



Утверждаю
Заместитель начальника колледжа
по учебно-методической работе
А.И.Колесниченко

МДК 01.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Методические указания для выполнения практических занятий
по специальности

09.02.07 Информационные системы и программирование

МО-09 02 07-МДК.01.01.04.ПЗ

РАЗРАБОТЧИКИ

ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ Круглень В.Ю.

ГОД РАЗРАБОТКИ 2024

ГОД ОБНОВЛЕНИЯ	2025
----------------	------

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 2/23

Содержание

Практическая работа №1. Первая программа	3
Практическая работа №2. Преобразование типов	4
Практическая работа №3. ОСНОВНЫЕ КОНСТРУКЦИИ ЯЗЫКА	6
Практическая работа №4. Функции	7
Практическая работа №5. Создание пользовательских объектов.....	9
Практическая работа №6. Объекты.....	11
Практическая работа №7. Основы работы с DOM	13
Практическая работа №8. Работа с HTML кодом элементов на JavaScript	18

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 3/23

Практическая работа №1. Первая программа

Цель занятия: Познакомить обучающихся с основными приемами работы с JavaScript;

Исходные данные: раздаточный материал

Содержание и порядок выполнения задания:

Задачи

1. Вызвать alert

Создайте страницу, которая отобразит сообщение «Я JavaScript!».

2. Покажите сообщение с помощью внешнего скрипта

Возьмите решение предыдущей задачи Вызвать alert, и измените его. Извлеките содержимое скрипта во внешний файл alert.js, лежащий в той же папке.

Откройте страницу, убедитесь, что оповещение работает.

3. Работа с переменными

1. Объявите две переменные: admin и name.
2. Запишите строку "Джон" в переменную name.
3. Скопируйте значение из переменной name в admin.
4. Выведите на экран значение admin, используя функцию alert (должна показать «Джон»).

4. Придумайте правильные имена

1. Создайте переменную для названия нашей планеты. Как бы вы её назвали?
2. Создайте переменную для хранения имени текущего посетителя сайта. Как бы вы назвали такую переменную?

5. Какие буквы (заглавные или строчные) использовать для имён констант?

Рассмотрим следующий код:

```
const birthday = '18.04.1982';
```

```
const age = someCode(birthday);
```

У нас есть константа birthday, а также age, которая вычисляется при помощи некоторого кода, используя значение из birthday (в данном случае детали не имеют значения, поэтому код не рассматривается).

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 4/23

Можно ли использовать заглавные буквы для имени birthday? А для age? Или одновременно для обеих переменных?

```
const BIRTHDAY = '18.04.1982'; // использовать заглавные буквы?
```

```
const AGE = someCode(BIRTHDAY); // а здесь?
```

6. Шаблонные строки

Что выведет этот скрипт?

```
let name = "Ilya";
```

```
alert( `hello ${1}` ); // ?
```

```
alert( `hello ${"name"}` ); // ?
```

```
alert( `hello ${name}` ); // ?
```

7. Простая страница

Создайте страницу, которая спрашивает имя у пользователя и выводит его.

Практическая работа №2. Преобразование типов

Цель занятия: Ознакомить обучающихся с основными понятиями работы над преобразованием типов

Исходные данные: раздаточный материал

Содержание и порядок выполнения задания:

Задачи

1. Постфиксная и префиксная формы

Чему будут равны переменные a, b, c и d в примере ниже?

```
let a = 1, b = 1;
```

```
let c = ++a; // ?
```

```
let d = b++; // ?
```

2. Результат присваивания

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 5/23

Чему будут равны переменные a и x после исполнения кода в примере ниже?

```
let a = 2;
```

```
let x = 1 + (a *= 2);
```

3. Преобразование типов

Какой результат будет у выражений ниже?

```
"" + 1 + 0
```

```
"" - 1 + 0
```

```
true + false
```

```
6 / "3"
```

```
"2" * "3"
```

```
4 + 5 + "px"
```

```
"$" + 4 + 5
```

```
"4" - 2
```

```
"4px" - 2
```

```
" -9 " + 5
```

```
" -9 " - 5
```

```
null + 1
```

```
undefined + 1
```

```
" \t \n" - 2
```

4. Исправьте сложение

Ниже приведён код, который запрашивает у пользователя два числа и показывает их сумму.

Он работает неправильно. Код в примере выводит 12 (для значения полей по умолчанию).

В чём ошибка? Исправьте её. Результат должен быть 3.

```
let a = prompt("Первое число?", 1);
```

```
let b = prompt("Второе число?", 2);
```

```
alert(a + b); // 12
```

5. Операторы сравнения

Каким будет результат этих выражений?

```
5 > 4
```

```
"ананас" > "яблоко"
```

```
"2" > "12"
```

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 6/23

```
undefined == null
undefined === null
null == "\n0\n"
null === +"\n0\n"
```

Практическая работа №3. ОСНОВНЫЕ КОНСТРУКЦИИ ЯЗЫКА

Цель занятия:

Ознакомить учащихся с основными конструкциями языка;

Исходные данные: раздаточный материал

Содержание и порядок выполнения задания:

Задачи Условное ветвление: if, '?'

1. if (строка с нулём)

Выведется ли alert?

```
if ("0") {
  alert( 'Привет' );
}
```

2. Название JavaScript

Используя конструкцию if..else, напишите код, который будет спрашивать: „Какое «официальное» название JavaScript?“

Если пользователь вводит «ECMAScript», то показать: «Верно!», в противном случае – отобразить: «Не знаете? ECMAScript!»



3. Покажите знак числа

Используя конструкцию if..else, напишите код, который получает число через prompt, а затем выводит в alert:

- 1, если значение больше нуля,

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 7/23

- -1, если значение меньше нуля,
- 0, если значение равно нулю.

Предполагается, что пользователь вводит только числа.

4. Перепишите 'if' в '?'

Перепишите конструкцию if с использованием условного оператора '?':

```
let result;
```

```
if (a + b < 4) {
  result = 'Мало';
} else {
  result = 'Много';
}
```

5. Перепишите 'if..else' в '?'

Перепишите if..else с использованием нескольких операторов '?'.

Для читаемости рекомендуется разбить код на несколько строк.

```
let message;
```

```
if (login == 'Сотрудник') {
  message = 'Привет';
} else if (login == 'Директор') {
  message = 'Здравствуй';
} else if (login == '') {
  message = 'Нет логина';
} else {
  message = '';
}
```

Практическая работа №4. Функции

Цель занятия: знакомство с основными функциями JavaScript.

Исходные данные: раздаточный материал

Содержание и порядок выполнения задания:

Задачи Функции

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 8/23

1. Обязателен ли "else"?

Следующая функция возвращает true, если параметр age больше 18.

В ином случае она запрашивает подтверждение через confirm и возвращает его результат:

```
function checkAge(age) {
  if (age > 18) {
    return true;
  } else {
    // ...
    return confirm('Родители разрешили?');
  }
}
```

Будет ли эта функция работать как-то иначе, если убрать else?

```
function checkAge(age) {
  if (age > 18) {
    return true;
  }
  // ...
  return confirm('Родители разрешили?');
}
```

Есть ли хоть одно отличие в поведении этого варианта?

2. Перепишите функцию, используя оператор '?' или '||'

Следующая функция возвращает true, если параметр age больше 18.

В ином случае она задаёт вопрос confirm и возвращает его результат.

```
function checkAge(age) {
  if (age > 18) {
    return true;
  } else {
    return confirm('Родители разрешили?');
  }
}
```

Перепишите функцию, чтобы она делала то же самое, но без if, в одну строку.

Сделайте два варианта функции checkAge:

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 9/23

1. Используя оператор ?
2. Используя оператор ||

3. Функция `min(a, b)`

Напишите функцию `min(a,b)`, которая возвращает меньшее из чисел `a` и `b`.

Пример вызовов:

```
min(2, 5) == 2
min(3, -1) == -1
min(1, 1) == 1
```

4. Функция `pow(x,n)`

Напишите функцию `pow(x,n)`, которая возводит `x` в степень `n` и возвращает результат.

```
pow(3, 2) = 3 * 3 = 9
pow(3, 3) = 3 * 3 * 3 = 27
pow(1, 100) = 1 * 1 * ... * 1 = 1
```

Создайте страницу, которая запрашивает `x` и `n`, а затем выводит результат `pow(x,n)`.

P.S. В этой задаче функция обязана поддерживать только натуральные значения `n`, т.е. целые от 1 и выше.

Практическая работа №5. Создание пользовательских объектов

Цель занятия: знакомство с основными понятиями и принципами создания пользовательских объектов.

Исходные данные: Раздаточный материал

Содержание и порядок выполнения задания:

Задачи

№ 1

Создайте объект с ключами от 1 до 7, в качестве значений содержащий имена дней недели. Выведите на экран все его элементы.

№ 2

Создайте объект с ключами от 1 до 12, в качестве значений содержащий названия месяцев. Выведите этот объект в консоль.

№ 3

Создайте объект `user` с ключами `'name'`, `'surname'`, `'patronymic'` и какими-то произвольными значениями. Выведите на экран фамилию, имя и отчество через пробел.

№ 4

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 10/23

Создайте объект `date` с ключами `'year'`, `'month'` и `'day'` и значениями, соответствующими текущему дню. Выведите созданную дату на экран в формате *год-месяц-день*.
№ 5

Дан объект:

```
let obj = {
  '1a': 1,
  'b2': 2,
  'c-c': 3,
  'd 4': 4,
  'e5': 5
};
```

Для каких ключей данного объекта кавычки обязательны, а для каких нет?

№ 6

Исправьте ошибки, допущенные в следующем коде:

```
let obj = {
  '1a': 1,
  'b2': 2,
  'c-c': 3,
  'd 4': 4,
  'e5': 5
};
```

```
console.log(obj.1a);
console.log(obj.b2);
console.log(obj.c-c);
console.log(obj.d 4);
console.log(obj.e5);
```

№ 7

Дан объект:

```
let obj = {x: 1, y: 2, z: 3};
```

Возведите в квадрат каждый элемент этого объекта.

№ 8

Получите массив ключей следующего объекта:

```
let obj = {x: 1, y: 2, z: 3};
```

№ 9

Найдите количество элементов в следующем объекте:

```
let obj = {x: 1, y: 2, z: 3};
```

№ 10

Дан следующий объект:

```
let obj = {x: 1, y: 2, z: 3};
```

Дана переменная `key`, в которой хранится один из ключей нашего объекта. Выведите с помощью этой переменной соответствующий элемент объекта.

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 11/23

Практическая работа №6. Объекты

Цель занятия: знакомство с инструментами рисования, создание пиксельного рисунка

Исходные данные: раздаточный материал

Содержание и порядок выполнения задания:

Задача

№ 1 1

Дан следующий код:

```
let test = {x: 1, y: 2, z: 3};
console.log(test);
```

Какой тип данных выведется в консоль? Это примитив или объект?

№ 1 2

Дан следующий код:

```
let test = {x: 1, y: 2, z: 3};
console.log(test.x);
```

Какой тип данных выведется в консоль? Это примитив или объект?

№ 1 3

Дан следующий код:

```
let test = [1, 2, 3];
console.log(test);
```

Какой тип данных выведется в консоль? Это примитив или объект?

№ 1 4

Дан следующий код:

```
let test = [1, 2, 3];
console.log(test[1]);
```

Какой тип данных выведется в консоль? Это примитив или объект?

№ 1 5

Дан следующий код:

```
let test1 = [1, 2, 3];
let test2 = 1;
```

```
console.log(test1);
```

Какой тип данных выведется в консоль? Это примитив или объект?

№ 1 6

Дан следующий код:

```
let test1 = [1, 2, 3];
let test2 = 1;
```

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 12/23

```
console.log(test1[test2]);
```

Какой тип данных выведется в консоль? Это примитив или объект?

№ 17

Не запуская код, определите, что выведется в консоль:

```
let arr1 = [1, 2, 3];
```

```
let arr2 = arr1;
```

```
arr1[0] = 'a';
```

```
console.log(arr2);
```

№ 18

Не запуская код, определите, что выведется в консоль:

```
let arr1 = [1, 2, 3];
```

```
let arr2 = arr1;
```

```
arr1[0] = 'a';
```

```
arr2[1] = 'b';
```

```
console.log(arr1);
```

№ 19

Не запуская код, определите, что выведется в консоль:

```
let arr1 = [1, 2, 3];
```

```
let arr2 = arr1;
```

```
arr1[0] = 'a';
```

```
arr2[0] = 'b';
```

```
console.log(arr2);
```

№ 20

Что выведется в результате выполнения следующего кода:

```
const arr = ['a', 'b', 'c'];
```

```
arr[1] = '!';
```

```
console.log(arr);
```

№ 21

Что выведется в результате выполнения следующего кода:

```
const arr = ['a', 'b', 'c'];
```

```
arr = [1, 2, 3];
```

```
console.log(arr);
```

№ 22

Что выведется в результате выполнения следующего кода:

```
const arr = ['a', 'b', 'c'];
```

```
arr = ['a', 'b', 'c'];
```

```
console.log(arr);
```

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 13/23

Практическая работа №7. Основы работы с DOM

Цель занятия: изучить основы работы с DOM-элементами.

Исходные данные: раздаточный материал

Содержание и порядок выполнения задания:

Получение DOM элемента

Каждому тегу страницы в JavaScript соответствует некоторый объект со своими свойствами. В этом объекте хранится текст тега, значения его атрибутов, а также другие полезные вещи. В JavaScript такие объекты называют *DOM элементами*.

Пусть в нашем HTML коде есть следующий тег:

```
<div id="elem">text</div>
```

Пусть в переменной **elem** лежит ссылка на этот тег. В этом случае переменная **elem** будет объектом со свойствами **id** и **textContent**. Свойство **id** будет содержать значение атрибута **id** нашего тега, а свойство **textContent** - его текст.

Сейчас мы с вами научимся получать DOM элементы страницы, чтобы в дальнейшем производить с ними какие-нибудь манипуляции.

Пусть у нас есть некоторая кнопка:

```
<input id="button" type="submit">
```

Давайте получим ссылку на эту кнопку в переменную. Для этого следует использовать метод **querySelector** специального объекта **document**. Этот метод параметром принимает CSS селектор и возвращает ссылку на найденный по этому селектору элемент.

У нашей кнопки есть атрибут **id** со значением **button**. Значит, мы можем найти эту кнопку по селектору **#button**. Итак, давайте найдем нашу кнопку и запишем ссылку на нее в переменную:

```
let button = document.querySelector('#button');
console.log(button);
```

Привязывание обработчиков к элементам в JavaScript

Давайте научим наши DOM элементы реагировать на действия пользователя сайта. Например, пользователь нажмет куда-либо мышкой, а наш код в ответ должен будет обработать это нажатие и вывести на экран какую-либо информацию.

Действия пользователя, которые мы можем отследить через JavaScript, называются *событиями*. События могут быть следующими: клик мышкой на элемент страницы, наведение мышкой на элемент страницы или наоборот - уход курсора мыши с элемента и так далее. Кроме того, есть события, не зависящие от действий пользователя, например, событие по загрузке HTML страницы в браузер.

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 14/23

Давайте для примера сделаем кнопку, по нажатию на которую на экран алертом выведется какой-то текст. Для начала сделаем HTML код кнопки:

```
<input id="button" type="submit">
```

Получим теперь ссылку на кнопку в переменную:

```
let button = document.querySelector('#button');
```

Теперь нам необходимо задать реакцию нашей кнопки при клике по ней. Для этого в JavaScript существует специальный метод **addEventListener**, первым параметром принимающий название события (клик на кнопку имеет название 'click'), а вторым параметром - функцию-коллбэк, выполняющуюся при возникновении этого события.

Давайте, например, по клику на кнопку выведем какой-нибудь текст:

```
button.addEventListener('click', function() {
    console.log('!!!');
});
```

Задание №1

Даны 3 кнопки:

```
<input id="button1" type="submit">
```

```
<input id="button2" type="submit">
```

```
<input id="button3" type="submit">
```

Сделайте так, чтобы по клику на первую кнопку на экран выводилось число 1, по клику на вторую – число 2, а по клику на третью – число 3.

Именованные обработчики событий в JavaScript

До этого мы использовали в качестве обработчиков событий анонимные функции. Это на самом деле не обязательно - функция может быть и обычной, с именем. Пусть для примера у нас есть такая функция:

```
function func() {
    console.log('!!!');
}
```

Пусть также есть кнопка:

```
<input id="button" type="submit">
```

Давайте сделаем так, чтобы по клику на кнопку выполнялась наша функция **func**. Для этого параметром **addEventListener** передадим имя нашей функции, вот так:

```
let button = document.querySelector('#button');
button.addEventListener('click', func);
```

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 15/23

```
function func() {
    console.log('!!!');
}
```

Задание №2

Дан следующий HTML код:

```
<input id="button1" type="submit" value="button1">
<input id="button2" type="submit" value="button2">
```

Даны следующие функции:

```
function func1() {
    console.log(1);
}
function func2() {
    console.log(2);
}
```

Сделайте так, чтобы по клику на первую кнопку выполнялась функция **func1**, а по клику на вторую – функция **func2**.

Один обработчик ко многим элементам в JavaScript

Одну функцию можно привязать сразу к нескольким элементам. Пусть для примера у нас есть следующая функция:

```
function func() {
    console.log('!!!');
}
```

Есть также две кнопки:

```
<input id="button1" type="submit" value="button1">
<input id="button2" type="submit" value="button2">
```

Получим ссылки на эти кнопки в переменные:

```
let button1 = document.querySelector('#button1');
let button2 = document.querySelector('#button2');
```

А теперь давайте привяжем нашу функцию **func** и к первой, и ко второй кнопке:

```
button1.addEventListener('click', func);
button2.addEventListener('click', func);
```

Задание №3

Даны 5 абзацев:

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 16/23

```
<p id="elem1">text</p>
```

```
<p id="elem2">text</p>
```

```
<p id="elem3">text</p>
```

```
<p id="elem4">text</p>
```

```
<p id="elem5">text</p>
```

Дана следующая функция:

```
function func() {
    console.log('message');
}
```

Привяжите эту функцию ко всем 5-ти абзацам.

Несколько обработчиков одного события в JavaScript

К одному элементу можно привязать сразу несколько функций. Давайте посмотрим на примере. Пусть у нас есть кнопка:

```
<input id="button" type="submit">
```

Пусть у нас есть две функции:

```
function func1() {
    console.log('1');
}
```

```
function func2() {
    console.log('2');
}
```

Получим ссылку на нашу кнопку в переменную:

```
let button = document.querySelector('#button');
```

А теперь давайте привяжем к нашей кнопке в качестве обработчиков клика и первую, и вторую функции:

```
button.addEventListener('click', func1);
button.addEventListener('click', func2);
```

Задание №4

Дан абзац: `<p id="elem">text</p>`

Даны следующие функции:

```
function func1() {
    console.log('1');
}
```


МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 17/23

```
function func2() {
    console.log('2');
}
function func3() {
    console.log('3');
}
```

Привяжите все эти функции к нашему абзацу.

Обработчики разных событий в JavaScript

Кроме клика по элементу, существуют и другие события. Например, с помощью события **dblclick** можно отловить двойной клик по элементу, с помощью события **mouseover** - наведение курсора на элемент, а с помощью события **mouseout** - уход курсора с элемента.

При этом к одному элементу можно привязывать обработчики различных типов событий. Давайте, например, привяжем к одному элементу реакцию на наведение курсора и реакцию на уход:

```
button.addEventListener('mouseover', function() {
    console.log('1');
});
button.addEventListener('mouseout', function() {
    console.log('2');
});
```

Задание №5

Дана кнопка. По двойному клику по ней выведите какое-нибудь сообщение.

Задание №6

Дана кнопка. По наведению на нее выведите какое-нибудь сообщение.

Задание №7

Дана кнопка. По уходу курсора с нее выведите какое-нибудь сообщение.

Задание №8

Дана кнопка. По наведению на нее выведите одно сообщение, а по уходу с нее - другое.

Работа с текстом элементов на JavaScript

У DOM элементов есть свойство **textContent**, позволяющее прочитывать текст этих элементов. Давайте посмотрим на примере. Пусть у нас есть следующий тег:

```
<p id="elem">text</p>
```

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 18/23

Получим ссылку на этот тег в переменную:

```
let elem = document.querySelector('#elem');
```

Прочитаем текст тега:

```
console.log(elem.textContent);
```

Поменяем текст тега:

```
elem.textContent = '!!!';
```

Задание №9

Дан абзац и кнопка. По клику на кнопку прочитайте текст абзаца и выведите его в консоль.

Задание №10

Дан абзац и кнопка. По клику на кнопку запишите в абзац новый текст.

Задание №11

Даны два абзаца, содержащие своим текстом какие-то числа, и кнопка. По нажатию на кнопку выведите на экран сумму хранящихся чисел.

Задание №12

Дан абзац с числом и кнопка. По нажатию на кнопку прибавьте к значению абзаца единицу и запишите полученное число обратно.

Задание №13

Дан абзац с текстом и кнопка. По нажатию на кнопку запишите в конец текста абзаца восклицательный знак.

Практическая работа №8. Работа с HTML кодом элементов на JavaScript

Цель занятия: освоение методики работы с HTML кодом элементов на JavaScript

Исходные данные: раздаточный материал

Содержание и порядок выполнения задания:

Работа с HTML кодом элементов на JavaScript

У DOM элементов есть свойство **innerHTML**, позволяющее прочитывать HTML код этих элементов. Давайте посмотрим на примере. Пусть у нас есть следующий тег:

```
<p id="elem"><b>text</b></p>
```

Получим ссылку на этот тег в переменную:

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 19/23

```
let elem = document.querySelector('#elem');
```

Прочитаем HTML код тега:

```
console.log(elem.innerHTML); // выведет <b>text</b>
```

Поменяем текст тега:

```
elem.innerHTML = '<i>!!!</i>';
```

Задание №1

Дан абзац и кнопка. По клику на кнопку запишите в абзац новый текст так, чтобы он был полужирным курсивом красного цвета, размер шрифта - 7.

Атрибуты тегов как свойства в JavaScript

Давайте теперь научимся получать атрибуты тегов. Здесь действует следующее правило: *каждому атрибуту тега соответствует одноименное свойство DOM элемента.*

Давайте посмотрим на примере. Пусть у нас есть вот такой тег:

```
<input id="elem" type="text">
```

Получим ссылку на наш элемент в переменную:

```
let elem = document.querySelector('#elem');
```

Выведем значения нужных нам атрибутов:

```
console.log(elem.id); // выведет 'elem'
```

```
console.log(elem.type); // выведет 'text'
```

А теперь для примера поменяем значение атрибута:

```
elem.type = 'submit';
```

Задание №2

Дан следующий инпут:

```
<input id="elem" type="email">
```

Дана также кнопка. По нажатию на кнопку выведите на экран содержимое атрибута **type** указанного выше инпута.

Задание №3

Дан следующий инпут:

```
<input id="elem" type="email">
```

Дана также кнопка. По нажатию на кнопку запишите в атрибут **type** значение **submit**.

Задание №4

Пусть у вас есть ссылка в виде тега **a**, кнопка и абзац. По нажатию на кнопку выведите в абзац и в консоль содержимое атрибута **href** ссылки.

Задание №5

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 20/23

Пусть у вас есть ссылка и кнопка. По нажатию на кнопку добавьте в конец текста ссылки содержимое ее атрибута **href** в круглых скобках.

Задание №6

Пусть у вас есть картинка. Разместите ее на странице с помощью тега **img**. Пусть даны также кнопка и абзац. Сделайте так, чтобы по клику на кнопку в абзац записался путь к картинке из ее атрибута **src**.

Задание №7

Дана картинка в теге **img** и кнопка. По нажатию на кнопку в атрибут **width** запишите значение 300.

Задание №8

Дана картинка в теге **img** и кнопка. Пусть в атрибуте **width** задана некоторая ширина. Сделайте кнопку, по нажатию на которую ширина картинки будет увеличиваться в 2 раза.

Задание №9

Пусть у вас есть две картинки. Сделайте на странице тег **img** и две кнопки. По нажатию на первую кнопку запишите в атрибут **src** путь к первой картинке, а по нажатию на вторую - путь ко второй картинке.

Работа с текстовыми полями в JavaScript

Сейчас мы с вами научимся получать текст от пользователей нашего сайта. Для этого в HTML предусмотрен специальный тег **input**, представляющий собой текстовое поле для ввода данных.

`<input>`

У данного тега есть специальный атрибут **value**, задающий начальный текст, который будет написан в инпуте по заходу на страницу:

`<input value="text">`

Пользователь нашего сайта после захода на страницу может поменять текст инпута. При этом, если у нас есть переменная, содержащая ссылку на этот инпут, то свойство **value** этой переменной всегда будет содержать текущее значение текста инпута.

Попробуем на практике. Пусть у нас инпут с атрибутом **value**:

`<input id="elem" value="text">`

Получим ссылку на этот инпут в переменную:

`let elem = document.querySelector('#elem');`

А теперь выведем на экран текущий текст инпута:

`console.log(elem.value);`

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 21/23

А теперь поменяем текст инпута на другой:

```
elem.value = 'new text';
```

Задание №10

Дан инпут и кнопка. По нажатию на кнопку запишите в инпут какой-нибудь текст.

Задание №11

Дан инпут, абзац и кнопка. По нажатию на кнопку запишите в абзац текст из инпута.

Задание №12

Даны два инпута и кнопка. В первый инпут пользователем вводится число. По нажатию на кнопку запишите во второй инпут квадрат введенного числа.

Задание №13

Даны два инпута и кнопка. По нажатию на кнопку запишите в первый инпут значение второго инпута, а во второй инпут - значение первого. Ваш код должен работать универсально, для любых значений инпутов.

Задание №14

Даны 5 инпутов, абзац и кнопка. В инпут вводятся числа. По нажатию на кнопку запишите среднее арифметическое введенных чисел в абзац.

Фокус текстовых полей в JavaScript

Пусть у нас есть инпут. Вы можете нажать на этот инпут и после этого в нем заморгает курсор-палочка и можно будет вводить в него текст.

Про такое состояние говорят, что инпут сейчас имеет *фокус ввода*. На практике это означает, что если начать вводить текст с клавиатуры, то этот текст будет попадать в тот инпут, который сейчас имеет фокус ввода. Если затем кликнуть куда-нибудь вне инпута, то этот инпут *потеряет* фокус ввода и в него нельзя будет вводить текст.

Для того, чтобы поймать момент получения или потери фокуса инпутом, в JavaScript предусмотрены специальные события: событие `focus` позволяет отловить получение фокуса инпутом, а событие `blur` - потерю. Попробуем на практике. Пусть у нас инпут:

```
<input id="elem" value="text">
```

Получим ссылку на него в переменную:

```
let elem = document.querySelector('#elem');
```

А теперь сделаем так, чтобы по получению фокуса в консоль вывелся текущий текст инпута:

```
elem.addEventListener('focus', function() {
    console.log(elem.value);
});
```

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 22/23

Задание №15

Дан инпут. По получению фокуса запишите в него число 1, а по потери фокуса – число 2.

Задание №16

Дан инпут. Пусть в него вводится число. По потери фокуса выведите на экран квадрат этого числа.

Задание №17

Дан инпут, в котором изначально есть какой-то текст. По получению фокуса инпутом очистите содержимое этого инпута.

Исключения при работе с атрибутами в JavaScript

При работе с атрибутами существует исключение - это атрибут **class**. Это слово является специальным в JavaScript и поэтому мы не можем просто написать **elem.class**, чтобы считать значение атрибута **class**. Вместо этого следует писать **elem.className**.

Давайте посмотрим на примере. Пусть у нас дан вот такой инпут:

```
<input id="elem" class="aaa bbb">
```

Давайте выведем значение атрибута class для нашего инпута:

```
let elem = document.querySelector('#elem');
console.log(elem.className); // выведет 'aaa bbb'
```

Существуют и другие исключения, например, атрибут **for**. К нему следует обращаться через **htmlFor**.

Задание №18

Дан див:

```
<div id="elem" class="content no-gap"></div>
```

Дана также кнопка. По нажатию на кнопку прочитайте и выведите на экран значение атрибута **class** нашего дива.

Задание №19

Дан див:

```
<div id="elem"></div>
```

Дана также кнопка. По нажатию на кнопку запишите в атрибут **class** нашего дива какой-нибудь класс.

Задание №20

Дан див с несколькими CSS классами, записанными через пробел:

```
<div id="elem" class="aaa bbb ccc"></div>
```

МО-09 02 07- МДК 01.01.04.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ	С. 23/23

Дана также кнопка. По нажатию на кнопку получите массив CSS классов нашего дива.