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.  
  
Там, где требуется замена частей или преобразование в строках – выводить ***исходную*** ***И*** ***преобразованные*** строки при нажатии на кнопку запуска задания. Например:  
*исходная строка: 'aaa@bbb@ccc'  
преобразованная строка: 'aaa!bbb!ccc'*

ПЕРВЫЙ БЛОК ЗАДАНИЙ

**Задание 1.**

С помощью цикла for сформировать строку '123456789' и записать ее в переменную str.

**Задание 2.**

С помощью цикла for сформировать строку '987654321' и записать ее в переменную str.

**Задание 3.**

С помощью цикла for сформировать строку '-1-2-3-4-5-6-7-8-9-' и записать ее в переменную str.

**Задание 4.**

Дана строка 'aaa@bbb@ccc'. Заменить все @ на '!' с помощью стандартных функций класса String .

**Задание 5.**

Дана строка 'aaa bbb ccc'. Вырезать из нее слово 'bbb' тремя разными способами (использовать [substr](http://theory.phphtml.net/javascript/string/substr.html), [substring](http://theory.phphtml.net/javascript/string/substring.html), [slice](http://theory.phphtml.net/javascript/string/slice.html)).

**Задание 6.**

В переменной date лежит дата в формате '2025-12-31'. Преобразовать эту дату в формат '31/12/2025'.

**Задание 7.**

В переменной date лежит дата в формате '2025-12-31'. Преобразовать эту дату в формат '31.12.2025'.

**Задание 8.**

Дана строка 'я изучаю javascript!'. Используя стандартные функции класса String, решить следующие задачи:

А. Найти количество символов в этой строке.

B. Вырезать из нее слово 'изучаю' всеми возможными способами.

C. Найти позицию подстроки 'изучаю'.

**Задание 9.**

Дана строка 'Я-изучаю-javascript!'. Заменить все дефисы на '!'.

**Задание 10.**

Дана строка 'я изучаю javascript!'. Используя метод split, решить следующие задачи:

A. Записать каждое слово этой строки в отдельный элемент массива.

B. Записать каждый символ этой строки в отдельный элемент массива.

ВТОРОЙ БЛОК ЗАДАНИЙ

**Задание 11.**

Преобразуйте первую букву строки в верхний регистр.

**Задание 12.**

Преобразуйте первую букву каждого слова строки в верхний регистр.

**Задание 13.**

Дана строка, например, '123456'. Сделайте из нее '214365'.

**Задание 14.**

Дана переменная str, в которой хранится какой-либо текст. Если количество символов этого текста больше n, то в переменную res записать первые n символов строки str и в конец добавить троеточие '...'. В противном случае в переменную resзаписать содержимое переменной str.

**Задание 15.**

Ввести строку с помощью метода prompt(). Во введенной строке подсчитать количество гласных букв.

**Задание 16.**

Подсчитать количество слов во введенной строке, найти длину максимального слова, минимального слова и среднюю длину слов. Предусмотреть возможность присутствия в тексте нескольких пробелов и знаков препинания.

**Задание 17.**

Слова строки разбить по двум массивам по принципу четного или нечетного количества букв в слове, используя пробел в качестве разделителя между словами.

**Задание 18.**

Дана строка символов, например, 'abcde'. Проверить, что первым символом этой строки является буква 'a'. Если это так – вывести 'да', в противном случае – 'нет'.

**Задание 19.**

Дана строка. Она может начинаться с http:// или не начинаться. Пример такой строки 'http://phphtml.net'. Проверить то, что строка начинается с http://. Если это так – вывести 'да', в противном случае – 'нет'.

**Задание 20.**

Преобразуйте первую букву каждого слова строки в верхний регистр.

***Упражнения 2***

Выполнить 3 задания, реализовав их во встроенном javascript внутрь html. Каждое задание – **запускается отдельной кнопкой** (всего должно быть **3 кнопки**, с текстом, соответствующим номеру задания). Результат выполнения задания выводится в «консоль» (блок в html документе).

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.  
  
Там, где требуется замена частей или преобразование в строках – выводить ***исходную*** ***И*** ***преобразованные*** строки при нажатии на кнопку запуска задания. Например:  
*исходная строка: 'aaa@bbb@ccc'  
преобразованная строка: 'aaa!bbb!ccc'*

ПЕРВЫЙ БЛОК ЗАДАНИЙ

**Задание 1.**

Дана строка 'ahb acb aeb aeeb adcb axeb'. Написать регулярное выражение, которое найдет строки ahb, acb, aeb по шаблону: буква 'a', любой символ, буква 'b'.

**Задание 2.**

Дана строка 'aba aca aea abba adca abea'. Написать регулярное выражение, которое найдет строки abba adca abea по шаблону: буква 'a', 2 любых символа, буква 'a'.

**Задание 3.**

Дана строка 'aba aca aea abba adca abea'. Написать регулярное выражение, которое найдет строки abba и abea.

**Задание 4.**

Дана строка 'aa aba abba abbba abca abea'. Написать регулярное выражение, которое найдет строки aba, abba, abbba по шаблону: буква 'a', буква 'b' любое количество раз, буква 'a'.

**Задание 5.**

Дана строка 'aa aba abba abbba abca abea'. Написать регулярное выражение, которое найдет строки aa, aba, abba, abbba по шаблону: буква 'a', буква 'b' любое количество раз (в том числе ни одного раза), буква 'a'.

**Задание 6.**

  Дана строка 'aa aba abba abbba abca abea'. Написать регулярное выражение, которое найдет строки aa, aba по шаблону: буква 'a', буква 'b' один раз или ни одного, буква 'a'.

**Задание 7.**

 Дана строка 'aa aba abba abbba abca abea'. Написать регулярное выражение, которое найдет строки aa, aba, abba, abbba.

**Задание 8.**

  Дана строка 'ab abab abab abababab abea'. Написать регулярное выражение, которое найдет строки по шаблону: строка 'ab' повторяется 1 или более раз.

**Задание 9.**

 Дана строка 'a.a aba aea'. Написать регулярное выражение, которое найдет строку a.a, не захватив остальные.

**Задание 10.**

 Дана строка '2+3 223 2223'. Написать регулярное выражение, которое найдет строку 2+3, не захватив остальные.

ВТОРОЙ БЛОК ЗАДАНИЙ

**Задание 11.**

 Дана строка '23 2+3 2++3 2+++3 345 567'. Написать регулярное выражение, которое найдет строки 2+3, 2++3, 2+++3**,** не захватив остальные (+ может быть любое количество).

**Задание 12.**

 Дана строка '23 2+3 2++3 2+++3 445 677'. Написать регулярное выражение, которое найдет строки 23, 2+3, 2++3, 2+++3, не захватив остальные.

**Задание 13.**

 Дана строка '\*+ \*q+ \*qq+ \*qqq+ \*qqq qqq+'. Написать регулярное выражение, которое найдет строки \*q+, \*qq+, \*qqq+, не захватив остальные.

**Задание 14.**

Дана строка 'aa aba abba abbba abbbba abbbbba'. Написать регулярное выражение, которое найдет строки abba, abbba, abbbba и только их.

**Задание 15.**

Дана строка 'aa aba abba abbba abbbba abbbbba'. Написать регулярное выражение, которое найдет строки вида aba, в которых 'b' встречается не более 3-х раз.

**Задание 16.**

Дана строка 'aa aba abba abbba abbbba abbbbba'. Написать регулярное выражение, которое найдет строки вида aba, в которых 'b' встречается не менее 4-х раз.

**Задание 17.**

 Дана строка 'aba aea aca aza axa'. Написать регулярное выражение, которое найдет строки aba, aea, axa, не затронув остальных.

**Задание 18.**

Дана строка 'aba aea aca aza axa a.a a+a a\*a'. Написать регулярное выражение, которое найдет строки aba, a.a, a+a, a\*a, не затронув остальных.

**Задание 19.**

Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – цифра от 3-х до 7-ми.

**Задание 20.**

 Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними любые- цифра от 3-х до 7-ми любых цифр.

ТРЕТИЙ БЛОК ЗАДАНИЙ

**Задание 21.**

 Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – буква от a до g.

**Задание 22.**

 Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – буква от a до f и от j до z.

**Задание 23.**

 Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – буква от a до f и от A до Z.

**Задание 24.**

Дана строка 'aba aea aca aza axa a-a a#a'. Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – не 'e' и не 'x'.

**Задание 25.**

 Дана строка 'wйw wяw wёw wqw'. Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'w', а между ними – буква кириллицы.

**Задание 26.**

 Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – буква от a до g.

**Задание 27.**

 Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – буква от a до f и от j до z.

**Задание 28.**

 Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – буква от a до f и от A до Z.

**Задание 29.**

Дана строка 'aba aea aca aza axa a-a a#a'. Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'a', а между ними – не 'e' и не 'x'.

**Задание 30.**

 Дана строка 'wйw wяw wёw wqw'. Написать регулярное выражение, которое найдет строки следующего вида: по краям стоят буквы 'w', а между ними – буква кириллицы.

## Лабораторная работа 7

**Работа с объектами и обработка событий в javascript**

В рамках работы решаются задачи, связанные с созданием и манипулированием составными объектами, а также с обработкой событий, происходящих на странице. Необходимо реализовать ряд функций, осуществляющих работу с составными объектами (объектами самой html-страницы), которые активируются при загрузке страницы, нажатии на кнопки мыши и клавиатуры, изменении фокуса, и так далее. Функции редактируют CSS-свойства html-элементов, позволяя динамически изменять интерфейс в зависимости от пользовательских действий.

***Упражнения 1***

Выполнить 2 задания, реализовав их во встроенном javascript внутрь html. Каждое задание – **запускается отдельной кнопкой** (всего должно быть **2 кнопки**, с текстом, соответствующим номеру задания). Результат выполнения задания выводится в «консоль» (блок в html документе).

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.

ПЕРВЫЙ БЛОК ЗАДАНИЙ

**Задание 1.**

Создать объект с ключами 'Минск', 'Москва', 'Киев' и т.д. с элементами 'Беларусь', 'Россия', 'Украина' и т.д. Вывести на экран сначала все ключи объекта, затем все значения, и наконец, с помощью цикла for-in вывести строки такого формата: 'Минск – это Беларусь'.

**Задание 2.**

Создать объект с ключами 'Минск', 'Москва', 'Киев' и т.д. с элементами 'Беларусь', 'Россия', 'Украина' и т.д. Вывести на экран с помощью цикла for-in строки такого формата: 'Беларусь : столица Минск '.

**Задание 3.**

Создать объект футбольная команда с ключами Николай, Андрей, Петр и т.д. с элементами 'вратарь', 'нападающий', 'правый защитник' и т.д. С помощью цикла for-in вывестие на экран строки такого формата: 'Николай - вратарь'. и т.д.

**Задание 4.**

Создать объект футбольная команда с ключами Николай, Андрей, Петр и т.д. с элементами 'вратарь', 'нападающий', 'правый защитник' и т.д. С помощью цикла for-in вывестие на экран строки такого формата: 'Николай – вратарь'. и т.д. Предусмотреть случай с одинаковыми именами.

**Задание 5.**

Создать объект финал чемпионата мира по забегу на 100 метров, ключами которого являются фамилии спринтеров Усэйн Болт, Джастин Гэтлин, Кристиан Коулман и т.д., а элементами – результаты забега в секундах 9.97, 9.92, 9.94 и т.д. На страницу документа распечатать список спортсменов в соответствии с показанными результатами.

**Задание 6.**

 Создать объект цвета, ключами у которого являются названия цветов на английском языке, например, red, green и т.д. а элементами их русские эквиваленты. С помощью цикла for-in вывестие на экран все названия цветов и через дефис их перевод.

**Задание 7.**

 Создать объект неделя, ключами у которого являются названия дней недели, а элементами признак того, является ли день будним или выходным. С помощью цикла for-in вывести все дни недели, а выходные дни вывести жирным шрифтом.

**Задание 8.**

 Составьте массив дней недели. С помощью цикла for выведите все дни недели, а текущий день выведите курсивом. Текущий день должен храниться в переменной day.

**Задание 9.**

Создать объект city1 (var city1 = {}). Добавить к нему свойства name (название города) и population(население). Вывести свойства на экран, обратившись к ним напрямую.

**Задание 10.**

Создать объект city2, используя упрощенный метод, со свойствами название города, его население  и выведите их.

ВТОРОЙ БЛОК ЗАДАНИЙ

**Задание 11.**

Создать ассоциативный массив (объект) заработных плат obj, используя упрощенный синтаксис. Вывести на экран зарплату Петрова и Иванова, если таковые имеются.

**Задание 12.**

 Создать объект obj. Вывести на экран элемент с ключом 'c' двумя способами: через квадратные скобки и как свойство объекта:

var obj = {a: 1, b: 2, c: 3};

**Задание 13.**

Создать объект с днями недели. Ключами в нем должны служить номера дней от начала недели (понедельник – первый и т.д.). Вывести на экран текущий день недели.

**Задание 14.**

Создать объект с ключами 'Минск', 'Москва', 'Киев' и т.д. с элементами 'Беларусь', 'Россия', 'Украина' и т.д. Вывести на экран сначала все ключи объекта, затем все значения, и наконец, с помощью цикла for-in вывести строки такого формата: 'Минск – это Беларусь'.

**Задание 15.**

Создать объект с ключами 'Минск', 'Москва', 'Киев' и т.д. с элементами 'Беларусь', 'Россия', 'Украина' и т.д. Вывести на экран с помощью цикла for-in строки такого формата: 'Беларусь : столица Минск '.

**Задание 16.**

Создать объект футбольная команда с ключами Николай, Андрей, Петр и т.д. с элементами 'вратарь', 'нападающий', 'правый защитник' и т.д. С помощью цикла for-in вывестие на экран строки такого формата: 'Николай - вратарь'. и т.д.

**Задание 17.**

Создать объект футбольная команда с ключами Николай, Андрей, Петр и т.д. с элементами 'вратарь', 'нападающий', 'правый защитник' и т.д. С помощью цикла for-in вывестие на экран строки такого формата: 'Николай – вратарь'. и т.д. Предусмотреть случай с одинаковыми именами.

**Задание 18.**

Создать объект финал чемпионата мира по забегу на 100 метров, ключами которого являются фамилии спринтеров Усэйн Болт, Джастин Гэтлин, Кристиан Коулман и т.д., а элементами – результаты забега в секундах 9.97, 9.92, 9.94 и т.д. На страницу документа распечатать список спортсменов в соответствии с показанными результатами.

**Задание 19.**

 Создать объект цвета, ключами у которого являются названия цветов на английском языке, например, red, green и т.д. а элементами их русские эквиваленты. С помощью цикла for-in вывестие на экран все названия цветов и через дефис их перевод.

**Задание 20.**

 Создать объект неделя, ключами у которого являются названия дней недели, а элементами признак того, является ли день будним или выходным. С помощью цикла for-in вывести все дни недели, а выходные дни вывести жирным шрифтом.

***Упражнения 2***

Выполнить 2 задания, реализовав их во встроенном javascript внутрь html. Каждое задание – **выделено областью с border**, с текстом, соответствующим номеру задания. Всего должно быть 2 области. В зависимости от задания результат выполнения выводится либо в «консоль» (блок в html документе), либо в другое поле (например, текстовой поле), либо появляется в модальном окне при вызове функции alert.

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.

ПЕРВЫЙ БЛОК ЗАДАНИЙ

**Задание 1.**

 Создать кнопку с надписью «press» . По щелчку на кнопке вывести диалоговое окно с сообщением «Ура!».

**Задание 2.**

Создать кнопку с надписью «yes». При щелчке на кнопке надпись изменить на «no».

**Задание 3.**

Создать кнопку с надписью «press». По щелчку на кнопке вызывается пользовательская функция, которая спрашивает, как зовут пользователя, а затем приветствует его по имени.

**Задание 4.**

Создать кнопку с надписью «press». По щелчку на кнопке меняется надпись на ней и цвет надписи.

**Задание 5.**

Создать две кнопки с соответствующими надписями «план» и «отчет». Вызов пользовательской функции plan() или report() осуществляется в зависимости от выбранной кнопки. При этом с помощью метода alert() на экран выводится сообщение о том, какая кнопка активизирована.

**Задание 6.**

Создать кнопку с надписью «yes» . При наведении мыши на кнопку меняется запись и ее цвет, при выведении все возвращается в начальное положение

**Задание 7.**

Создать выпадающий список «Любимое время года» с именем season. Добавить соответствующие элементы в этот список. Написать функцию JavaScript с именем selectchange (), которая вызывается каждый раз при выборе нового элемента списка. Функция в диалоговом окне печатает выбранное время года. Для обращения к значениям тегов <option> использовать массив свойств options и свойство selectedIndex, которое возвращает выбранный в настоящий момент элемент списка.

**Задание 8.**

Создать набор радиокнопок «Любимый язык программирования» с именем language. Добавить соответствующие элементы в этот список. Написать функцию JavaScript с именем selectchange (), которая вызывается каждый раз при выборе нового элемента. Функция на странице документа печатает фразу: «Вы выбрали и название выбранного языка» в диалоговом окне .

**Задание 9.**

Создать два текстовых поля, озаглавив их соответственно “Поле1” и “Поле2”. Поле1 содержит текст: “input something… “, а Поле2 – “and here”. Когда Поле2 получает фокус, оно очищается.

**Задание 10.**

Создать два текстовых поля, озаглавив их соответственно “отсюда” и “сюда”. Получить данные введенные в первом текстовом поле и вставить их во второе.

ВТОРОЙ БЛОК ЗАДАНИЙ

**Задание 11.**

Создать кнопку с надписью «я – кнопка». При каждом щелчке на кнопке меняется цвет ее фона. Цвета задаются массивом.

**Задание 12.**

Запрограммировать возможность добавления слов из текстового поля в поле для ввода многострочного текста. Добавить слово можно с помощью кнопки «Добавить» или с помощью клавиши <Enter>. При переносе слова из текстового поля оно очищается.

**Задание 13.**

Создать поле для многострочного ввода текста textarea**.** Ввести туда некоторый текст, методом alert вывести его и количество символов в нем.

Введите текст и кликните мышкой вне поля...</textarea>

**Задание 14.**

Создать текстовое поле, в которое можно записывать числа, и кнопку с надписью «Играть». При щелчке на кнопку вызывается функция, которая выводит на страницу сообщение "Меньше ста" или "Большое ста" в зависимости от числа, введенного в текстовое поле.

**Задание 15.**

Изменить цвет фона документа, кликнув на кнопку.

**Задание 16.**

Используя функцию, изменить текст элемента <p> при нажатии на кнопку.

**Задание 17.**

Используя функцию, изменить стиль элемента < body> при нажатии на кнопку.

**Задание 18.**

Изменить стиль элемента <p> при нажатии на кнопку.

**Задание 19.**

Создать кнопку с надписью «я – кнопка». При каждом щелчке на кнопке меняется цвет ее фона. Цвета задаются массивом.

**Задание 20.**

Запрограммировать возможность добавления слов из текстового поля в поле для ввода многострочного текста. Добавить слово можно с помощью кнопки «Добавить» или с помощью клавиши <Enter>. При переносе слова из текстового поля оно очищается.

## Лабораторная работа 8

**Разработка комплексного интерактивного HTML-документа**

В рамках работы необходимо создать HTML-документ с использованием технологий CSS и Javascript, решающий комплексную задачу в интерактивном режиме взаимодействия с пользователем. Примерами задач выступают: разработка простого калькулятора, разработка финансового калькулятора, разработка виджета для построения графиков заданных функций, разработка виджета для демонстрации формул и графиков распределений теории вероятностей, и так далее. При разработке заданий может потребоваться использование сторонних библиотек CSS или Javascript.

**Лабораторная работа**

Выполнить 1 задание, реализовав его в виде отдельной HTML-страницы с использованием технологий CSS и Javascript. Таблицы стилей CSS и коды Javascript реализовать через внутри HTML-документа в соответствующих тегах (не в виде отдельных файлов).

При разработке HTML-страницы обязательно использовать следующие элементы:

* Теги разметки структуры страницы – div, p, span, a, button, img, и так далее.
* Стили шрифтов (через CSS)
* Стили фона и рамок страницы, блоков, кнопок (через CSS)
* Обработку событий мыши и/или клавиатуры

Остальные элементы (формы, регулярные выражения, массивы) – используются в зависимости от особенностей решаемой задачи.

Задания выбираются в зависимости от ***последней цифры номера студенческого билета студента***.

**Задание 1**

Разработать финансовый калькулятор, вычисляющий сумму дохода клиента банка при внесении денег на вклад (сложный процент).

Входные данные:

* Начальная сумма
* Процентная ставка, в год
* Срок вклада – в годах или месяцах

Выходные данные:

* Сумма на вкладе в конце срока вклада
* Доход клиента банка за указанный период

Построить график роста суммы вклада в зависимости от времени с использованием библиотеки Chart.js (<https://www.chartjs.org>) – по оси X идет срок вклада, по оси Y – сумма на вкладе (при использовании сложного процента график должен иметь экспоненциальную форму).

**Задание 2**

Разработать виджет для построения графиков базовых полиномиальных функций – степени от 1 до 4. Выбор степени полинома реализовать в виде кнопок, вкладок или радиокнопок. Пользователей имеет возможность изменять коэффициенты полиномиальной функции.

Входные данные:

* Степень полинома, от 1 до 4
* Коэффициенты полинома (количество коэффициентов = степень полинома + 1). Например, y = ax + b (степень 1); y = ax2 + bx + c (степень 2), и так далее.

Выходные данные:

* При нажатии кнопки – построить график указанной функции с использованием библиотеки Chart.js (<https://www.chartjs.org>).

**Задание 3**

Разработать страницу, демонстрирующую подробное пошаговое решение линейных и квадратных уравнений, с отображением математических формул. Коэффициенты уравнений задаются пользователем через текстовые поля.

Входные данные:

* Вид уравнения – линейной или квадратное (кнопка, вкладки, радиокнопка, и т.д.)
* Коэффициенты уравнения – через текстовые поля (заглавные буквы):
  + Ax + B = C (для линейного уравнения)
  + Ax2 + Bx + C = D (для квадратного)

Выходные данные:

* Пошаговое решение задачи (по действиям), вплоть до ответа (x = … - для линейного уравнения, x1 = … x2 = … - для квадратного)

Математические формулы выводить на экран с использованием библиотеки KaTeX (<https://katex.org/>).

**Задание 4**

Разработать простой калькулятор, поддерживающий основные математические операции – сложение, вычитание, умножение, деление. Набор цифр и выбор операций реализовать при помощи кнопок, без использования текстовых полей.

Входные данные:

* Пользователь набирает на «экранной клавиатуре» число, затем выбирает операцию, набирает другое число, и жмет на «равно».

Выходные данные:

* В «окне калькулятора» отображается результат. Результат становится текущим (левым) числом, и можно продолжать вычисления, нажав на кнопку операции и введя очередное число.

**Задание 5**

Разработать страницу, демонстрирующую понятие куайна в программировании (программа, которая выводит на экран свой собственный исходный код), а также примеры исходного кода куайнов на нескольких языках программирования, с подсветкой синтаксиса для каждого языка.

Входные данные:

* Коды куайнов отображаются при нажатии пользователем на соответствующие кнопки. Количество кнопок – минимум 3: Python, Java, C++.

Выходные данные:

* Краткое описание понятия куайн – по аналогии с соответствующими страницами Википедии (<https://en.wikipedia.org/wiki/Quine_(computing)>)
* Исходные коды куайнов на нескольких языках программирования, отображающиеся при нажатии на соответствующие кнопки. В один момент отображается только один код на одном языке программирования. Используется подсветка синтаксиса.

Для подсветки синтаксиса программного кода использовать библиотеку Prism (<https://prismjs.com/>).

**Задание 6**

Разработать аналог страницы «Российский университет дружбы народов» из открытой интернет-энциклопедии Wikipedia. Воспроизвести первые параграфы, а также раздел «История», сохранив все форматирование и ссылки, кроме ссылок на примечания.

Реализовать всплывающие окна при наведении мыши на ссылки первого параграфа. Контент всплывающих окон формируется по аналогии с тем, как это реализовано в Wikipedia. Реализовать всплывающие окна для следующих терминов:

* Университет (текст)
* Москва (текст, картинка)
* Россия (текст, картинка)
* Совет министров СССР (текст, картинка)
* Патрис Лумумба (текст, картинка)
* Ректор (текст)

Для реализации всплывающих окон при наведении мыши использовать библиотеку Tippy.js (<https://atomiks.github.io/tippyjs/>).

**Задание 7**

Разработать страницу, представляющую собой форму для внесения персональных данных о человеке. На странице содержатся следующие обязательные поля (другие атрибуты могут быть добавлены студентом по желанию). Реализовать автоматическое форматирование ввода пользователя:

* Фамилия, имя, отчество – ограничение на количество символов, первый символ прописной, остальные строчные
* Дата рождения – дата в формате дд.мм.гггг.
* Номер мобильного телефона (в России) – с префиксом +7, выделением кода в скобки, разграничением разрядов символом «-». Например, пользователь вводит в поле «9161234567», а в текстовом поле автоматически отображается «+7 (916) 123-45-67».
* Заработная плата (в рублях в месяц) – с префиксом в виде символа рубля
* Данные о кредитной карте: номер, срок действия, CVC код – в соответствии с международными форматами

При нажатии на кнопку отправки формы – вся информация дублируется в «консоль» в HTML-документе (печатается).

Для реализации автоматического форматирования ввода использовать библиотеку Cleave.js (<https://nosir.github.io/cleave.js/>).

**Задание 8**

Разработать страницу, посвященную описанию нормального распределения теории вероятностей – по аналогии со страницей на Wikipedia (<https://en.wikipedia.org/wiki/Normal_distribution>). Страница содержит:

* Краткое описание распределения
* Формулы его основных характеристик – от Notation до Variance (в соответствии с порядком на Wikipedia)
* Графики плотностей распределения для различных значений параметров, графики функций распределения для различных значений параметров.

Математические формулы выводить на экран с использованием библиотеки KaTeX (<https://katex.org/>).

Графики функций и плотностей распределений построить с использованием библиотеки Chart.js (<https://www.chartjs.org>).

**Задание 9**

Разработать страницу, представляющую собой небольшую Excel-подобную таблицу студентов, которым выставляются баллы за лабораторные работы. Таблица должна поддерживать добавление новых студентов и изменение оценок в интерактивном режиме – как в Excel. Таблица содержит:

* Колонка с именами студентов. Изначально задано 5 человек. Можно добавлять еще.
* Колонки с оценками за лабораторные работы (минимум 3). Оценки можно менять, добавлять
* Колонки со средним и суммарным значением оценок для каждого студента. Значения в колонках автоматически меняются при изменении оценок студента в таблице
* Строка с общим итогом по всем студентам – количество студентов, средняя оценка за каждую лабораторную работу.

Для реализации Excel-подобной интерактивной таблицы в HTML воспользоваться библиотекой Jexcel (<https://bossanova.uk/jexcel/v4/>).

**Задание 10**

Разработать страницу, представляющую собой форму для внесения персональных данных о человеке. На странице содержатся следующие обязательные поля (другие атрибуты могут быть добавлены студентом по желанию). Реализовать автоматическую валидацию (корректность) вносимых данных в соответствии с их ограничениями:

* Фамилия, имя, отчество
* Дата рождения – указать диапазоны минимальной и максимальной дат
* Паспорт – серия, номер, дата выдачи, место выдачи
* Адрес местожительства
* Номер мобильного телефона (в России)
* Email
* ИНН
* СНИЛС
* Данные о кредитной карте: номер, срок действия, CVC код – в соответствии с международными форматами

При нажатии на кнопку отправки формы – вся информация валидируется. При наличии некорректной информации – она подсвечивается красным. Если вся информация корректна, она выводится в «консоль» в HTML-документе (печатается).

Для реализации автоматической валидации введенных данных воспользоваться библиотекой Parsley (<https://parsleyjs.org/>).